УДК 004.942+656.052.1

Інтелектуальний алгоритм управління міським трафіком транспортних засобів

Д.Г. Богуто[1], В.Ф. Комаров[1], П.К. Ніколюк[1], П.П. Ніколюк[2]
[1] *Донецький національний університет імені Василя Стуса, вул. 600-річчя, 21, м. Вінниця, 21021, Україна*
[2] *Вінницький національний технічний університет, вул. Хмельницьке шосе, 95, м. Вінниця, 21021, Україна*
e-mail: nikolyuk54@gmail.com

Дослідження представляє собою алгоритм побудови оптимального маршруту для кожного транспортного засобу (ТЗ) у великому місті з корекцією маршруту при зміні дорожньої обстановки. Технічно процедура регулювання потоків ТЗ здійснюється за рахунок динамічної взаємодії в режимі реального часу між центральним пунктом керування трафіком (ЦПКТ) та кожним ТЗ, що задав свої початкові та кінцеві координати. ЦПКТ передає кожному водієві голосові команди щодо маршруту руху до заявленого водієм кінцевого пункту як при звичайній GPS-навігації. Особливість полягає у тому, що програма аналізує динамічну ситуацію на кожному перехресті і по всьому місту і відповідно прокладає маршрут з урахуванням трафік-ситуації на кожен конкретний момент часу. Остаточною метою даного дослідження є синхронізація транспортних потоків, оптимальне використання транспортних артерій всього міста, запобігання утворенню заторів, а також супровід кожного ТЗ до місця призначення з таким розрахунком, щоб затрачений на поїздку час був мінімальним.

Ключові слова: *орієнтований навантажений граф, GPS-навігатор, алгоритм Дейкстри,оптимальний маршрут, Java-програма.*

The research is an algorithm for constructing an optimal route for each vehicle in a large city with a correction of the route taking into account the changing in a road situation. The procedure for regulating of flows of vehicles is carried out through the real-time dynamic interaction between the central management traffic system (CMTS) and vehicles which have a set of initial and final coordinates. The CMTS sends to each driver voice commands along the route up to the endpoint declared by the driver as a common GPS navigation does. The distinctive feature of the program is that it analyzes the dynamic situation at each junction throughout the city and, accordingly, sets the route taking into account the traffic situation for each particular moment of time. The ultimate goal of this research is to synchronize traffic flows, to use the transport arteries throughout the city in the optimal way, to prevent traffic congestions, as well as, track each vehicle to its destination minimizing the trip time. The Java program which implements the Dijkstra algorithm is used to create the optimal routes. The important condition for implementing this algorithm is the dynamism of the edges of the graph, which corresponds to the dynamic situation associated with an urban traffic. In this regard, the weight of the edges of the graph which models the city transport network varies according to the change in the traffic load between the adjacent intersections. Therefore, the database that stores traffic information for urban streets must be constantly updated. In our case, such updating occurs every 10 seconds. This allows the program that manages the traffic to change the vehicle route quickly choosing the optimal one.

Keywords: *oriented loaded graph, GPS-navigator, Dijkstra algorithm, optimal route, Java program.*

Исследование представляет собой алгоритм построения оптимального маршрута для каждого транспортного средства (ТС) в большом городе с коррекцией маршрута при изменении дорожной обстановки. Технически процедура регулирования потоков ТС осуществляется за счет динамического взаимодействия в режиме реального времени между центральным пунктом управления трафиком (ЦПУТ) и каждым ТС, задавшим свои начальные и конечные координаты. ЦПКТ передает каждому водителю голосовые команды по маршруту движения к заявленному водителем конечному пункту как при обычной GPS-навигации. Особенность заключается в том, что программа анализирует динамичную ситуацию на каждом перекрестке и по всему городу и соответственно прокладывает маршрут с учетом трафик-ситуации на каждый конкретный момент времени. Окончательной целью данного исследования является синхронизация транспортных потоков, оптимальное использование транспортных артерий всего города, предотвращения образования заторов, а также сопровождение каждого ТС к месту назначения с таким расчетом, чтобы затраченное на поездку время было минимальным.

Ключевые слова: *ориентированный нагруженный граф, GPS-навигатор алгоритм Дейкстры, оптимальный маршрут, Java-программа.*

1 Постановка проблеми

Дана стаття є логічним продовженням роботи [1]. Кінцевою метою дослідження є вирішення проблеми трафіку для всіх транспортних засобів (ТЗ), що знаходяться на вулицях великого міста – мегаполісу. А найголовнішою проблемою в цьому відношенні є затори. Як уникнути цього негативного явища? Як організувати проїзд кожного окремого ТЗ (а таких об'єктів у великому місті може бути понад мільйон) до заявленого водієм пункту призначення з таким розрахунком, щоб поїздка зайняла найменший час? Технологія, що пропонується, дозволяє ефективно вирішувати поставлені проблеми.

