

УДК 004.942+656.052.1

Вплив інтелектуального світлофорного регулювання на пропускну здатність міського перехрестя

Петро Ніколюк

професор, д-р фіз.-мат. наук
ORCID: 0000-0002-0286-297X
p.nikolyuk@donnu.edu.ua

Донецький національний університет імені Василя Стуса

Ключові слова:

інтелектуальне перехрестя,
п'єзокристалічний сенсор,
цикл роботи світлофора,
пропускну здатність перехрестя,
модель міського перехрестя,
AnyLogic

Міські транспортні мережі працюють у різних режимах. Знання цих режимів і переходів між ними є важливим фактором, що визначає ефективність роботи міської транспортної мережі. Перевантаження цієї мережі приводить до заторів. Уже протягом кількох десятиліть міський трафік транспортних засобів зіштовхнувся з проблемою заторів. Особливо актуальною ця проблема є у великих містах-мегаполісах, де кількість транспортних засобів стрімко зростає, а пропускну здатність транспортних мереж залишається практично незмінною. Ключовим елементом і головною причиною заторів є перехрестя міських доріг, де сходяться потоки автомобілів з різних напрямків. Першим кроком у технології регулювання міського трафіку є організація світлофорного регулювання. Зазвичай тривалість фаз зеленого та червоного світла є сталими і не враховують поточної завантаженості автомобілями різних напрямків. У цьому дослідженні йдеться про перехрестя зі світлофорним регулюванням інтелектуального типу. Таке регулювання принципово відрізняється від стандартного світлофорного регулювання. Інтелектуальна регуляція проїзду транспортних засобів через окреме регульоване перехрестя здійснюється з допомогою комп'ютерної програми, що враховує завантаженість напрямків на перехресті та здійснює відповідну корекцію протяжності фаз горіння червоного та зеленого світла з метою максимізації пропускну здатності. Пропонована процедура дасть змогу оптимізувати пропускну здатність кожного окремого перехрестя, що приведе до переходу міського трафіку на якісно новий рівень. Водночас кількість заторів на міських транспортних магістралях різко зменшиться. Система працює в режимі реального часу, тому динамічно регулює фази горіння світлофора, синхронізовані із завантаженістю різних напрямків на перехресті. Тривалість проїзду кожного автомобіля вибраним маршрутом скоротиться, подорож стане більш комфортною та безпечною.

DOI: 10.31558/2786-9482.2023.1.3

Вступ

Проблема трафіку у великому місті стає все більш актуальною, оскільки кількість транспортних засобів на вулицях зростає, а пропускну здатність міських транспортних артерій залишається практично незмінною [1, 2]. Насамперед проблема пов'язана із

заторами на перехрестях, де сходяться потоки транспортних засобів із різних напрямків. Як результат такого негативного явища спостерігається різке зростання часу проїзду кожного автомобіля до пункту призначення. Спостерігається також забруднення навколишнього повітряного басейну продуктами горіння автомобільного палива.

Певною мірою вирішенню цієї проблеми сприятиме застосування мережі «інтелектуальних» світлофорів. На кожному перехресті монтується система сенсорів (або відеокамер), що дає можливість контролювати завантаженість доріг в околі кожного перехрестя. Потоки даних із різних напрямків перехрестя надходять у блок світлофорного регулювання, де відбувається аналіз отриманих сигналів і вибір фаз горіння зеленого і червоного світла. Дослідження того, на основі якої початкової інформації і у який спосіб формувати режим роботи кожного світлофора у транспортній мережі міста, і є *метою* цієї статті.

Огляд літератури

Будемо вважати, що протяжність горіння зеленої фази світлофора встановлюється пропорційною до завантаженості транспортними засобами у вибраному напрямку. Отже, горіння зеленої фази світлофора синхронізується із кількістю транспортних засобів у напрямку горіння вказаного світла. Як результат, пропускна здатність перехрестя суттєво зросте, і відповідно затори зникнуть або стануть меншими. Організувавши оптимальний рух через такий об'єкт, досягнемо більш високої ефективності трафіку по всьому місту. Для ефективної організації міського трафіку транспортні засоби потрібно реєструвати. Одним із ефективних способів реєстрації потоків транспортних засобів є використання п'єзокристалічних сенсорів, вмонтованих у полотно дороги в околі самого перехрестя. Так, компанія International Road Dynamics Inc. розробила досить ефективну конструкцію п'єзоелектричного сенсора дорожнього руху RoadTrax BL [3, 4]. Найбільш придатними для поставленої задачі є п'єзокристалічні та індуктивні сенсори, що монтуються в дорожнє полотно. Але п'єзоелектричні сенсори більш чутливі, тому доцільно вибрати саме їх.

У дослідженні [5] наведені технічні характеристики п'єзокристалічного сенсора іншої конструкції типу 12-DOF. Це високоякісний високочутливий сенсор для реєстрації величини тисків у широкому діапазоні. Пристрій працює на частотах в околі 11 кГц з похибкою не більше 1%.

