

Д.С.Федоряка
Комп'ютерне моделювання
у застосуванні до міського автомобільного руху

Ключові слова: комп'ютерне моделювання, автомобільний рух у великому місті, автомобілі, дороги, будинки, перехрестя, оптимізація транспортних потоків, середовище програмування VisualBasic.NET 2010.

Keywords: computer simulation, automobile traffic in bigcities, automobiles, roads, buildings, crosses, transport streams optimization, environment VisualBasic.NET 2010.

Ключевые слова: компьютерное моделирование, автомобильное движение в большом городе, автомобили, дороги, здания, перекрёстки, оптимизация транспортных потоков, среда программирования VisualBasic.NET 2010.

Розроблено модель руху транспорту в великому місті. Враховуються: 1) тип руху по вулицях (одно-чи двосторонній), 2) наявність перехресть – регульованих або нерегульованих, 3) основні вимоги безпеки руху. Створено систему підтримки прийняття рішень для оптимізації транспортного руху. Для комп'ютерної реалізації моделі створено проект із використанням засобів об'єктно-орієнтованого програмування. Проект виконано в середовищі програмування VisualBasic.NET 2010.

Транспортна система великих міст не витримує постійно зростаючого потоку машин. Виникають затори. зростають забруднення довкілля, витрати енергоресурсів та часу пасажирів.

Цю проблему можна розв'язати, розбудовуючи та реорганізуючи транспортну систему, для чого потрібні значні кошти та час. Важливо уявляти наслідки цих змін. При проектуванні нових міст треба передбачити можливі слабкі місця їх транспортної системи.

Отже, проблеми оптимізації міського руху є надзвичайно актуальними для великих міст. Моделювання є ефективним засобом вирішення цієї проблеми.

Мета роботи та об'єкт дослідження

Транспортну систему міста зручно аналізувати методом комп'ютерного моделювання. У статті описано розробку та програмну реалізацію комп'ютерної моделі цієї системи. Мета роботи - розробка такої моделі, що відображує динаміку руху автотранспорту в широкому інтервалі кількості відображуваних об'єктів. Тут мова йде про: 1) автомобілі, які можуть виїжджати з підземних стоянок у будинках, рухатися з різними швидкостями та в'їжджати до підземних стоянок в інших будинках; 2) будинки з підземними автостоянками; 3) вулиці з одностороннім та двостороннім рухом і різними пріоритетами для проїзду нерегульованих перехресть; 4) перехрестя зі світлофорами та без них. Для всіх цих об'єктів слід забезпечити можливість зміни значень параметрів у розумних діапазонах. Такими параметрами є 1) для автомобілів - наприклад, розміри, швидкості та будинки, до яких вони прямують; 2) для підземних стоянок – кількості всіх місць та заповнених місць; 3) для вулиць – кількості смуг руху в обидва боки та кількість перехресть, крізь які вони проходять; 4) для світлофорів – час, протягом якого вони не перемикають світловий сигнал. Крім перелічених є ще параметри, які описують зв'язки між вказаними вище об'єктами.

Об'єктом дослідження в роботі виступає транспортна система, типова для великих міст (сукупність доріг, будинків та транспортних засобів).

Задачі роботи

1. Проаналізувати запропоновані іншими авторами методи моделювання міського автомобільного руху.

2. Розробити математичну модель та відповідні їй алгоритми для моделювання динаміки міського автомобільного руху.

3. Створити програмний продукт для моделювання. Визначити й реалізувати в ньому структури даних, які дозволять представити транспортну систему в динаміці зручними для сприйняття засобами.

4. Інтегрувати до нього інструменти для аналізу транспортної системи, що дозволить, таким чином, запропонувати систему підтримки прийняття рішень, метою яких є з'ясування можливостей оптимізація руху міського автомобільного транспорту.

5. Розробити технічну документацію до програмного продукту.