2 Аналіз останніх досліджень і публікацій

Заявлена проблема «стоїть на порядку денному» багатьох крупних міжнародних компаній, що розробляють технічні засоби організації дорожнього руху [2]. Найбільш близькою системою

до розглядуваної нами технології навігації є система навігації типу GPS/GLONASS [3]. Кожного року GPS-навігація модифікується. Деяке автомобільне навігаційне обладнання може повідомляти про затори на вулицях міста та пропонувати альтернативний маршрут об'їзду таких місць. Інтеграція с автомобілем стає глибшою, і це дозволяє задавати кожному конкретному водієві голосові маршрут-команди. Особливий вид взаємодії між автомобілем та дорожньою інфраструктурою описаний у роботі [4]. Тут описана технологія взаємодії між автомобілем та дорожньою інфраструктурою з допомогою так званих точок доступу (access points), розташованих вздовж автомобільної дороги та на перехрестях. Близькою до нашого дослідження є робота [5]. В основі технології лежать чотири складових: світлофори, детектори черги (queue detectors), дорожні відеокамери і центральна контрольна система (central control system). Кожні дві секунди система моніторить ситуацію щодо зміни фаз горіння світлофорів і завдяки цьому досягається оптимізація трафіку.

Автори роботи [6] аналізують проблему розташування датчиків, оскільки їх розташування суттєвим чином впливає на те, які транспортні потоки реєструються і тому можуть бути керованими. В дослідженні [7] застосовується модифікована модель стільникових автоматів (modified cellular automata model) для вивчення процесів взаємодії між автомобілями завдяки дослідженню процесів обгону шляхом вивчення даних про трафік ТЗ. Проведено моделювання потоків ТЗ. Дослідження [8] застосовує «розумну» мережеву імітаційну модель, використовуючи в якості об'єкта дослідження центральну та західну частину міста Сінгапур. На карті міста показані автомагістралі, звичайні дороги, автобусні зупинки, комерційні зони, перехрестя та автомобільні розв'язки. Для проведення імітаційних досліджень застосовувалась імітаційна модель PARAMICS. Для реєстрації потоків ТЗ використовувались петлеві детектори (loop detectors) та камери спостереження (surveillance cameras). Можливим варіантом покращення трафіка є координоване регулювання автомобільних потоків за допомогою бездротового зв'язку між автомобілями [9]. З розвитком автотранспорту все більшого поширення набуває теорія стільникових автоматів. У цитованій вище роботі моделювання рухом через перехрестя ведеться на платформі NetLogo, що є багатоагентним програмним середовищем для моделювання різних динамічних процесів.

Винахід [10] відноситься до інтелектуальної системи керування режимом роботи світлофорів через двосторонній обмін інформацією з ЦПКТ. Перехрестя використовується переважно для перетворення інформації світлофорів на бездротові сигнали та реалізації інтелектуального управління світлофорною системою. ЦПКТ здійснює управління міськими дорогами та надає інформацію водіям автомобілів а також передає електронні карти маршрутів. Інтелектуальна система управління роботою світлофорів на базі інтелектуального терміналу дозволяє водіям своєчасно отримувати інформацію про перемикання світлофорів. Система світлофорів автоматично визначає кількість автомобілів на дорогах, щоб миттєво змінювати час проїзду для автомобілів різних напрямків. Система та спосіб, передбачені цим винаходом, допомагають водіям оптимізувати маршрути руху. Дослідження [11] розглядає проблему трафіку під кутом зору проблеми паркінгу у великому місті. Технологія запропонована компанією Siemens. Система контролює завантаженість вулиць і передає інформацію автомобілістам, використовуючи для кожної окремої стоянки інформацію, зчитувану за допомогою наземних датчиків або на основі кількості проданих паркувальних дозволів. В обох випадках система направляє водіїв безпосередньо на наявні місця для паркування, що запобігає переповненню вулиць та зменшує навантаження на трафік. Має місце інтеграція в загальну систему керування трафіком. Це дає змогу використовувати бази даних паркування для надання автомобілістам рекомендацій щодо маршрутизації вже при в'їзді в межі міста.