Теоретичні аспекти оптимізації трафіку на окремому перевантаженому перехресті досліджені в роботах [6, 7]. У них розглядається так звана дискретно-часова модель для хрестоподібного перехрестя, що записується у вигляді системи рівнянь такого виду:

$$J_D = \sum_{k=0}^N (q_1(k) + q_2(k)) + \frac{a_1 + a_2}{2} \cdot T \cdot \sum_{k=0}^{N-1} u(k) \rightarrow \min ; \quad (1)$$

$$q_1(k+1) \geq \max(q_1(k) + d_1 \cdot T \cdot (u(k) - u_L), a_1 \cdot T(1 - u(k))); \quad (2)$$

$$q_2(k+1) \geq \max((q_2(k) + d_2 \cdot T(u_H - u(k))), 0); \quad (3)$$

$$u_{\min} \leq u(k) \leq u_{\max}; \quad (4)$$

$$q_i(0) = q_{i,\text{int}}, \quad i = 1, 2; N = 0, 1, \dots, N-1, \quad (5)$$

де J_D – модельна величина, що являє собою кількість транспортних засобів на перевантаженому перехресті, що не перетнули його;

N – кількість циклів перемикання світлофора;

k – нумератор циклів;

$q_1(k), q_2(k)$ – кількість транспортних засобів на перехресті в k -му циклі переключення світлофора на горизонтальному та вертикальному напрямках хрестоподібного перехрестя відповідно;

a_1, a_2 – кількість транспортних засобів, що прибули за одиницю часу на горизонтальний та вертикальний напрямки відповідно;

T – період перемикання світлофора (цикл роботи);

$u(k)$ – співвідношення величин фаз горіння зеленого та червоного світла;

d_1, d_2 – кількість транспортних засобів, що перетнули перехрестя за одиницю часу у горизонтальному та вертикальному напрямках відповідно;

$$u_H = 1 - a_2/d_2, u_L = a_1/d_1;$$

u_{\min}, u_{\max} – мінімальне та максимальне значення;

$q_{i,\text{int}}$ – початкова величина транспортних засобів на перехресті для горизонтального та вертикального напрямів.

Цільова функція за вищенаведеної системи рівнянь представляється виразом (1). Вона являє собою кількість транспортних засобів, що не перетнули перехрестя. Зрозуміло, що таку величину треба мінімізувати. Керованими змінними є величини $u(k)$. Рівняння (2) і (3) представляють еволюцію з часом обох автомобільних черг. Рівняння (4) дає верхню і нижню границі в кожному циклі. Рівняння (5) представляє стартові (початкові) довжини автомобільних черг на перехресті. Загалом система рівнянь (1)–(5) являє собою задачу лінійного програмування. Графічний розв'язок цієї задачі представлений на рис. 1.

Дослідження методів обробки інформації з різноманітних джерел, як от сенсори, детектори, відеокамери і радары, проведено в роботі [9]. У ній розглядається технологія зв'язаних транспортних засобів (connected vehicles), яка дає можливість збирати та аналізувати зв'язки типу *транспортний засіб – інфраструктура* (V2I) та *транспортний засіб – транспортний засіб* (V2V). Врахування зв'язків такого типу дає змогу зменшити імовірність заторів, збільшити безпеку руху та зменшити витрати палива.

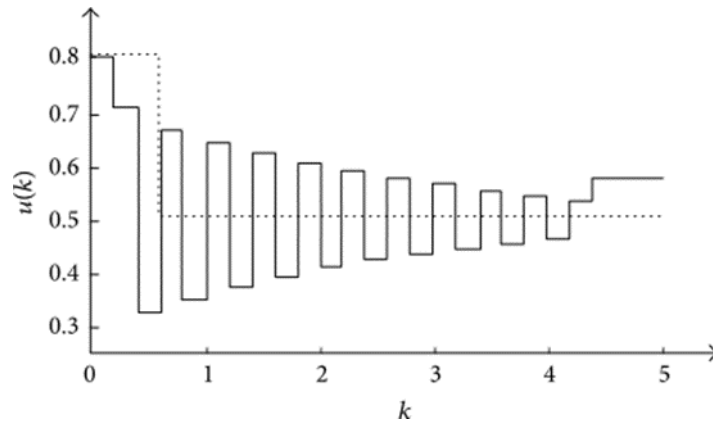


Рисунок 1. Розв'язок системи рівнянь (1)–(5):
суцільна лінія – точний розв'язок; пунктирна лінія – наближений [7]

Постановка задачі

На рис. 2 зображена схема фіксації автомобілів, що проїжджають перехрестя. На ній показані вхідні та вихідні сенсори, що реєструють проїзд автомобілів через перехрестя. Система передачі даних прив'язана до сенсорів, розташованих по периметру кожного перехрестя.

