

УДК 004.94

Ю.В. Ульяновська, О.Д. Фірсов, М.Ф. Мормуль, Д.О. Пікулін

ОБ'ЄКТНО-ОРІЄНТОВАНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ДОРОЖНЬОГО РУХУ

Університет митної справи та фінансів, Дніпро

Анотація. Робота присвячена розробці веб-застосунку для імітаційного моделювання міського дорожнього руху на перехресті. Оскільки на даний час транспортна система не задовольняє на 100% попит через свою моральну застарілість та існування багатьох факторів, що впливають на неї, і які людині важко постійно враховувати, актуальним завданням є автоматизація процесу моделювання транспортних потоків шляхом розробки програмного забезпечення на основі сучасних веб-технологій. У роботі проаналізовано класи первісних та потрібних даних, логічні змінні, класи необхідних функцій. Розроблено програму на мові програмування JavaScript, зроблено експериментальні розрахунки. У підсумку розроблено веб-застосунок, який спрощує дослідження транспортних ділянок завдяки імітаційному моделюванню, що дозволяє удосконалити моніторинг дорожніх перехресть та підвищити їх пропускну здатність. Веб-застосунок має простий інтуїтивно-розумний інтерфейс та робить перевірку коректності введених даних, що дозволяє працювати з ним людям з мінімальними навичками і без спеціальних знань. Програмне забезпечення було розроблено адаптивно, що дозволяє працювати з ним з будь-якого пристрою, який має встановлений веб-браузер. Завдяки розробці одного головного файлу на мові програмування JavaScript, його легко інтегрувати та використовувати в готових рішеннях на різних мовах як у веб-додатках, так і у програмному забезпеченні, наприклад у C# чи C++. Це дозволяє впровадити використання програмного засобу з мінімальними витратами коштів та часу. За рахунок застосування розробленої системи імітаційного моделювання дорожнього руху на перехресті буде значно підвищено ефективність функціонування регульованих перехресть зі світлофорним типом регулювання.

Ключові слова: транспортна мережа, імітаційне моделювання, веб-застосунок.

Abstract. Work is devoted to the development of web-application for the imitation design of municipal travelling motion on crossing. As on this time a transport system dissatisfies on 100% of demand through the moral ramshackleness and existence of many factors that influence on her, and that it is difficult constantly to take into account a man, an actual task is automation of process of design of transport streams by software development on the basis of modern web technologies. The classes of primary and necessary data are in-process analyzed, boolean variables, classes of necessary functions. The program is worked out in programming of JavaScript language, experimental calculations are done. In the total it is worked out web-appendix, that simplifies research of transport areas due to an imitation design, that allows to perfect monitoring of the travelling crossing and promote them carrying capacity. web-appendix has a simple intuitively-clear interface and does verification of correctness of the entered data, that allows to work with him to the people with minimum skills and without the special knowledge. Software was worked out adaptive, that allows to work with him from any device that has the set web-browser. Due to development of one main file in programming of JavaScript language, him it easily to integrate and use in the prepared decisions on different languages both in web-additions and in programmatic provided, for example in C# or C++. It allows to inculcate the use of programmatic means with the minimum charges of money and time. Due to application of the worked out system of imitation design of travelling motion on crossing efficiency of functioning of the managed crossing will considerably increase with a traffic-light type adjusting.

Key words: transport network, simulation modeling, web application.

DOI: <https://doi.org/10.31649/1999-9941-2023-57-2-78-86>.

Вступ

Через особливості транспортних систем неможливо створити адекватну аналітику для програмного забезпечення, яка досліджувала б можливі варіанти управління в цій системі та її характеристику в різних умовах. У той же час програмне забезпечення імітаційного моделювання як метод вивчення подібних об'єктів є адекватним підходом для вирішення цієї проблеми, оскільки швидко та з високою точністю прогнозує характеристики складних систем подібної природи та оптимізує суттєві параметри, вибираючи відповідні параметри оптимізації. З цієї причини у дорожньому русі існує складна технічна система, що визначає його специфіку як об'єкт управління. Проте, навіть розглядаючи лише технічні аспекти управління дорожнім транспортом, необхідно враховувати, що цей об'єкт дуже своєрідний і складний з точки зору управління його властивостями. Транспортна система України характеризується значними обсягами перевезень вантажів залізничним та автомобільним транспортом. При цьому на вітчизняному ринку існує ряд прикладних інформаційних систем та програмних продуктів, що мають широкий функціонал та позитивну практику використання для вирішення задач організації перевезень та оптимізації транспортного процесу [1]. В Національній транспортній стратегії України з метою створення ефективно працюючого транспортного комплексу України та, як наслідок, досягнення Україною статусу регіонального транспортного «хабу», враховані наступні світові тенденції, що притаманні транспорту: висока технологічна складність транспортних засобів та ергономічність, супутникова навігація, інтелектуальні транспортні системи, інформаційні технології, тощо [2]. Питання інформатизації транспортної галузі є актуальним не тільки для України. Так в роботі [3] розглядається вплив цифрових технологій з економічного погляду. Дослідження зосереджене лише на макроекономічних показниках Словаччини такого сектора національної економіки як транспортна галузь. Застосування інтернет технологій в