В роботі [12] приведені результати досліджень трафіку, проведені в штаті Юта (США).У розгляд введена система показників ефективності регулювання руху потоків ТЗ на основі аналізу даних, отриманих із спеціальних мікрохвильових датчиків. Автори [13] використовують динаміку рідини для вирішення проблем, пов'язаних із дорожнім рухом. Це дослідження представляє методологію для моделювання проблем, пов'язаних із дорожнім трафіком. Розроблена теорія може полегшити розв'язання проблеми заторів. Приблизно такого ж типу є робота [14]. Дуже цікавим є дослідження [15], присвячене взаємодії приватних автомобілів та громадського транспорту. Для розв'язання проблем такої взаємодії автори вводять поняття двотипних (бімодальних) міських мереж (bi-modal urban networks). Для організації ефективної взаємодії вказаних типів ТЗ вводяться у розгляд бімодальна Макроскопічна Фундаментальна

Діаграма (МФД), що моделює змішаний трафік ТЗ згаданих видів. Результати показують, що запропонована технологія може значно: (і) зменшити затори в мережі; (іі) поліпшити показники трафіку автобусів з точки зору часу проїзду маршруту руху; (ііі) знизити рівень скупченості ТЗ на критичних ділянках транспортної мережі. В якості об'єкта досліджень вибрана транспортна мережа Сан-Франціско.

3 Формулювання мети статті

Стратегічна мета заключається в побудові оптимальних маршрутів для кожного ТЗ та синхронізації потоків ТЗ. Ставиться задача провести кожен ТЗ по місту по оптимальному маршруту з урахуванням можливої зміни такого маршруту, що коригується кожні 10 секунд. Тобто система регулювання трафіка всієї сукупності автомобілів на трасах міста повинна прокладати динамічний – в режимі реального часу – і оптимальний маршрут кожному ТЗ, що замовляє лише кінцеву позицію маршруту ЦПКТ (стартова позиція фіксується автоматично при підключенні до ЦПКТ). Центральний комп'ютер на ЦПКТ, використовуючи комп'ютерну програму, записану знизу, співпрацює з кожним водієм та передає йому голосові команди щодо маршруту руху до заявленого водієм пункту призначення як при звичайній GPS-навігації. Особливість полягає у тому, що програма аналізує динамічну ситуацію на кожному перехресті і по всьому місту і відповідно прокладає маршрут з урахуванням ситуації на кожен конкретний момент часу; при цьому використовується комп'ютерна програма, що реалізує алгоритм Дейкстри щодо знаходження оптимального шляху. Остаточною метою даного дослідження є синхронізація транспортних потоків, оптимальне використання транспортних артерій всього міста, запобігання утворенню заторів а також супровід кожного ТЗ до місця призначення з таким розрахунком, щоб затрачений на поїздку час був мінімальним.

4 Виклад основного матеріалу

Технологія являє собою автоматизовану інтелектуальну систему регуляції дорожнього руху у великих містах, яку умовно можна розділити на два етапи. Перший етап викладений в роботі [1]. На цьому етапі здійснюється регулювання трафіку через одне окреме перехрестя, що взаємодіє із сусіднім, а на другому – через все місто. Розглянемо тепер детально другий етап регулювання трафіку. Оскільки всі перехрестя міста знаходяться під контролем ЦПКТ, то в даному разі є можливість застосування «розумної» технології регулювання проїзду ТЗ із стартової позиції S до фінішної F (останню позицію кожен водій задає ЦПКТ). При цьому важливо знайти оптимальний маршрут руху для кожного ТЗ (у великих містах таких об'єктів може бути понад мільйон) на основі використання отриманих з кожного перехрестя даних, що дуже швидко змінюються. Останнє означає, що потрібно використовувати динамічну базу даних, спроможну оновлюватись, скажімо, кожні 10 с. Іншими словами, канали «Перехрестя ↔ ЦПКТ» працюють в режимі реального часу, постійно оновлюючи дані про завантаженість ділянок дороги між перехрестями.

Отже, завдання полягає у прокладанні оптимального маршруту руху для кожного ТЗ з урахуванням ситуації на транспортній мережі міста в кожен конкретний момент. Пакети даних поступають з кожного перехрестя на ЦПКТ, який працює на основі спеціальної комп'ютерної програми. У відповідь на вхідну інформацію ЦПКТ видає керуючі сигнали на світлофори кожного перехрестя. Оскільки всі перехрестя міста перебувають під контролем вказаного типу, ЦПКТ постійно володіє ситуацією щодо завантаженості перехресть та ділянок дороги між ними. Така ситуація дозволяє не тільки покращити проїзд через окреме перехрестя, а і дає можливість прокладати маршрут кожному i-му ТЗ ($i = 1$ё N, де N- число автомобілів, що замовили маршрут ЦПКТ, тобто заявили свою пару (S_i, F_i). Крім того водій – при необхідності – заявляє також мову супроводу. Наприклад, водій-українець у Сакраменто (США) заявляє в якості звукового гіда українську мову.