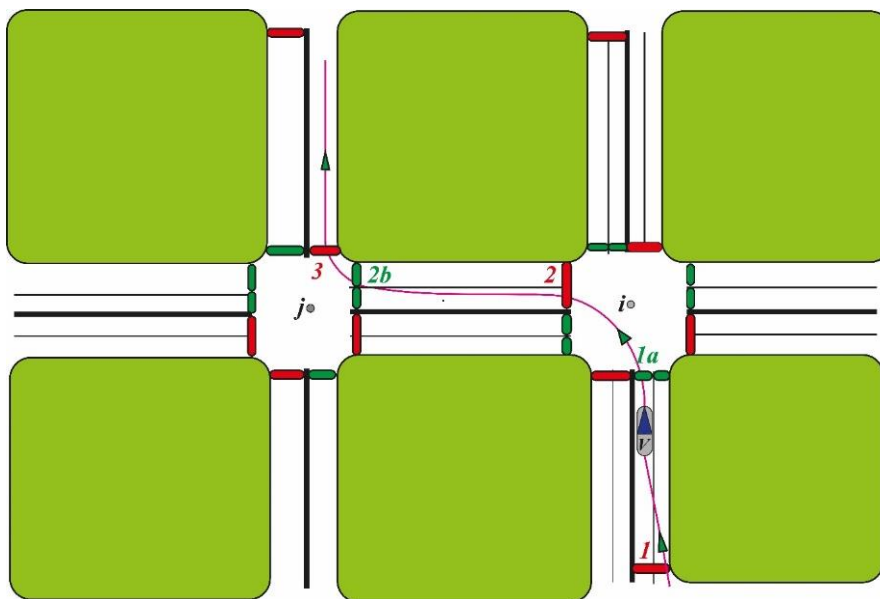


Рисунок 2. Маршрут транспортного засобу через два сусідні перехрестя

З аналітичного боку необхідно створити алгоритм та автоматизовану систему, що забезпечать ефективне перемикання світлофорних фаз, синхронізоване із завантаженістю дорожніх напрямків. Отже, йдеться про систему інтелектуальних перехрестя. Під інтелектуальним перехрестям розуміється перехрестя, для якого протяжність зеленої фази горіння світлофора у певному напрямку пропорційна кількості транспортних засобів у цьому напрямку. Пропускна здатність такого перехрестя максимальна, порівняно з будь-яким іншим варіантом регуляції транспортних потоків через такий об'єкт.

Із технічного боку реєстрація транспортних засобів здійснюється з допомогою п'єзокристалічних сенсорів, які фіксують кількість транспортних засобів, що або в'їжджають на проїзну частину дороги між сусідніми перехрестями (вхідні сенсори), або виїжджають із цієї ділянки дороги (вихідні сенсори). Водночас вхідні та вихідні сенсори на сусідніх перехрестях працюють сумісно (див. рис. 2). Це дає змогу оптимізувати пропускну здатність не тільки окремих міських перехресть як автономних об'єктів, а також і аналогічну величину для всієї транспортної мережі міста.

Ідея рішення

Під час руху автомобіля по вулицях міста важливою перешкодою стають, з одного боку, перехрестя, а з іншого – затори, які можуть виникати як на перехрестях, так і поза ними. Отже, причина виникнення заторів комплексна. Часто виникає ситуація, коли один напрямок перехрестя значно менш навантажений за перпендикулярний йому. Запобігати таким випадкам повинні інтелектуальні перехрестя.

У нашому випадку модернізація звичайного перехрестя до інтелектуального буде реалізовуватись завдяки інсталяції системи, що діє на основі принципу про те, що протяжність зеленої фази горіння світлофора напрямку пропорційна відповідній кількості транспортних засобів. Отже, інтелектуальне перехрестя здатне аналізувати трафік-ситуацію та за необхідності пролонгувати довжину зеленої фази для найбільш завантаженого напрямку. Як видно з вищенаведених міркувань, світлофорним комплексом на перехресті має керувати система, яка має:

- підраховувати кількість автомобілів на кожному з напрямків перехрестя;
- регулювати довжину сигналів на кожному зі світлофорів у строгій відповідності із трафік-ситуацією;
- досягати максимальної пропускну здатності.

Розвиваючи ідею далі, потенційно можливо об'єднати всі системи окремих перехресть так, щоб наступні перехрестя могли передбачати, де виникне затор, і запобігати цьому ще до того, як він утвориться. На абстрактному рівні систему світлофорів на інтелектуальному перехресті можна порівняти з роботою регулювальника, котрий миттєво реагує на стан трафіку, завдяки чому досягається максимальна пропускну здатність.

Кожне перехрестя оснащується двома типами сенсорів (рис. 3) – вхідними (input sensors) та вихідними (output sensors). Кількість вихідних сенсорів залежить від кількості смуг проїзної частини дороги одного напрямку (на рис. 3 таких сенсорів 3, відповідно до кількості смуг). На рис. 3 скругленими прямокутниками зображені транспортні засоби, маршрути яких пролягають у напрямку $i \rightarrow j$. Кожен вхідний та вихідний сенсори з'єднані з центром керування трафіком (на рисунку зображений лише один сигнальний провід, що виходить від одного із сенсорів).

Отже, кожна смуга оснащується власним вихідним сенсором. Для чого? Річ у тім, що автомобілі розташовуються по смугах відповідно до маршруту руху кожного з них. Наприклад, якщо автомобіль розташований у смузі, що примикає до осьової лінії дороги, то водій цього транспортного засобу може рухатись прямо, наліво або здійснювати