транспортній галузі розглядається в роботі [4]. Транспортна галузь, особливо в останні роки, з появою Інтернету речей, створює можливості для розвитку. Технологія IoT може значно підвищити безпеку та надійність транспортних засобів і вантажів під час транспортування. Розвиток транспортної галузі може не лише підвищити рівень зайнятості та стимулювати економічне зростання, а й скорегувати економічну та промислову структуру та сприяти постійному прискоренню інтеграційних процесів між містом та регіонами. Можливість використання засобів автоматизації імітаційного моделювання для залізничної мережі розглядається в роботі [5], де автори описують методіку вирішення типових завдань, що дозволяють визначити найбільш ефективний варіант обслуговування транспортних потоків. Використання інтелектуальних систем моделювання транспортних потоків та аналізу сучасних програмних продуктів для моделювання транспортних потоків присвячена робота [6]. Моделювання транспортних потоків було здійснено з використанням власної програми «Free way», яка дозволяє відслідковувати завантаженість доріг України на основі аналізу динаміки швидкості на обраних ділянках доріг. Моделювання здійснювалось з застосуванням програмного продукту, що реалізований з використанням системи управління «CMS Drupal», заснований на мові програмування PHP, мови програмування javascript, елементів HTML, та CSS.

Загальну методологію імітаційного моделювання та вдосконалено формалізацію методу агентного моделювання створено в роботі [7]. Результати роботи покладені в основу системи імітаційного моделювання транспортних потоків, що дозволяє аналізувати властивості наявних і проєктованих транспортних вузлів. Система реалізована у вигляді програмного комплексу, який може бути використаний в установах державного управління, проєктних організаціях і консалтингових компаніях, що займаються проєктуванням і реорганізацією схем дорожнього руху. Запропонована модель агента може бути використана для більш складних імітаційних моделей організаційно-технічних систем.

Виходячи з вище сказано можна зробити висновок, що впровадження інформаційних систем в моделювання та регулювання транспортних потоків є важливим практичним завданням. В той же час можливість застосування веб-застосунків з можливостями моделювання транспортних потоків не досліджувалась. (Стиль @Основний текст)

Актуальність

Транспортна галузь є одним з важливіших елементів економіки, але на даний час вона не задовольняє на 100% попит, через те, що вона морально застаріла, бо в ній закладені принципи, які не відповідають сучасним стандартам та вимогам. При розгляді проблем транспортної галузі треба враховувати велику кількість факторів, які впливають на неї.

Значну роль у забезпеченні безпечних умов пересування грають спеціальні технічні засоби регулювання: дорожні знаки (знаки пріоритету та обмеження), світлофори, розмітка, що встановлюються на черговість, пріоритетність і допустимий напрямок руху транспортних засобів. У системі управління транспортними потоками об'єктом дослідження є транспортний процес, пов'язаний із дорожньою діяльністю у межах ділянки транспортної мережі міста, але водії автомобілів поведуться на дорозі і реагують на різні події по-різному, не завжди прогнозовано, що ускладнює аналіз такої системи. Програмне забезпечення для управління дорожнім рухом спрямоване на розробку заходів щодо вдосконалення організації дорожнього руху за рахунок підвищення її ефективності та раціональності використання засобів організації дорожнього руху, а також обґрунтування доцільності їх застосування.

Мета

Метою даної роботи є автоматизація процесу моделювання транспортних потоків шляхом розробки програмного забезпечення на основі сучасних веб технологій.