Технічно ситуація виглядає наступним чином: програма на ЦПКТ працює з кожним ТЗ і для кожного такого об'єкта розраховує оптимальний маршрут руху і передає дані водієві на GPS-навігатор чи на мобільний телефон із спеціальним додатком. «Знаючи» розрахований оптимальніший маршрут – на даний момент часу – ЦПКТ «веде» водія по цьому маршруту, тобто постійно передає інформацію про те, де водій має повернути і в який бік, розвернутись чи перелаштуватись на іншу смугу руху і т.п. Але дорожня ситуація у сучасному місті змінюється

щосекундно. І тому по мірі руху програма контролює маршрут кожного ТЗ та знаходиться в постійному пошуку найефективнішого маршруту. Якщо для даного водія був прокладений певний маршрут на момент часу t = 0 і в момент часу t>0 ситуація на цьому конкретному маршруті погіршилась (затор, ДТП тощо), тобто з'явився новий оптимальний маршрут, то програма «поведе» водія по цьому новому маршруту. І так далі, поки ТЗ із номером i не досягне кінцевого пункту F_i.

Технічно проїзд ТЗ через все місто, в якому кожен світлофор оснащений описаною в [1] системою регулювання трафіком, здійснюється з допомогою алгоритму, який із всіх можливих варіантів маршруту між точками S_i і F_i обирає оптимальний. Тут оптимальність визначається з допомогою виконання наступної умови [1]:

$$\sum_{h=1}^{f} (N_{A_hB_h} / n_{A_hB_h} \to min, \qquad (1)$$

де $N_{A_hB_h}$ – число ТЗ, що в'їжджають на ділянку дороги A_hB_h одного напрямку; $n_{A_hB_h}$ – число ТЗ, що виїжджають із ділянки дороги A_hB_h одного напрямку; h - індекс, що нумерує проїзну частину дороги одного напрямку руху вздовж маршруту; іншими словами, ребро графа між інцидентними (сусідніми) вузлами (перехрестями), що зчеплені між собою та утворюють простий ланцюг, з'єднуючи початкову S_i та кінцеву F_i координати, заявлені водієм автомобіля i. Символ f означає число смуг виду A_hB_h, які формують прокладений маршрут із мультиплікат типу (1).

Співставимо транспортну мережу міста (рис.1) з орієнтованим навантаженим графом (рис.2). Процедура прокладання маршруту здійснюється з допомогою програми MiniWay, яка використовує алгоритм Дейкстри [16]. Для складання такої програми в нашій роботі використовується мова програмування Java. Програма використовує в якості даних, які вводять в консолі, набір всіх значень виду (1), величини яких (для деякого моменту часу!) представлені на рис.2 біля ребер графа.