розворот на перехресті, але в жодному разі не повертати направо: такий маневр заборонений правилами дорожнього руху. Завдяки сумісній роботі вхідних та вихідних сенсорів є можливість реєструвати транспортні засоби, що в'їжджають на ділянку дороги між сусідніми перехрестями (див. рис. 2). Ідеться про те, що вхідний сенсор на перехресті i працює сумісно із комплектом вихідних сенсорів на перехресті j , і завдяки цьому кожна смуга ділянки дороги одного напрямку $i \rightarrow j$ знаходиться під контролем. Це дає можливість визначити як завантаженість ділянки дороги $i \rightarrow j$ загалом, так і завантаженість кожної окремої смуги на цій ділянці. Власне, будь-який маршрут складається із певної сукупності смуг на ділянках дороги одного напрямку, з'єднаних перехрестями. Отже, вся транспортна мережа міста знаходиться під контролем. Вираз *під контролем* означає, що є можливість визначати як динаміку руху транспортного засобу на кожній ділянці дороги загалом, так і на кожній окремій смузі руху цієї ділянки дороги. Отже, можна говорити про пропускну здатність як вибраної ділянки дороги $i \rightarrow j$, так і окремої смуги на цій ділянці. Мовою теорії графів, можна визначати вагу кожного ребра та всього маршруту загалом. Завдання полягає в тому, щоб знайти маршрут найменшої ваги.

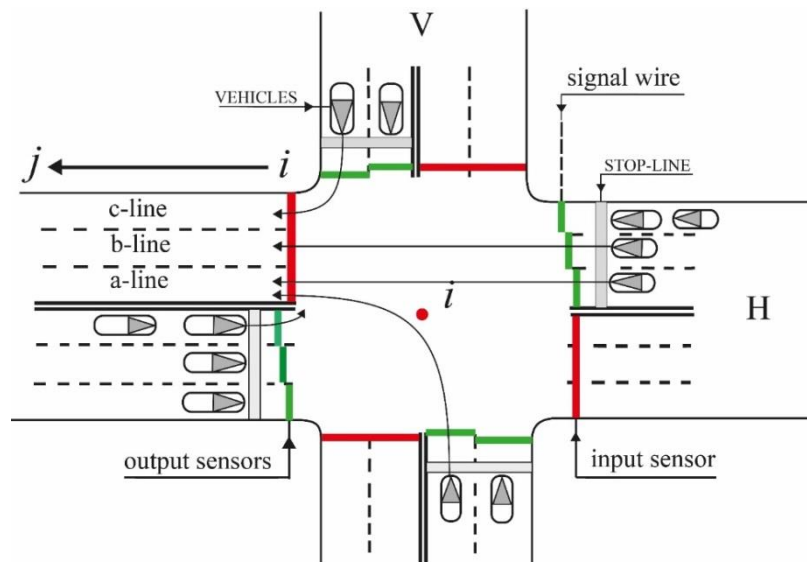


Рисунок 3. Інтелектуальне перехрестя зі вхідними та вихідними сенсорами

Автомобілі, що перетнули перехрестя i та рухаються в напрямку перехрестя j , перелаштовуються по смугах руху відповідно до їх подальшого маршруту. Якщо водій транспортного засобу має намір повернути на перехресті j направо, тоді він повинен зайняти праву смугу руху. Якщо маршрут транспортного засобу пролягає прямо в напрямку $i \rightarrow j$, то тоді водій повинен зайняти смугу b , тобто центральну смугу. Нарешті, якщо водій повертає на перехресті j направо, тоді він повинен зайняти праву смугу. Виникає питання: як реєструвати динаміку транспортних засобів по кожній зі смуг руху?

На кількісному рівні динаміка руху вздовж ділянки дороги $i \rightarrow j$ обумовлена співвідношенням між величиною транспортних засобів, що в'їхали за певний час, наприклад, один цикл переключення світлофора, та виїхали із цієї ділянки дороги.

Технічно процес організований так: вхідний сенсор на перехресті i реєструє автомобілі, які в'їжджають на ділянку дороги $i \rightarrow j$ з усіх можливих напрямків перехрестя i . Комплекс вихідних сенсорів на перехресті j реєструє автомобілі, що виїхали за межі цієї ділянки. Відношення цих величин якраз і свідчить про динаміку руху на конкретній ділянці дороги.

Моделювання роботи перехрестя

В основі всієї технології регулювання трафіку як вихідна комірка слугує окреме регульоване перехрестя. Модель інтелектуального перехрестя реалізована за допомогою програми імітаційного моделювання AnyLogic (рис. 4.) Робота моделі базується на імітації потоків транспортних засобів, що створюються за допомогою об'єктів типу carSource. Перехрестя оснащено світлофорами. Протяжність горіння зеленої фази цих світлофорів синхронізована зі ступенем завантаженості відповідних напрямків руху. Зображене на рис. 4 перехрестя дає змогу контролювати та оптимізувати пропускну здатність завдяки кореляції між протяжністю горіння зеленої фази у певному напрямку (горизонтальному чи вертикальному) та завантаженістю транспортними засобами на вказаному напрямку.