Задачі

1. Розробити веб-застосунок, за допомогою якого провести експерименти з моделями розподілення транспортних засобів.
2. Провести моделювання розподілення транспортних засобів за допомогою імовірнісних законів розподілу та імітаційного моделювання. Перша. (Стиль @Задачі дослідження)

Розв'язання задач

Для вирішення поставленої задачі було розроблено єдиний клас «Crossroads», в якому виконуються усі задачі. Завдяки тому, що розроблено єдиний клас, готове рішення легко інтегрувати та впроваджувати у вже використовувані системи. Для створення об'єкту класу потрібно передати HTML елемент «canvas» та необов'язково об'єкт налаштувань, який в свою чергу може мати наступні параметри:

- w – ширина;
- h – висота;
- bg_color – фоновий колір рядком у одному з форматів: HEX, RGB, RGBA;
- bg_image – посилання на фонове зображення рядком;

- roads_color – колір доріг рядком у одному з форматів: HEX, RGB, RGBA;
 - roads_lines_number – кількість смуг доріг;
 - roads_lanes_size – ширина смуги дороги;
 - roads_median_color – колір розділення доріг рядком у одному з форматів: HEX, RGB, RGBA;
 - roads_dividing_color – колір розділення доріг рядком у одному з форматів: HEX, RGB, RGBA;
 - roads_sidewalk_color – колір тротуару доріг рядком у одному з форматів: HEX, RGB, RGBA;
 - roads_sidewalk_width – ширина тротуару доріг рядком у одному з форматів: HEX, RGB, RGBA;
 - crosswalk_color – колір розмітки перехрестя дороги рядком у одному з форматів: HEX, RGB, RGBA;
- RGBA;
- roads – масив доріг який в свою чергу складається з масивів параметрів доріг:
 - x – координата початку дороги по осі X;
 - y – координата початку дороги по осі Y;
 - vertical – вертикальна дорога чи ні, логічний параметр;
 - cars_delay_y – максимальна затримка генерації автомобілів по осі Y, обов'язково для вертикальних доріг;
 - cars_delay_my – максимальна затримка генерації автомобілів по осі -Y, обов'язково для вертикальних доріг;
 - cars_delay_x – максимальна затримка генерації автомобілів по осі X, обов'язково для горизонтальних доріг;
 - cars_delay_mx – максимальна затримка генерації автомобілів по осі -X, обов'язково для горизонтальних доріг;
 - cars_count_y – кількість генерованих автомобілів по осі Y, обов'язково для вертикальних доріг;
 - cars_count_my – кількість генерованих автомобілів по осі -Y, обов'язково для вертикальних доріг;
 - cars_count_x – кількість генерованих автомобілів по осі X, обов'язково для горизонтальних доріг;
 - cars_count_mx – кількість генерованих автомобілів по осі -X, обов'язково для горизонтальних доріг.

Клас має публічні та не публічні змінні. Публічні:

- canvas – HTML елемент згідно з назвою змінної;
- ctx – об'єкт CanvasRenderingContext2d.

Непублічні змінні класу:

- #settings – об'єкт в якому зберігаються налаштування;
- #roads – масив об'єктів доріг;
- #crossroads – масив об'єктів перехресть доріг;
- #cars – масив об'єктів автомобілів;
- #cars_created_count – кількість створених автомобілів;
- #cars_count – загальна кількість автомобілів;
- #cars_created_exit – логічна змінна, чи закінчилось створення автомобілів;
- #currentSignalY – логічна змінна, для переключення світлофорів на перехресті дороги;
- #signal_time – час переключення світлофорів на перехресті дороги;
- #exited – логічна змінна, чи закінчилось виконання робіт;
- #uniform – логічна змінна, чи рівномірне розподілення автомобілів;
- #time – час виконання робіт;
- #bg_img – об'єкт зображення для фону;
- #linear_up – логічна змінна, чи спадне розподілення автомобілів;
- #linear_down – логічна змінна, чи зростаюче розподілення автомобілів;
- #custom – об'єкт розподілу автомобілів, якщо задано вручну.