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА

1.1. Способи вирішення проблем транспортної системи

Найпростішим способом розв'язання проблем виникнення заторів, на перший погляд, є розширення пропускної здатності існуючих доріг. Але вздовж доріг розташовано будинки. До того ж, реконструкція доріг пов'язана з великими фінансовими витратами та тимчасовими незручностями, які погіршують транспортну ситуацію.

Інший спосіб боротьби з заторами – створення нових магістральних шляхів та об'їзних доріг. Це допомагає розвантажити центральні частини міст, де, як правило, дороги вузькі, від транзитних потоків, але вимагає великих витрат і пов'язане зі складною проблемою виділення територій під будівництво доріг.

Ще один спосіб боротьби із заторами – оптимізація трафіку за допомогою світлофорів, розмітки доріг, зміни напрямків транспортних потоків. Цей спосіб найдешевший, але потребує певних теоретичних розрахунків.

Крім цих способів, спрямованих безпосередньо на оптимізацію інфраструктури, для зменшення транспортних потоків відомі й інші (наприклад, створення мережі громадського транспорту, заохочення населення користуватися ним, заборона проїзду для певних видів транспорту тощо).

1.2. Використання комп'ютерного моделювання для оптимізації міського трафіку

Модель – це штучний об'єкт, який відображує важливі для її розробника властивості іншого природного або штучного об'єкту (оригіналу). Комп'ютерне моделювання – це процес створення та вдосконалення комп'ютерної програми, яка реалізує дану модель.

При реорганізації транспортних систем зазвичай проводять інженерні розрахунки, які базуються на даних відслідковування динаміки транспортних потоків. Але вони не можуть у повному обсязі передбачувати наслідки, тому що транспортний потік буде підлаштовуватись під зміни транспортної системи.

Моделювання є необхідним для прийняття рішень щодо оптимізації транспортної системи через наступні її властивості.

- Компенсація зростання пропускної здатності при розвитку мережі збільшенням попиту на неї.
- Непередбачуваність поведінки кожного водія.
- Вплив випадкових факторів (ДТП, погодні умови, свята тощо).

1.3. Способи моделювання транспортних потоків

Всі моделі транспортних потоків можна розбити на три класи [1].

- **Моделі-аналоги, або макроскопічні.** Рух транспортних засобів уподібнюється певному фізичному потоку та описується рівняннями гідро- або газодинаміки.
- **Моделі слідування за лідером, або мікроскопічні.** Окремо розглядається рух кожного автомобіля. Враховується, що водії узгоджують швидкість руху власної машини зі швидкістю руху автомобіля попереду.
- **Ймовірнісні моделі.** У них транспортний потік розглядається як стохастичний процес.

Модель, яка розглядається в даній роботі – мікроскопічна. Її перевага в тому, що вона найбільш точно описує поведінку транспортної системи в реальному часі (відлік часу в моделі співпадає з відліком часу в реальності). Крім того, вона є наочною.

РОЗДІЛ 2. ОПИС МОДЕЛІ ТА ПРОГРАМИ

Під час виконання даної роботи створено програмний продукт **VirtualCity**. Для цього використано середовище **Microsoft VisualStudio 2010**, мова програмування **VisualBasic .NET**.

2.1. Модель міста в програмі та її реалізація

Математична модель міста у програмі така. Воно відображується прямокутником певних розмірів. У деяких його точках розташовані перехрестя. Деякі з них сполучені дорогами. Дороги мають певну ширину – від нуля до шести смуг в обох напрямках (нуль смуг може бути лише в одному напрямку – така дорога одностороння). У місті є будинки, вони відображуються прямокутниками зі сторонами, паралельними осям координат.

Головним об'єктом дослідження у програмі є автомобіль. Він відображується прямокутником та прив'язується до міста координатами середини сторони з того боку, куди він рухається (далі називатимемо це «передньою точкою») та кутом повороту. Крім того, він може бути прив'язаний до дороги, на якій знаходиться, номером смуги та відстанню від початкового перехрестя. Автомобілі в програмі виїжджають з одних будинків та рухаються за наперед прорахованим маршрутом до інших.