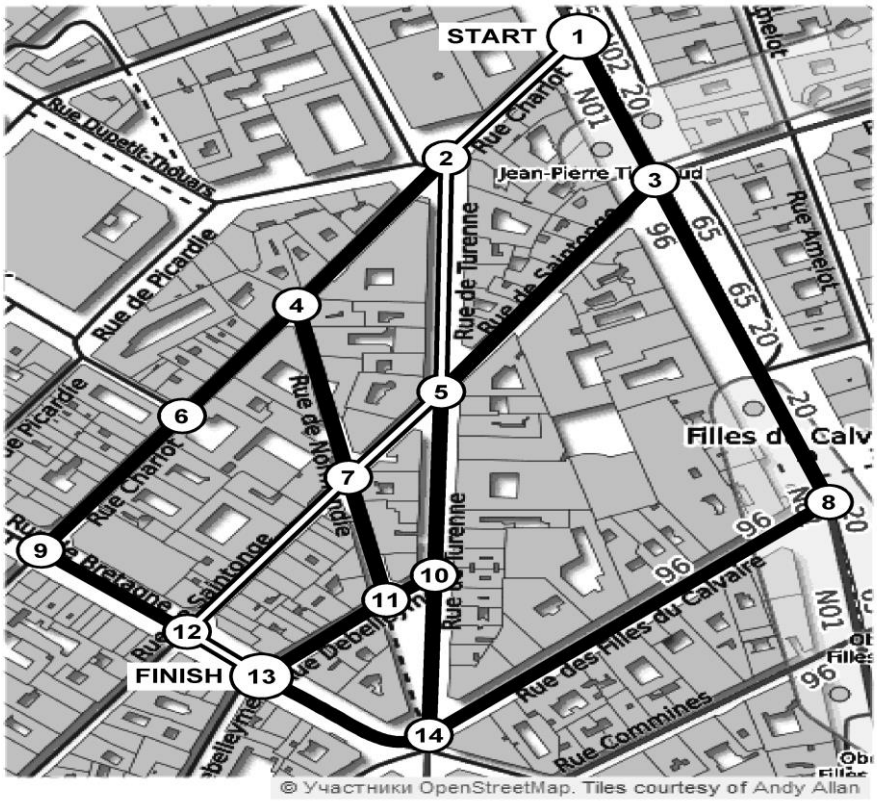

Рис.1 Транспортна мережа центрального району Парижа. Тут зроблені позначення перехресть (цифри від 1 до 14). Перехрестя 1 відмічене надписом START, а перехрестя 13 визначено як кінцевий пункт маршруту – (FINISH).

Перед технічним впровадженням пропонованого проекту необхідно здійснити імітаційне моделювання, тобто провести електронну апробацію даної технології. Дуже перспективним методом апробації є програма візуального імітаційного моделювання AnyLogic Profesional 7.3.7 [17], в якій представлено ряд імітаційних моделей, зокрема чотири імітаційні моделі, що регулюють дорожній трафік. Найбільш ілюстративною є імітаційна модель Traffic Light Phases Optimization, що візуально імітує проїзд ТЗ через три сусідні перехрестя – два Т-подібні перехрестя та одне хрестоподібне. Програма дозволяє максимізувати інтенсивність проїзду ТЗ через кожне із названих перехресть за рахунок зміни вручну з допомогою слайдерів протяжності фаз горіння зеленого світла у взаємно перпендикулярних напрямках проїзних частин дороги.