При створенні об'єкту класу перевіряється чи було передано не пустий об'єкт налаштувань за допомогою функції keys() з класу Object, якщо передано не пустий, то за допомогою функції assign() з класу Object об'єкт об'єднується з налаштуваннями класу. Після чого викликається функція #setFrame(), яка в свою чергу встановлює кадрову частоту. Далі отримується об'єкт класу CanvasRenderingContext2D у змінну ctx. Викликається функція #setSize() для встановлення розмірів canvas. Далі перевіряються налаштування доріг, якщо їх немає, то задаються стандартні 2 дороги. До об'єкту класу CanvasRenderingContext2D додається функція для швидкого округлення при рисуванні за допомогою функції #setCtxRoundedRect(). Далі викликаються функції #createRoads() та #createCrossroads(). Наприкінці перевіряється налаштування фонового зображення, якщо його не має – то викликається функція #drawScene(), якщо є – то створюється об'єкт зображення, при завантаженні якого вже викликається функція #drawScene().

`#createRoads()` – функція створює об'єкти доріг з налаштувань доріг, при цьому вона використовує функцію `forEach()` з класу `Array` для проходження по всім налаштуванням доріг по черзі.

`#createCrossroads()` – функція створює об'єкти перехрестя доріг після чого викликає функцію `#updSignals()`.

`#drawScene()` – функція задає фон для елемента `canvas` за допомогою класу `CanvasRenderingContext2D`, після чого в циклі викликає рисування кожної дороги у функції `#drawRoad()` та таким же чином рисування кожного перехрестя доріг у функції `#drawCrossroad()`. Щоб пройти по кожному елементу дороги та перехрестя, використовуються функція `forEach()` з класу `Array`.

`#updSignals()` – функція задає параметри для світлофорів у об'єктах перехрестя згідно з значенням змінної `#currentSignalY`. Також при увімкненні червоного світла світлофора до масиву черг перехрестя додається пустий масив за допомогою функції `push()` з класу `Array`, в подальшому в цей масив додаються автомобілі які чекають увімкнення зеленого кольору світлофора.

`#drawRoad()` – функція рисує дорогу на елементі `canvas` за допомогою класу `CanvasRenderingContext2D`.

`#drawCrossroad()` – функція рисує перехрестя доріг на елементі `canvas` використовуючи клас `CanvasRenderingContext2D`.

Деякі дані можливо задати після створення об'єкту класу, для цього є наступні функції:

`setSignalTime(time)` – функція для задання часу переключення світлофорів;

`setRoadCarsDelay(road, axis, value)` – функція для задання максимальної затримки генерації автомобілів на дорогах, `road` дорівнює порядковому номеру дороги, `axis` відповідає осі, на якій потрібно задання, `value` дорівнює значенню максимальної затримки;

`setRoadCars(road, axis, value)` – функція для задання кількості автомобілів на дорогах, `road` дорівнює порядковому номеру дороги, `axis` відповідає осі, на якій потрібно задання, `value` дорівнює значенню максимальної затримки;

`setUniform()` – функція для задання рівномірної генерації автомобілів;

`setLinearUp()` – функція для задання спадної генерації автомобілів;

`setLinearDown()` – функція для задання зростаючої генерації автомобілів;

`setUniform()` – функція для задання ручної генерації автомобілів.

Для отримання даних є наступні функції:

`getRoads()` – функція повертає масив об'єктів доріг;

`getTime()` – функція повертає часи виконання робіт;

`exited()` – функція повертає значення чи працює в даний момент клас.

Щоб почати виконання робіт з імітаційного моделювання дорожнього руху на перехресті, потрібно викликати статичну функцію `start()`, в якості змінної в неї потрібно передати об'єкт розробленого класу `Crossroads`. Данна функція задає значення поточного часу змінній `#time` за допомогою функції `now()` класу `performance` та задає значення істинності (хибності) в логічну змінну `#exited`. Після чого запускаю по черзі наступні функції: `#drawCars()`, `#drawScene()`, `#signal()`, `#tick()`.

Функція `#drawCars()` проходить по дорогам з метою створення та рисування автомобілів. Створюються автомобілі за допомогою функції `#createCar()` та після створення записуються до масиву автомобілів `#cars`, сам автомобіль є об'єктом даних. Рисування автомобілів виконується за допомогою функції `#drawCar()`, яка для цього використовує клас `CanvasRenderingContext2D`. Також перед створенням автомобілів використовується функція `#roadSetCarsDelay()`, яка створює дані затримок генерації автомобілів, при цьому вона використовує функції класу `Math` для обчислень, функцію `push()` класу `Array` для додавання даних до масивів, функцію `entries()` класу `Object` для обходу даних об'єктів.