Для кожного будинку відома його місткість (максимальна кількість автомобілів, що може в ньому знаходитись), кількість машин у ньому в даний момент часу та популярність (параметр, який визначає середню частоту в'їзду машин до даного будинку).

Описана модель реалізована в програмі за допомогою концепції об'єктно-орієнтованого програмування. Об'єктна система програми складається із шести класів:

- **City** – місто. В екземплярі класу зберігаються параметри міста та всі його компоненти (перехрестя, дороги, автомобілі та будинки). Методи класу забезпечують редагування, відображення та моделювання міста.
- **Road** – дорога. Клас забезпечує відображення доріг.
- **Intersection** – перехрестя. В екземплярі класу зберігаються координати перехрестя. Методи забезпечують проїзд машин через перехрестя.
- **TrafficControl** – світлофор. Клас забезпечує проїзд машин через перехрестя зі світлофором.
- **Car** – автомобіль. Клас забезпечує моделювання руху та відображення автомобілів.
- **Building** – будинок. Клас забезпечує відображення будинків та їх взаємодію з містом.

При роботі з містом створюється екземпляр класу **City**, з яким взаємодіють всі елементи інтерфейсу програми.

2.3. Опис алгоритмів

2.3.1. Моделювання

Моделювання в програмі відбувається дискретно із кроком по часу **dt**(у програмі - 50 мс). Це реалізується за допомогою компоненту **Timer**. На кожному кроці відбуваються наступні дії:

1.Рух автомобілів. Для кожного автомобіля прораховується, на яку відстань він має зміститись, рухаючись із максимальною швидкістю. Потім перевіряється, чи не зіткнеться він при цьому з автомобілем, що рухається попереду. Якщо так, то ця відстань зменшується. Потім координата машини на дорозі збільшується на одержану відстань.

Крім того, виконуються наступні додаткові перевірки.

- Якщо попереду даної машини рухається машина з меншою швидкістю та сусідній ряд є вільним, вона перестроюється до нього.
- Якщо машина під'їжджає до кінця дороги, надсилається запит на проїзд до відповідного перехрестя.
- Якщо відомо, що місце, куди прямує машина, знаходиться на даній дорозі, і вона наблизилась до цього місця, то машина відкріплюється від дороги й рухається прямолінійно до відповідного будинку; після прибуття до нього вона зникає з екрану, а кількість автомобілів у цьому будинку збільшується на одиницю. Якщо кількість автомобілів у будинку-цілі дорівнює його місткості, машина продовжує рух до найближчого будинку, де є вільні місця.

Окремо моделюється рух автомобілів, що виїжджають на дорогу. Вони рухаються прямолінійно до точки виїзду на дорогу. При під'їзді до цієї точки, якщо там немає іншої машини, вони виїжджають на дорогу.

2.Проїзд автомобілів на перехрестях. Автомобілі зі списку **MovingCars** на всіх перехрестях зміщуються на відстань $dt \cdot V_{п}$, де $V_{п}$ – швидкість проїзду перехрестя. Потім перевіряється, чи є довжина шляху, пройденого через перехрестя, більшою від розрахованої довжини траєкторії проїзду перехрестя. Якщо так, машина відкріплюється від перехрестя та прикріплюється до дороги, на яку має виїхати, при цьому їй встановлюється координата, що відповідає початку нової дороги.