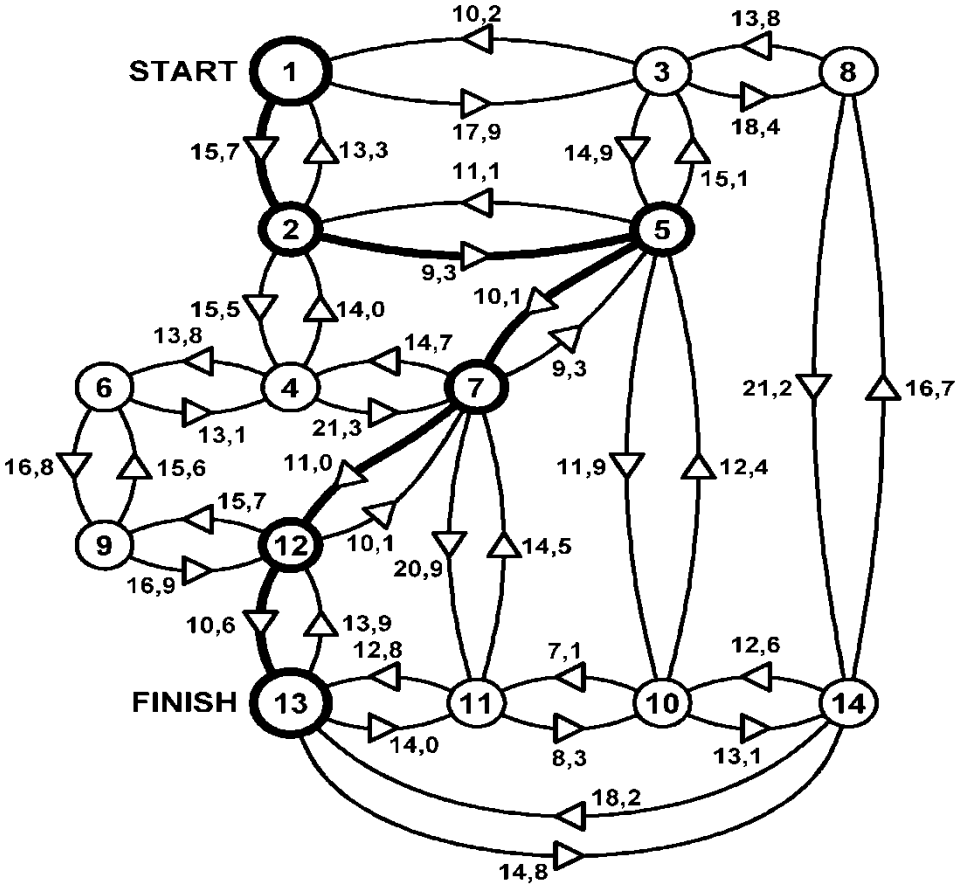

Рис.1.Орієнтований зв'язний навантажений граф. Напрямки ребер задаються трикутниками, біля яких приведені ваги ребер. Написи «START» та «FINISH» відповідають вершинам графа відповідно 1 та 13 і символізують собою початок та кінець маршруту конкретного ТЗ. Один із можливих маршрутів між пунктами 1 і 13 для певного моменту часу зображений жирною лінією.

```
package MiniWay;
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 дуг у графі
private ArrayList adj[];//список суміжності
private ArrayList weight[];//вага ребра в орграфі
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 із стартової вершини
ist[s] = 0; //найкоротша відстань від стартової вершини рівна 0
for (int k = 0; k < n; ++k) {
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 файл, 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();
dejkstra(start);
printData();
cin.close();
cout.close();
}
public static void main(String[] args) throws IOException {
MiniWay solution = new MiniWay();
solution.run();
}}
```

Для кожного заданого маршруту програма, що керує трафіком, повинна вибрати оптимальніший маршрут, тобто для кожної пари (S_i, F_i) підібрати такий комплект із (1), який є нерозривним маршрутом та відповідає мінімуму величини.

Застосуємо приведену вище програму, написану мовою програмування JAVA, для знаходження оптимального маршруту, застосувавши цю програму до графа, приведеного на рис.1, вважаючи, що шуканий маршрут пролягає від вузла (перехрестя) 1 (START) до вузла (перехрестя) 13 (FINISH). Приведений знизу спектр даних означає наступне: ***14*** – число вершин графа; ***40*** – число ребер графа; ***1*** – номер вершини, від якої стартує маршрут. Далі приведено 40 тріад чисел, перше з яких означає номер вихідної вершини, друге – номер вершини, до якої прокладається маршрут, третє – це дійсне число, що являє собою вагу ребра графа. Ці дані ми вводимо в консолі. Особливо наголосимо, що приведений спектр дійсних чисел (третя колонка) оновлюється кожні 10 секунд – у відповідності із зміною дорожнього трафіка. Приведені величини розраховуються на основі мультиплікати виду (1). Дані для відповідних розрахунків отримуються в результаті аналізу даних із вхідних та вихідних датчиків.

14 ***40*** ***1***	13 14 14.8	
1 2 15.7	7 5 9.3	7 11 20.9
2 1 13.3	7 4 14.7	11 7 14.5
1 3 17.9	4 7 21.3	10 5 12.4
3 1 10.2	4 6 13.8	5 10 11.9
3 8 18.4	6 4 13.1	8 14 21.2
8 3 13.8	6 9 16.8	14 8 16.7
3 5 14.9	9 6 15.6	13 11 14.0
5 3 15.1	9 12 16.9	11 13 12.8
2 5 9.3	12 9 15.7	11 10 8.3
5 2 11.1	12 7 10.1	10 11 7.1
2 4 15.5	7 12 11.0	10 14 13.1
4 2 14.0	12 13 10.6	14 10 12.6
5 7 10.1	13 12 13.9	14 13 18.2

Результат роботи програми – спектр чисел (виділених жирним шрифтом), що представляють собою відстані від вершини 1 до інших 14 вершин. Потім ідуть 14 рядків, що являють собою оптимальні маршрути від вершини 1 до інших вершин графа. Власне нас цікавить маршрут виду «*1® 13*», що показує конкретний шлях від вершини (перехрестя) 1 до вершини 13 – кінцевої позиції маршруту. Саме цей маршрут буде передавати водієві (на певний момент часу!) GPS-навігатор.

0.0 15.7 17.9 31.2 25.0 45.0 35.1 36.3 61.8 36.9 44.0 46.1 *56.7* 50.0

1: *1*
2: *1® 2*
3: *1® 3*
4: *1® 2® 4*
5: *1® 2® 5*
6: *1® 2® 4® 6*
7: *1® 2® 5® 7*
8: *1® 3® 8*
9: *1® 2® 4® 6® 9*
10: *1® 2® 5® 10*
11: *1® 2® 5® 10® 11*
12: *1® 2® 5® 7® 12*
13: *1® 2® 5® 7® 12® 13*
14: *1® 2® 5® 10® 14*

Маршрут до пункту призначення 13 виглядає наступним чином: *1® 2® 5® 7® 12® 13*. Програма передає дані про цей маршрут водієві, що його замовив, через GPS-навігатор або спеціальний додаток на мобільному телефоні. Цей маршрут, як і кожне ребро графа на рис.2, оновлюється кожні 10 секунд завдяки автоматизації кожного перехрестя в місті; іншими словами це означає, що ваги ребер постійно оновлюються у відповідності із даними, отримуваними із кожного перехрестя. В цьому, зокрема, є перевага нашої програми над дорожніми картами Google, які працюють із запізненням і тому констатують дорожню ситуацію постфактум, що власне і знецінює такі дані, оскільки вони не дають можливості водієві своєчасно реагувати на зміну дорожньої обстановки по його маршруту. Впровадження пропонованої системи дозволить здійснити синхронізацію руху величезної маси автомобілів (наприклад, у місті Пекін на вулиці курсує понад 1 млн. ТЗ).