`#signal()` – це функція для зміни значення змінної `#currentSignalY` на протилежне у заданий період часу та при цьому викликається функція `#updSignal()`. Потрібно це для переключення кольору світлофорів у заданий час.

Функція `#tick()` викликає функції `#drawScene()` та `#driveCars()`. Далі якщо змінна `#exited` має значення істини, що каже про потребу закінчення роботи, викликається функція `#exitSignals()` та `#drawScene()`, після чого у змінну `#time` записується значення обчислення часу виконання роботи класу за допомогою функції `now()` з класу `performance` та викликається функція `#triggerEvent()`. Якщо ж змінна `#exited` має значення хибності, то виклик функції `#tick()` встановлюється на оновлення кадру.

Функція `#driveCars()` обходить всі автомобілі з цілю їх переміщення та рисування за допомогою функції `#drawCar()`. При роботі використовується функція `#carDistanceCheck()` для перевірки відстані між попереднім авто, якщо є попереду авто та воно не рухається – використовується функція `#carCheckCrossroadStop()` для перевірки, чи попередне авто стоїть на перехресті через червоне світло на ньому, при цьому якщо це так, то авто додається до черги на дорозі. Також перевіряється, чи зайнята зліва полоса іншим автомобілем, якщо не зайнята – то автомобіль переходить до II смуги дороги. Якщо попереду автомобіля не має, то перевіряються перехрестя – чи не підїхав автомобіль до них; коли

це так – перевіряється колір світлофора перехрестя, від значення якого встановлюється, як автомобіль буде рухатись далі. При обході перехрестя використовується функція `includes()` класу `Array` для перевірки чи є автомобіль в масиві перехрестя з автомобілями які рухаються на ньому, якщо є то за допомогою функції `indexOf()` класу `Array` вираховується порядковий номер автомобіля у масиві та за допомогою цього видаляється функцією `splice()` класу `Array`. Коли автомобіль рухається по перехрестю, то його номер заноситься до масиву руху на перехресті за допомогою функції `push()` класу `Array`, а коли автомобіль зупинився через червоний колір світлофора на перехресті, використовуються функції цього ж класу для запису автомобіль у масив черги на дорозі. В кінці своєї роботи перевіряється кількість автомобілів, які покинули межі побудованої дороги, якщо вона співпадає з загальною кількістю автомобілів та при цьому згенеровані всі автомобілі, змінній `#exited` присвоюється значення істини.

За допомогою функції `#exitSignals()` на дорогах в масивах черг автомобіль, замість масивів з автомобілями, записується їх кількість.

`#triggerEvent()` дана функція викликає подію в залежності від браузера. На цю подію інші розробники можуть додавати виклик своїх функцій, що додає гнучкості використанню розробленого класу.

Аналіз результатів роботи моделей

Розроблений веб-додаток має простий та інтуїтивно зрозумілий інтерфейс. В правій більшій частині відображається побудована карта перехрестя для візуалізації імітаційного моделювання дорожнього руху. В лівій частині знаходиться інтерфейс для встановлення налаштувань, скинення налаштувань на за замовчуванням та запуску імітаційного моделювання. Після запуску у правій частині відображається рух авто та робота перехрестя дороги (рис. 1).



Рисунок 1 – Веб-додаток з відображенням процесу імітаційного моделювання дорожнього руху на перехресті

В результатах роботи (рис. 2) відображається наступна час виконання, максимальна довжина черги по x , максимальна довжина черги по tx , максимальна довжина черги по y , максимальна довжина черги по ty , графік довжини черги по x , графік довжини черги по tx , графік довжини черги по y , графік довжини черги по ty , графік розподілу автомобілей по x , графік розподілу автомобілей по tx , графік розподілу автомобілей по y , графік розподілу автомобілей по ty .

Моделювання в розробленому програмному застосунку можна виконати одним з п'яти методів розподілу (рис. 3). При цьому використовувались наступні параметри які, за потреби, можливо змінювати: максимальна затримка авто X (в мс); максимальна затримка авто $-X$ (в мс); максимальна затримка авто Y (в мс); максимальна затримка авто $-Y$ (в мс). Було проведено 12 експериментів з рівномірним розподілом, при цьому використовувались наступні налаштування: кількість авто $X = 200$; максимальна затримка авто X (в мс) = 1000; кількість авто $-X = 200$; максимальна затримка авто $-X$ (в мс) = 1000; кількість авто $Y = 200$; максимальна затримка авто Y (в мс) = 1000; кількість авто $-Y = 200$; максимальна затримка авто $-Y$ (в мс) = 1000; час переключення світлофора = 3000; рівномірний розподіл авто = включено.