3.Випадкова генерація автомобілів у будинках. Кількість машин N у кожному будинку в початковий момент задана. Для кожного будинку генерується випадкове число Z від 0 до 1. Воно порівнюється з величиною $dt \cdot V$, де dt – крок моделювання (с), V – ймовірність виїзду автомобіля протягом секунди. Ймовірність пропорційна наповненості будинку: $V = V_0 \cdot (N/X)$, де $V_0 = 0.2$ – константа (її значення підібрано експериментально з тих міркувань, що при такому V_0 одержуємо картину трафіку, досить близьку до реальної), X – місткість будинку. Якщо $Z < dt \cdot V$, у будинку є хоча б одна машина і при цьому дорога, на яку має виїжджати машина, вільна – створюється новий автомобіль, який виїжджає з даного будинку. Для створеного автомобіля знаходиться будинок, до якого він має рухатись і розраховується маршрут. При виїзді машини з будинку кількість автомобілів у ньому зменшується на одиницю.

2.3.2. Визначення цілі та оптимального маршруту автомобілів

Для кожної машини при її генерації визначається ціль – будинок, до якого вона має рухатись. Чим більше ймовірність того, що буде обрано даний будинок, тим більше його популярність. Нехай P_1, P_2, \dots, P_n – популярності будинків. Позначимо $S(i) = P_1 + P_2 + \dots + P_i$. Програма генерує випадкове число X в інтервалі від 0 до $S(n)$. Ціллю обирається будинок з номером Y – такий, що $S(Y-1) \leq X < S(Y)$.

Після вибору цілі для машини визначається маршрут її руху, а саме послідовність перехресть, які вона проїжджає. Сукупність доріг розглядається як орієнтований граф, дуги якого – дороги (дорога з двостороннім рухом відображується двома дугами з різними напрямками, з одностороннім – однією дугою), а вершини – перехрестя та точки в'їзду й виїзду даної машини. Вага ребер визначається як час проїзду дороги – відношення її довжини до середньої швидкості руху машин даною дорогою або до швидкості даної машини, якщо на дорозі машин немає або мало. Найкоротший шлях в даному графі розраховується за алгоритмом Дейкстри. Таким чином, коли рух автомобілів нормальний, обирається маршрут найменшої довжини з усіх допустимих, а коли в місті є затори, обирається маршрут найменшого часу.

2.3.3. Проїзд автомобілів через перехрестя

Кожне перехрестя має два списки: 1) **WaitingCars**, де зберігаються номери машин, що стоять безпосередньо перед перехрестям та очікують можливості проїзду; 2) **MovingCars**, де зберігаються номери машин, що рухаються перехрестям. Коли машина під'їжджає до перехрестя, до нього посилається запит на проїзд. При цьому розраховується початкова та кінцева точки траєкторії, за ними – кут та довжина шляху по перехрестю (цей шлях є прямолінійним), автомобіль видаляється з дороги та додається до списку **WaitingCars**. Раз на 500 мс для всіх машин цього списку визначається, чи можуть вони почати рух. Для цього їх шлях має не перетинати шлях машин, що вже рухаються перехрестям. Якщо на перехресті є світлофор, то перевіряється, чи дозволений у даний момент часу проїзд даного автомобіля. Якщо дане перехрестя – з головною дорогою, то спочатку визначається можливість проїзду для машин на головні дорозі. Якщо дозвіл на проїзд отримано, номер машини переводиться до списку **MovingCars**. Моделювання руху машин зі списку **MovingCars** описано вище.

2.3.4. Взаємодія автомобілів та будинків

Кожному будинку зіставляється точка **P** на певній дорозі, яка є основою перпендикуляру, опущеного з центра будинку на неї. Дорога обирається таким чином, щоб довжина перпендикуляру була мінімальною.

Коли машина виїжджає з будинку, вона спочатку рухається прямолінійно від його центру до точки **P**. Опинившись у ній, вона прикріплюється до даної дороги. Коли машина досягає точки, що відповідає будинку-цілі, вона відкріплюється від дороги та рухається до центру будинку. Після досягнення центру машина видаляється.

2.3.5. Візуалізація руху автомобілів

Автомобілі в програмі, крім розташування на дорозі, характеризуються координатами передньої точки та напрямком руху. При цьому напрямок руху може миттєво змінюватись. Щоб у такому випадку машина не розверталась миттєво, кут повороту прямокутника машини перераховується при кожному кроці пересування машини таким чином, ніби її тягнуть, прикладаючи силу до передньої точки. Це створює ефект повороту при русі через перехрестя, виїзді на дорогу або з дороги.