Кожен водій, що під'єднався до ЦПКТ, отримує вказівки щодо руху по маршруту. Більше водію немає про що турбуватись – програма ЦПКТ буде передавати голосові команди водієві, як це робить звичайний GPS-навігатор. При зміні дорожньої обстановки (заблоковане перехрестя, ДТП по маршруту і т.д.) програма миттєво передає водієві новий розрахований, але оптимальний маршрут.

ЛІТЕРАТУРА

1. Богуто Д.Г., Волинець В.І, Ніколюк П.К., Ніколюк П.П. Автоматизована система керування рухом транспортних засобів в межах міста / Д.Г. Богуто, В.І. Волинець, П.К. Ніколюк, П.П. Ніколюк // Вісник Харківського університету, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». – №5. – 2017. – С. 3-9.
2. A to Z List of Road Traffic Contractors - Road Traffic Technology [Електронний ресурс]. – Режим доступу: www.roadtraffic-technology.com/contractors/.
3. Как избавиться от пробок на дороге? - 1Gai [Електронний ресурс]. – Режим доступу: http://www.1gai.ru/publ/512991-kak-izbavitsya-ot-probok-na-doroge.html.
4. Hyuk Lim, In Shick Kim, Ryangsoo Kim, "Vehicle, vehicle cloud system, and data dissemination system " U.S. Patent 20150153176, June 4, 2015.
5. Smart Traffic Management - Smarter Cambridge Transport [Електронний ресурс]. – Режим доступу: www.smartertransport.uk.
6. Marco Rinaldi, Francesco Viti, "Exact and approximate route set generation for resilient partial observability in sensor location problems",Transportation Research Part B: Methodological, vol.105, № 11, pp. 86 – 119, 2017.
7. Gaurav Pandey, K. Ramachandra Rao, Dinesh Mohan, "Modelling vehicular interactions for heterogeneous traffic flown using cellular automata with position preference", J. Mod. Transport., vol. 25, №3, pp.163–177, 2017.
8. A. A. Memon, M. Meng, Y. D. Wong , S. H. Lam, "Calibration of a rule-based intelligent network simulation model", J. Mod. Transport, vol. 24, №1, pp.48–61, 2016.
9. Wei Wu, Yang Liu, Yue Xu, Quanlun Wei, Yi Zhang, "Traffic Control Models Based on Cellular Automata for At-Grade Intersections in Autonomous Vehicle Environment" Journal of Sensors, vol. 2017, Article ID 9436054, 6 pages, https://doi.org/10.1155/2017/9436054.
10. Xu Chunmao, Xu Jin, "Intelligent terminal based intelligent traffic light system and method", China Patent CN104575066, April 29, 2015.
11. Case studies for traffic solutions - Siemens [Електронний ресурс] – Режим доступу: www.mobility.siemens.com/...solutions/.../case-studies-for-tr.
12. David K. Chang, Mitsuru Saito, Grant G. Schultz, Dennis L. Eggett, "Use of Hi-resolution data for evaluating accuracy of traffic volume counts collected by microwave sensors", Journal of traffic and transportation engineering",vol.4, Is. 5, pp. 423-435, 2017.
13. Dazhi Sun, Jinpeng Lv, S. Travis Waller, "In-depth analysis of traffic congestion using computational fluid dynamics (CFD) modeling meth", Journal of Modern Transportation, vol.1, № 1, pp. 58-67, 2011.
14. T.R. Gopalakrishnan Nair, Kavitha Sooda, "Comparison of Genetic Algorithm and Simulated Annealing Technique for Optimal Path Selection in Network Routing", arXiv: 1001.3920. Available at: http://arxiv.org/abs/1001.3920 (2010).
15. Konstantinos Ampountolas, Nan Zheng, Nikolas Geroliminis, "Macroscopic modelling and robust control of bi-modal multi-region urban road networks", Transportation Research Part B: Methodological, vol.104, pp.616–637, 2017.
16. Cybern.ru » Алгоритм Дейкстры (реализация на Java) [Електронний ресурс]. – Режим доступу: cybern.ru/algoritm-dejkstry-realizaciya-na-java.html.

REFERENCIES

1. D.G. Boguto, V.I. Volynets, P.K. Nikolyuk, P.P. Nikolyuk, "Automated control system of motor vehicles within the city", Bulletin of Kharkiv University, series "Mathematical Modeling. Information Technology. Automated Control Systems", Is.35, pp. 3-9, 2017.