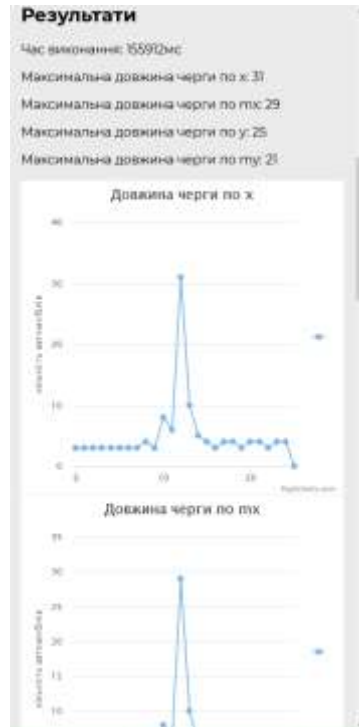


Рисунок 2 – Веб-додаток з відображенням процесу імітаційного моделювання дорожнього руху на перехресті



Рисунок 3 – Вигляд веб-додатку з мобільного пристрою а) з відображенням карти для імітаційного моделювання, б) настройка параметрів

Розглянемо основні результати експериментів за зазначеними розподілами.

Рівномірний розподіл.

Приклад результату проведення експериментів наведено у вигляді графіку (рис. 4).

Нормальний розподіл, при якому мінімальна довжина черги склала 23, а максимальна – 39. У розробленому веб-додатку за замовчуванням використовується нормальний розподіл.

Результати проведення експериментів наведено нижче у вигляді графіків (рис. 5). В наведеному графіку відображається довжина черги автомобілів по усім 4 напрямкам їх руху.



Рисунок 4 – Графік довжини черги автомобілей при рівномурно розподілу у експерименті №1

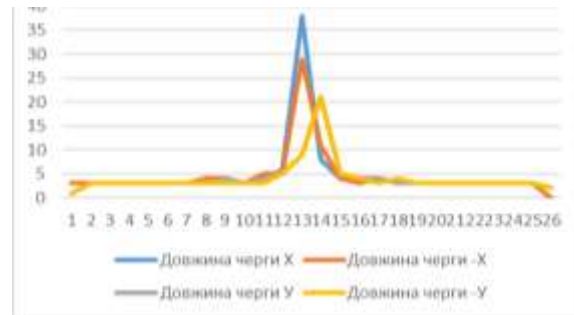


Рисунок 5 – Графік довжини черги автомобілей при нормальному розподілу у експерименті №1

Емпірична модель функціонування.

Спадний розподіл, в якому під час експериментів дані між ними співпадають. Результати проведення експериментів наведено нижче у вигляді графіків (рис. 6).

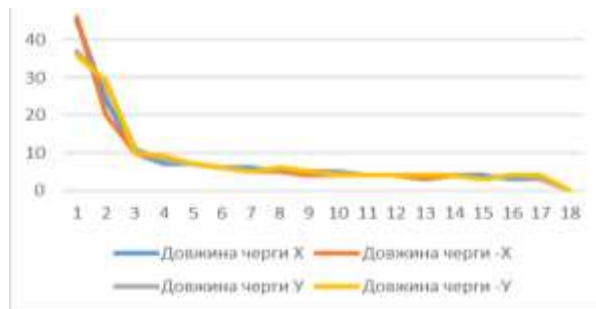


Рисунок 6 – Графік довжини черги автомобілей при спадаючому розподілу у експерименті №1

Зростаючий розподіл.

За даним розподілом було проведено 6 експериментів, при цьому використовувались налаштування як для спадаючого розподілу. Результати проведення експериментів наведено нижче у вигляді графіків (рис. 7).