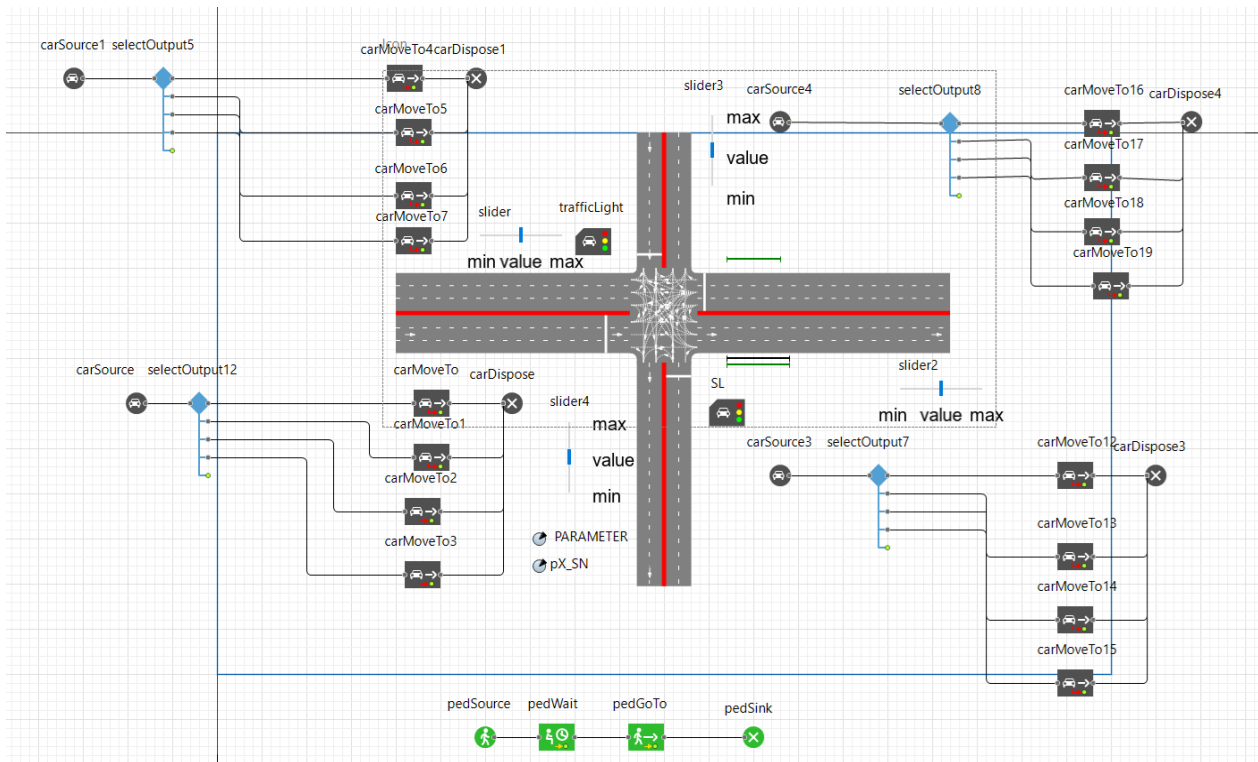


Рисунок 4. Модель інтелектуального хрестоподібного перехрестя в AnyLogic

Вважатимемо, що цикл роботи світлофора – це величина, що зазвичай складається з таких доданків:

$$T = trh + tgh + tyh + tp, \quad (6)$$

де trh – час горіння червоного світла у горизонтальному напрямку;

tgh – час горіння зеленого світла у горизонтальному напрямку;

tyh – час горіння жовтого світла також у горизонтальному напрямку;

tp – час горіння зеленого світла для пішоходів.

Таке саме співвідношення можна записати стосовно вертикального напрямку перехрестя, тобто:

$$T = trv + tgv + tyv + tp, \quad (6)$$

де trv – час горіння червоного світла у вертикальному напрямку;

tgv – час горіння зеленого світла у вертикальному напрямку;

tyv – час горіння жовтого світла так само у вертикальному напрямку.

Важливим чинником покращення надійності та достовірності даних зі вхідних та вихідних сенсорів є інформація з GPS-пристроїв. Майже кожен водій працює якщо не з GPS-навігатором, то із мобільним телефоном чи смартфоном із програмою Google Maps. Це дає змогу використати GPS-дані як додаткову інформацію. Описана ситуація детально розглядається в роботах [10–14]. Хоча GPS-дані не дають точної інформації, але для нашої задачі такі дані важливі як допоміжні, щоб правильно інтерпретувати дані від дорожніх сенсорів. Для цього статистичний характер GPS-даних цілком прийнятний, і немає необхідності точно визначати положення кожного транспортного засобу – важливим є лише загальна кількість таких об'єктів, яка є мірою завантаженості дорожніх смуг міста. Чому важливо поєднати дані з п'єзокристалічних сенсорів із GPS-даними? Річ у тім, що основну роль відіграють саме дані, отримувані зі вхідних та вихідних сенсорів, які розташовані між сусідніми перехрестями. З іншого боку, програма передбачає використання водіями GPS-навігаторів. Тому цілком логічно використати GPS-інформацію повною мірою. На доцільність поєднання стаціонарних даних та мобільних, отримуваних із GPS-пристроїв чи смартфонів, вказується, зокрема, і в роботі [15]. Подібний алгоритм використання GPS-даних для прокладання оптимальних маршрутів детально розглядається також у [16, 17]. У них використовуються набори GPS-даних, що характеризуються географічною широтою, довготою та часовим маркером. GPS-точки у цій роботі генеруються з певною вибраною частотою імітатором, і на основі них прокладається оптимальний маршрут, а також обраховується середня швидкість кожного транспортного засобу.