2. “A to Z List of Road Traffic Contractors - Road Traffic Technology”. Available: www.roadtraffic-technology.com/contractors/.
3. “How to get rid of traffic jams on the road? - 1Gai”. Available: http://www.1gai.ru/publ/512991-kak-izbavitsya-ot-probok-na-doroge.html.
4. Hyuk Lim, In Shick Kim, Ryangsoo Kim, “Vehicle, vehicle cloud system, and data dissemination system “ U.S. Patent US20150153176, June 4, 2015.
5. “Smart Traffic Management - Smarter Cambridge Transport”, Available: www.smartertransport.uk.
6. Marco Rinaldi, Francesco Viti, “Exact and approximate route set generation for resilient partial observability in sensor location problems”,Transportation Research Part B: Methodological, vol.105, № 11, pp. 86 – 119, 2017.
7. Gaurav Pandey, K. Ramachandra Rao, Dinesh Mohan, “Modelling vehicular interactions for heterogeneous traffic flown using cellular automata with position preference”, J. Mod. Transport, vol. 25, №3, pp.163–177, 2017.
8. A. A. Memon, M. Meng, Y. D. Wong , S. H. Lam, “Calibration of a rule-based intelligent network simulation model”, J. Mod. Transport, vol. 24, №1, pp.48–61, 2016.
9. Wei Wu, Yang Liu, Yue Xu, Quanlun Wei, Yi Zhang, “Traffic Control Models Based on Cellular Automata for At-Grade Intersections in Autonomous Vehicle Environment” Journal of Sensors, vol. 2017, Article ID 9436054, 6 pages, https://doi.org/10.1155/2017/9436054.
10. Xu Chunmao, Xu Jin, “Intelligent terminal based intelligent traffic light system and method”, China Patent CN104575066, April 29, 2015.
11. “Case studies for traffic solutions - Siemens”, Available: www.mobility.siemens.com /...solutions/.../case-studies-for-tr.
12. David K. Chang, Mitsuru Saito, Grant G. Schultz, Dennis L. Eggett, “Use of Hi-resolution data for evaluating accuracy of traffic volume counts collected by microwave sensors”, Journal of traffic and transportation engineering”,vol.4, Is. 5, pp. 423-435, 2017.
13. Dazhi Sun, Jinpeng Lv, S. Travis Waller, “In-depth analysis of traffic congestion using computational fluid dynamics (CFD) modeling meth”, Journal of Modern Transportation, vol.1, № 1, pp. 58-67, 2011.
14. T.R. Gopalakrishnan Nair, Kavitha Sooda, “Comparison of Genetic Algorithm and Simulated Annealing Technique for Optimal Path Selection in Network Routing”, arXiv: 1001.3920. Available at: http://arxiv.org/abs/1001.3920 (2010).
15. Konstantinos Ampountolas, Nan Zheng, Nikolas Geroliminis, “Macroscopic modelling and robust control of bi-modal multi-region urban road networks”, Transportation Research Part B: Methodological, vol.104, P.616–637, 2017.
16. “Cybern.ru » Dijkstra's algorithm (implementation in Java)”, Available: www.cybern.ru/algoritm-dejkstry-realizaciya-na-java.html.

Богуто Денис Геннадійович *– аспірант кафедри комп’ютерних технологій; Донецький національний університет імені Василя Стуса, м. Вінниця-21, вул. 600-річчя, 21, 21021; e-mail: d.boguto@donnu.edu.ua; ORCID: 0000-0003-0367-0788.*

Комаров Василь Федорович *– старший науковий співробітник кафедри радіофізики та кібербезпеки; Донецький національний університет імені Василя Стуса, м. Вінниця-21, вул. 600-річчя, 21, 21021; e-mail: v.komarov@donnu.edu.ua; ORCID:0000-0002-8797.*

Ніколюк Петро Карпович *– доктор фізико-математичних наук, професор кафедри комп’ютерних технологій; Донецький національний університет імені Василя Стуса, м. Вінниця-21, вул. 600-річчя, 21, 21021; e-mail: nikolyuk54@gmail.com; ORCID: 0000-0002-0286-297X.*

Ніколюк Павло Петрович *– аспірант кафедри інформаційних систем та технологій; Вінницький національний технічний університет, м. Вінниця-21, вул. Хмельницьке шосе,95, 21021; e-mail: pashanikoluk@gmail.com; ORCID:0000-0003-0490-0420.*

Надійшла у першій редакції 11.01.2018, в останній - 20.04.2018.