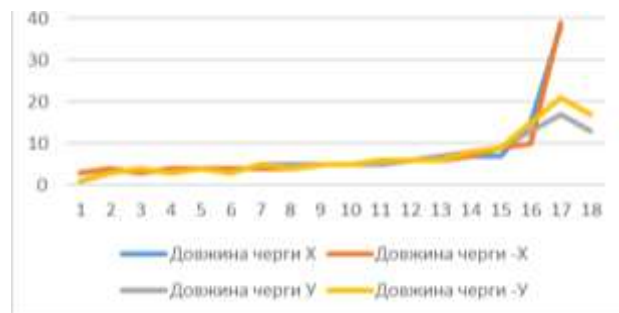


Рисунок 7 – Графік довжини черги автомобілей при наростаючому розподілу у експерименті №1

Моделювання з використанням реальних даних.

В веб-додатку було передбачено моделювання розподілу автомобілей на зібраних раніше фактичних даних. Аналогічно з попередніми розподілами було проведено 6 експериментів з використанням реальних даних з перехрестя. Приклад результату проведення експериментів наведено нижче у вигляді графіку (рис. 8).



Рисунок 8 – Графік довжини черги автомобілей при наростаючому розподілу у експерименті №1

Згідно графіків чітко видно, що рівномірний розподіл до 3 разів швидший порівняно з нормальним розподілом, але на жаль у житті неможливо розподіляти автомобілі і, як правило, вони розподіляються за нормальним розподілом, через що утворюються затори.

За рахунок заборони повороту автомобілів наліво можливо збільшити пропускну здатність перехрестя до 2 разів. Також за рахунок цього зменшиться кількість ДТП на перехрестях, тому що автомобілі рухатимуться тільки прямо або направо, де немає перешкоди для повороту. На даний час кількість ДТП на перехрестях майже в 8 разів більше ніж на прямій дорозі. Отже можливо припустити, що введення такої заборони зменшить ДТП на перехрестях з 83% до 11%, тобто загальна кількість ДТП зменшиться в теорії на 72%. При цьому отримуємо майже в 2 рази більшу пропускну здатність перехрестя.

Висновки

1. Розроблено веб-застосунок, який дозволяє виконувати імітаційне моделювання дорожнього руху на перехресті з метою дослідження та підвищення пропускну здатності. Він має простий інтуїтивно-зрозумілий інтерфейс та перевірку коректності введених даних, що дозволяє працювати з ним людям з мінімальними навичками та впровадити використання програмного засобу з мінімальними витратами коштів та часу.

2. Проведені дослідження щодо підвищення ефективності функціонування регульованих перехрестя зі світлофорним типом регулювання за рахунок застосування розробленої системи імітаційного моделювання дорожнього руху на перехресті.

Список літератури

- [1] Є. Б. Демченко, А. С. Дорош, І. Я. Сковрон, «Сучасні інформаційні системи на ринку вантажних перевезень України,» *Транспортні системи та технології перевезень*, вип. 23, с.79–88, 2022.
- [2] Про схвалення Національної транспортної стратегії України на період до 2030 року: Розпорядження Кабінету Міністрів України від 30.05.2018 № 430-р., *База даних «Законодавство України»*. [Online]. Available: <https://zakon.rada.gov.ua/go/430-2018-%D1%80>. [Accessed: Sept. 1, 2022].
- [3] R. Chinoracky, J. Kurotova, P. Janoskova, «Measuring the impact of digital technologies on transport industry – macroeconomic perspective,» *Transportation Research Procedia*, Vol. 55, pp. 434-441, 2021. [Online]. Available: <https://doi.org/10.1016/j.trpro.2021.07.092>. [Accessed: Sept. 30, 2022].
- [4] A. Nurpeisovaa, T. Blokhinab, A. Zuevac, S. Kievskayac, V. Varzin, «Infrastructure, technology and financial engineering in the transport services market,» *X International Scientific Siberian Transport Forum. Transportation Research Procedia*, вип. 63, pp.1396–1403, 2022. [Online]. Available: <https://doi.org/10.1016/j.trpro.2022.06.150>. [Accessed: Sept. 12, 2022].
- [5] Е.И. Сукач, «Автоматизация процесса исследования вариантов организации перемещения транспортных потоков в железнодорожной сети,» *Математичні машини і системи*, с.161-168, №4 2009. [Online]. Available: <http://dspace.nbuv.gov.ua/handle/123456789/47333>. [Accessed: Sept. 22, 2022].