Алгоритм регулювання трафіку на окремому перехресті, в якому використані змінні з формул (6) та (7) у вигляді java-програми, наведено нижче:

```
import java.util.Random;
import static java.lang.StrictMath.abs;
interface Lights{
    int REDH =0;
    int YELLOWH =1;
    int GREENH =2;
    int REDV =3;
    int YELLOWV =4;
    int GREENV =5;
    int GREENP = 6;
    int ERROR = -1;
}
```



```
class T implements Lights {
    private int delay;
    private static int light = REDH;
    T(int sec) {
        delay = 1000 * sec;
    }
    public int shift() {
        int count = (light++) % 7;
        try {
            switch (count) {
                case REDH:
                    Thread.sleep(delay);
                    break;
                case YELLOWH:
                    Thread.sleep(delay / 3);
                    break;
                case GREENH:
                    Thread.sleep(delay / 2);
                    break;
                case REDV:
                    Thread.sleep(delay);
                    break;
                case YELLOWV:
                    Thread.sleep(delay / 3);
                    break;
                case GREENV:
                    Thread.sleep(delay / 2);
                    break;
                case GREENP:
                    Thread.sleep(delay);
                    break;
            }
        } catch (Exception e) {
            return ERROR;
        }
        return count;
    }
}

class TrafficRegulator {
    static int T = 96;
    private static T t = new T(1);
    static Random gn1 = new Random();
    static Random gn2 = new Random();
    static Random gn3 = new Random();
    static Random gn4 = new Random();
    TrafficRegulator() {
    }
    public static void main(String[] args) {
        double k = Math.abs ((gn1.nextDouble() + gn2.nextDouble() ) /
            (gn3.nextDouble() + gn4.nextDouble()));
        double tg = 35.0;
        double tp = 23.0;
        double tyh = 2.0;
        double tgp = 15.0;
        double tgh = k * tg;
        double trh = (double) (T - tyh - tgh - tp);
    }
}
```

```

double tgv = abs(2.0 * tg - tgh);
double trv = (double)T - trh;
int tyv = 2;
for(int j = 0; j < 7; ++j) {
    switch(t.shift()) {
        case -1:
            System.out.println("Time error!");
            break;
        case 0:
            System.out.println("red horizontal!");
            System.out.format("%.1f%n", trh);
            break;
        case 1:
            System.out.println("yellow horizontal!");
            System.out.println(tyh);
            break;
        case 2:
            System.out.println("green horizontal!");
            System.out.format("%.1f%n", tgh);
            break;
        case 3:
            System.out.println("red vertical!");
            System.out.format("%.1f%n", trv);
            break;
        case 4:
            System.out.println("yellow vertical!");
            System.out.println(tyv);
            break;
        case 5:
            System.out.println("green vertical!");
            System.out.format("%.1f%n", tgv);
            break;
        case Lights.GREENP:
            System.out.println("green pedestrian!");
            System.out.println(tgp);
            break;
        default:
            System.err.println("Unknown light.");
            return;
    }
}
}
}
}

```

Зупинимось коротко на аналізі програми, яка регулює проїзд транспортних засобів через хрестоподібне перехрестя. Базовими змінними обрано величини типу *REDH* – червоне світло в горизонтальному напрямку. Принципово те, що програма аналізує завантаженість транспортними засобами напрямків перехрестя: чим більше завантажений напрямок – горизонтальний чи вертикальний, – тим довше горітиме зелене світло у відповідному напрямку. Звичайно, є обмеження на період перемикання світлофора, яке в цьому разі вибраний рівним 96 с. Для імітації реальної ситуації на перехресті у програмі включено 4 генератори випадкових величин, що імітують завантаженість перехрестя транспортними засобами. Ключовим є відношення кількості автомобілів на горизонтальному та

вертикальному напрямках. У програмі таке відношення задається коефіцієнтом k . Варіація величини k імітує якраз відношення у змінах завантажених. Відповідно до динаміки трафіку на перехрестях змінюється і видача програми, зокрема тривалість горіння зеленого світла у горизонтальному та вертикальному напрямках. Приклад видачі наведено нижче:

red horizontal!	29.6
yellow horizontal!	2.0
green horizontal!	41.4
red vertical!	66.4
yellow vertical!	2
green vertical!	28.6
green pedestrian!	15.0

Презентована програма дає можливість суттєво покращити пропускну здатність перехрестя завдяки «розумному» режиму їх роботи в плані протяжності горіння різних фаз, синхронізовану із завантаженистю різних напрямків перехрестя.

Шляхом регулювання з допомогою слайдерів навантаженості кожного із напрямків (див. рис. 4) з використанням результатів запропонованої програми досягаємо найвищої пропускну здатності перехрестя. Для оцінювання ефекту порівнюємо із ситуацією, коли перехрестя хоча і є регульованим, проте згаданої вище корекції не відбувається. Розрахунки показують, що у випадку інтелектуального перехрестя пропускну здатність зростає у 1.7–2.2 рази. Це довгий вагомий показник. Якщо кожне регульоване міське перехрестя збільшить свою пропускну здатність хоча б у 1.7 рази, то міський трафік загалом у місті перейде на якісно новий рівень. Це приведе до різкого зменшення заторів на міських вулицях.