- [6] Н. В. Халіпова, М. О. Сівак, «Аналіз та моделювання обслуговування транспортних потоків в міжнародному пункті пропуску автомобільного транспорту "Тиса",» *Системи та технології*, с.5-21, № 1 2017.
- [7] О. В. Горбова, О. Д. Мерзлий, «Дослідження автомобільних потоків засобами імітаційного моделювання,» *Наука та прогрес транспорту. Вісник Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна*, с.36–45, № 5(95) 2021.

Стаття надійшла: 30.04.2023

References

- [1] E. B. Demchenko, A. S. Dorosh, I. Ya. Skovron, «Suchasni informatsiini systemy na rynku vantazhnykh perevezen Ukrainy,» *Transportni systemy ta tekhnologii perevezen* («Modern information systems in the cargo transportation market of Ukraine,» *Transport systems and transportation technologies*), vyp. 23, s.79–88, 2022. [in Ukraine].
- [2] Pro skhvalennia Natsionalnoi transportnoi stratehii Ukrainy na period do 2030 roku: Rozporiadzhennia Kabinetu Ministriv Ukrainy vid 30.05.2018 № 430-r., *Baza danykh «Zakonodavstvo Ukrainy»* (On the approval of the National Transport Strategy of Ukraine for the period until 2030: Decree of the Cabinet of Ministers of Ukraine dated 30.05.2018 No. 430, *Database "Legislation of Ukraine"*). [Online]. Available: <https://zakon.rada.gov.ua/go/430-2018-%D1%80>. [Accessed: Sept. 1, 2022]. [in Ukraine].
- [3] R. Chinoracky, J. Kurotova, P. Janoskova, «Measuring the impact of digital technologies on transport industry – macroeconomic perspective,» *Transportation Research Procedia*, Vol. 55, pp. 434-441, 2021. [Online]. Available: <https://doi.org/10.1016/j.trpro.2021.07.092>. [Accessed: Sept. 30, 2022]. [in English].
- [4] A. Nurpeisovaa, T. Blokhinab, A. Zuevac, S. Kievskayac, V. Varzin, «Infrastructure, technology and financial engineering in the transport services market,» *X International Scientific Siberian Transport Forum. Transportation Research Procedia*, vyp. 63, rr.1396–1403, 2022. [Online]. Available: <https://doi.org/10.1016/j.trpro.2022.06.150>. [Accessed: Sept. 12, 2022]. [in English].
- [5] E.Y. Sukach, «Avtomatyzatsiya protsessa yssledovanyia varyantov orhanyzatsyyi peremeshchenyia transportnykh potokov v zheleznodorozhnoi sety,» *Matematychni mashyny i systemy* («Automation of the research process of options for the organization of movement of transport flows in the railway network,» *Mathematical machines and systems*), s.161-168, №4 2009. [Online]. Available: <http://dspace.nbuv.gov.ua/handle/123456789/47333>. [Accessed: Sept. 22, 2022]. [in Ukraine].
- [6] N. V. Khalipova, M. O. Sivak, «Analiz ta modeliuвання obsluhovuvannya transportnykh potokiv v mizhnarodnomu punkti propusku avtomobilnoho transportu "Tysa",» *Systemy ta tekhnologii* («Analysis and modeling of traffic flow maintenance at the international road transport checkpoint "Tysa",» *Systems and technologies*), s.5-21, № 1 2017. [in Ukraine].
- [7] O. V. Horbova, O. D. Merzlyi, «Doslidzhennia avtomobilnykh potokiv zasobamy imitatsiinoho modeliuвання,» *Nauka ta prohres transportu. Visnyk Dniprovskoho natsionalnoho univer-sytetu zaliznychnoho transportu imeni akademika V. Lazariana* ("Study of traffic flows by means of simulation modeling," *Science and Transport Progress. Bulletin of the Dnipro National University of Railway Transport named after Academician V. Lazaryan*), s.36–45, № 5(95) 2021. [in Ukraine].

Відомості про авторів

Ульяновська Юлія Вікторівна – кандидат технічних наук, доцент, завідувач кафедри

Фірсов Олександр Дмитрович – кандидат фізико-математичних наук, доцент, доцент

Мормуль Микола Федорович – кандидат технічних наук, доцент, доцент

Пікулін Денис Олегович – магістр

Yu.V. Ulianovska, A.D. Firsov, M.F. Mormul, D.O. Pikulin

OBJECT-ORIENTED IMPLEMENTATION OF A WEB APPLICATION FOR TRAFFIC SIMULATION

University of Customs and Finance, Dnipro