Висновки

Запропоновано інтелектуальний алгоритм проїзду транспортних засобів через міські перехрестя, які є головною причиною заторів (колапсу) міського трафіку. Суть процедури інтелектуалізації полягає у тому, щоб протяжність горіння зеленої фази світлофора корелювала із завантаженистю транспортними засобами відповідного напрямку на перехресті. Власне, це є головним трендом представленого дослідження. Дослідження має практичну спрямованість, його результати можуть бути використані на реальних міських перехрестях з метою покращення умов проїзду як через окреме перехрестя, так і через усе місто. Внаслідок чого динаміка руху транспортних засобів через лабіринт міських вулиць покращиться, а кількість заторів – зменшиться. Як результат – час проїзду кожного автомобіля по вибраному маршруту зменшиться.

Література

1. Богуто, Д. Г., Волинець, В. І., Ніколюк, П. К., Ніколюк, П. П. (2017). Автоматизована система керування рухом транспортних засобів в межах міста. *Вісник Харківського університету, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»,* 5(3). URL: <https://r.donnu.edu.ua/xmlui/handle/123456789/1072>

2. Богуто, Д. Г., Комаров, В. Ф., Ніколюк, П. К., Ніколюк, П. П. (2018). Інтелектуальний алгоритм управління міським трафіком. *Вісник Харківського університету, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, 38(46). URL: <https://r.donnu.edu.ua/xmlui/handle/123456789/1073>
3. Piezoelectric RoadTrax BL–International Road Dynamics, Inc. URL: www.irdinc.com/sensors/piezoelectric-roadtrax-bl.html.
4. Sobral, T., Galvao, T., Bornes, J. (2019). Visualization of urban mobility data from intelligent transportation systems. *Sensors*, 19(332). DOI: 10.3390/s19020332.
5. Liu, J., Liang, C.-W., Li, M., Jian, K., Qin, L., Liu, J.-C. (2017). Principle Research on a Novel Piezoelectric 12-DOF force/acceleration sensor. *Journal of Sensors*, 2017. DOI: 10.1155/2017/2836365.
6. Cai, Y., Liu, Z., Sun, X., Chen, L., Wang, H., Zhang, Y. (2017). Vehicle detection based on deep dual-vehicle deformable part models. *Journal of Sensors*, 2017. DOI: 10.1155/2017/5627281.
7. Tan, J., Shi, X., Li, Z., Yang, K., Xie, N., Yu, H., Wang, L., Li, Z. (2017). Continuous and discrete-time optimal controls for an isolated signalized intersection. *Journal of Sensors*, 2017. DOI: 10.1155/2017/6290248.
8. Ioslovich, I., Haddad, J., Gutman, P.-O., Mahalel, D. (2011). Optimal traffic control synthesis for an isolated intersection. *Control Engineering Practice*, 19(8), 900–911. DOI: 10.1016/j.conengprac.2011.05.004.
9. Wang, P.-W., Yu, H.-B., Xiao, L., Wang, L. (2017). Online traffic condition evaluation method for connected vehicles based on multisource data fusion, *Journal of Sensors*. 2017. DOI: 10.1155/2017/7248189.
10. Quddus, M., & Washington, S. (2015). Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, 55, 328–339. DOI: 10.1016/j.trc.2015.02.017.
11. Dabiri, S., Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, 86, 360–371. DOI: 10.1016/j.trc.2017.11.021.
12. Wang, F., Chen, C. (2018). On data processing required to derive mobility patterns from passively-generated mobile phone data. *Transportation Research Part C: Emerging Technologies*, 87, 58–74. DOI: 10.1016/j.trc.2017.12.003.
13. Ma, X., Yu, H., Wang, Y., Wang, Y., Gomez-Gardenes, J. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS ONE*, 10(3). DOI: 10.1371/journal.pone.0119044.
14. Hashemi, M., Karimi, H. A. (2016). A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 20(6), 673–590. DOI: 10.1080/15472450.2016.1166058.
15. Rahmani, M., Koutsopoulos, H. N. (2014). Floating car and camera data fusion for non-parametric route travel time estimation. *Procedia Computer Science*, 37, 390–395. DOI: 10.1016/j.procs.2014.08.058.

16. Oshyaniv, M. F., Sundberg, M., Karlström, A. (2014). Consistently estimating link speed using sparse GPS data with measured errors. *Procedia – Social and Behavioral Sciences*, 111(2), 829–838. DOI: 10.1016/j.sbspro.2014.01.117.
17. Wang, S., Jiang, X., Li, H., Hu, Z., Xu, C. (2023). Traffic safety assessment with integrated communication system of connected and automated vehicles at signalized intersections. *Transportation Research Record*. DOI: 10.1177/03611981231201107.

Рукопис отримано – 31/05/2023; прийнято до публікації – 26/06/2023.

© Донецький національний університет імені Василя Стуса, 2023
© Петро Ніколюк, 2023

The impact of intelligent traffic signal control on the capacity of an urban-controlled intersection

Petro Nikolyyuk

Abstract

Urban transportation networks operate in different modes. Knowing these modes and the transitions between them is an important factor in determining the efficiency of the urban transportation network. Overloading this network leads to traffic jams. For several decades, urban traffic has been facing the problem of congestion. This problem is particularly acute in large metropolitan cities, where the number of vehicles is growing rapidly and the capacity of transport networks remains virtually unchanged. A key element and the main cause of traffic jams is the intersection of city roads, where traffic flows from different directions converge. The first step in urban traffic management technology is the organization of traffic lights. Usually, the duration of the green and red light phases is constant, which does not take into account the current traffic congestion in different directions. In this study, we are talking about intersections with intelligent traffic control. This type of control is fundamentally different from standard traffic light control. Intelligent traffic control of vehicles passing through a separately controlled intersection is carried out using a computer program that takes into account the traffic congestion at the intersection and makes appropriate adjustments to the length of the red and green light phases to maximize throughput. The proposed procedure will optimize the throughput of each individual intersection, which will eventually lead to the transition of urban traffic to a qualitatively new level. At the same time, the number of traffic jams on city highways will be drastically reduced. The system operates in real time and therefore dynamically adjusts the phases of the traffic light, synchronized with the load of different directions at the intersection. The duration of each car's journey along the selected route will be reduced, making the journey more comfortable and safe.

Keywords: intelligent intersection, piezoelectric crystal sensor, traffic light cycle, intersection capacity, urban intersection model, AnyLogic.

References

1. Bohuto, D. H., Volynets, V. I., Nikoliuk, P. K., Nikoliuk, P. P. (2017). Avtomatyzovana systema keruvannia rukhom transportnykh zasobiv v mezhakh mista. *Visnyk Kharkivskoho universytetu, seriia «Matematychni modeliuvannia. Informatsiini tekhnologii. Avtomatyzovani systemy upravlinnia»*, 5(3). URL: <https://r.donnu.edu.ua/xmlui/handle/123456789/1072>. [in Ukrainian].
2. Bohuto, D. H., Komarov, V. F., Nikoliuk, P. K., Nikoliuk, P. P. (2018). *Intelektualnyi alhorytm upravlinnia miskym trafikom. Visnyk Kharkivskoho universytetu, seriia «Matematychni modeliuvannia. Informatsiini tekhnologii. Avtomatyzovani systemy upravlinnia»*, 38(46). URL: <https://r.donnu.edu.ua/xmlui/handle/123456789/1073> [in Ukrainian].
3. Piezoelectric RoadTrax BL–International Road Dynamics, Inc. URL: www.irdinc.com/sensors/piezoelectric-roadtrax-bl.html.
4. Sobral, T., Galvao, T., Bornes, J. (2019). Visualization of urban mobility data from intelligent transportation systems. *Sensors*, 19(332). DOI: 10.3390/s19020332.
5. Liu, J., Liang, C.-W., Li, M., Jian, K., Qin, L., Liu, J.-C. (2017). Principle Research on a Novel Piezoelectric 12-DOF force/acceleration sensor. *Journal of Sensors*, 2017. DOI: 10.1155/2017/2836365.
6. Cai, Y., Liu, Z., Sun, X., Chen, L., Wang, H., Zhang, Y. (2017). Vehicle detection based on deep dual-vehicle deformable part models. *Journal of Sensors*, 2017. DOI: 10.1155/2017/5627281.
7. Tan, J., Shi, X., Li, Z., Yang, K., Xie, N., Yu, H., Wang, L., Li, Z. (2017). Continuous and discrete-time optimal controls for an isolated signalized intersection. *Journal of Sensors*, 2017. DOI: 10.1155/2017/6290248.
8. Ioslovich, I., Haddad, J., Gutman, P.-O., Mahalel, D. (2011). Optimal traffic control synthesis for an isolated intersection. *Control Engineering Practice*, 19(8), 900–911. DOI: 10.1016/j.conengprac.2011.05.004.

9. Wang, P.-W., Yu, H.-B., Xiao, L., Wang, L. (2017). Online traffic condition evaluation method for connected vehicles based on multisource data fusion, *Journal of Sensors*. 2017. DOI: 10.1155/2017/7248189.
10. Quddus, M., & Washington, S. (2015). Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, 55, 328–339. DOI: 10.1016/j.trc.2015.02.017.
11. Dabiri, S., Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, 86, 360–371. DOI: 10.1016/j.trc.2017.11.021.
12. Wang, F., Chen, C. (2018). On data processing required to derive mobility patterns from passively-generated mobile phone data. *Transportation Research Part C: Emerging Technologies*, 87, 58–74. DOI: 10.1016/j.trc.2017.12.003.
13. Ma, X., Yu, H., Wang, Yun., Wang, Y., Gomez-Gardenes, J. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS ONE*, 10(3). DOI: 10.1371/journal.pone.0119044.
14. Hashemi, M., Karimi, H. A. (2016). A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 20(6), 673–590. DOI: 10.1080/15472450.2016.1166058.
15. Rahmani, M., Koutsopoulos, H. N. (2014). Floating car and camera data fusion for non-parametric route travel time estimation. *Procedia Computer Science*, 37, 390–395. DOI: 10.1016/j.procs.2014.08.058.
16. Oshyaniv, M. F., Sundberg, M., Karlström, A. (2014). Consistently estimating link speed using sparse GPS data with measured errors. *Procedia – Social and Behavioral Sciences*, 111(2), 829-838. DOI: 10.1016/j.sbspro.2014.01.117.
17. Wang, S., Jiang, X., Li, H., Hu, Z., Xu, C. (2023). Traffic safety assessment with integrated communication system of connected and automated vehicles at signalized intersections. *Transportation Research Record*. DOI: 10.1177/03611981231201107.