2.3.6. Визначення заторів

Для кожної машини на кожному кроці моделювання визначається її миттєва швидкість, як відношення фактичного зміщення до часу. Потім для кожної дороги, на якій є машини, визначається середнє арифметичне їх швидкостей. Якщо на певній дорозі значення цього параметру буде меншим від 10 м/с і при цьому машин більше від десяти, виводиться інформація про затор: дорога виділяється рамкою, колір якої змінюється від жовтого до червоного в залежності від середньої швидкості руху.

РОЗДІЛ 3. ПРИКЛАДИ ВИКОРИСТАННЯ ПРОГРАМИ

Нижче наводиться ряд прикладів, що демонструють використання засобів моделювання та аналізу у створеній програмі для оптимізації транспортних систем. Вони проілюстровані зображеннями міст та графічними звітами автоматичного моделювання.

Приклад 1

Маємо місто, що складається із двох частин, сполучених однією дорогою. Її пропускна здатність менша, ніж автомобільний потік між частинами міста, тому на ній та на прилеглих до неї дорогах виникає затор.

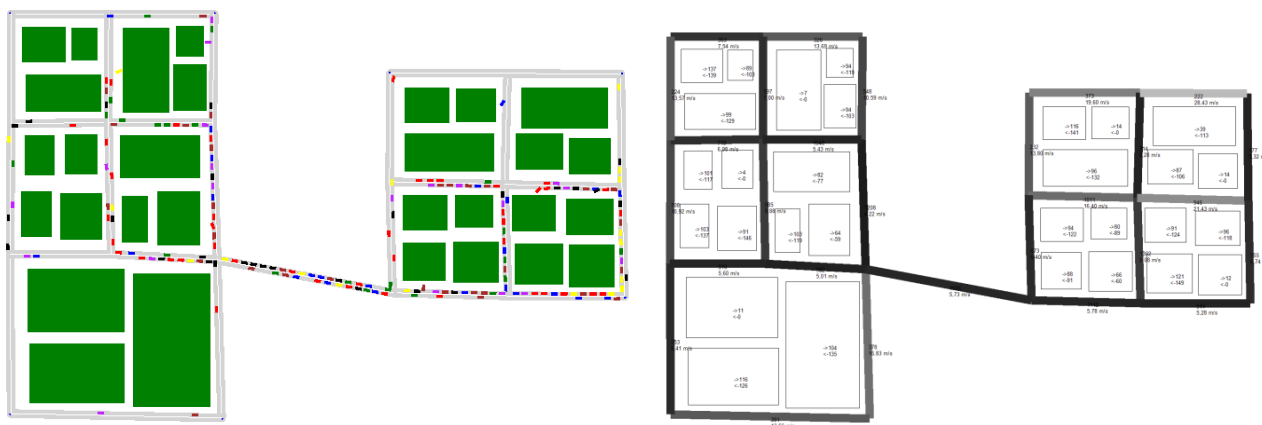


Рис.1 Схематичне зображення шляху автомобіля в місті.

Розширимо цю дорогу до двох смуг в обидва боки. При цьому середня швидкість руху автомобілів по ній та сусідніх дорогах збільшується.



Рис. 2. Розширення автошляху.

Замість того, щоб розширювати дорогу, можна побудувати нові дороги між частинами міста, знімаючи таким чином частину навантаження з першої дороги.



Рис. 3. Збільшенн числа автошляхів.

Приклад 2

Маємо проспект, на якому встановлені три світлофори (на перехрестях, що мають по чотири дороги). Цикл їх роботи складає 100с. При цьому машини вимушені довго стояти при червоному світлі, коли перпендикулярна дорога вільна. Це призводить до невиправданих заторів.

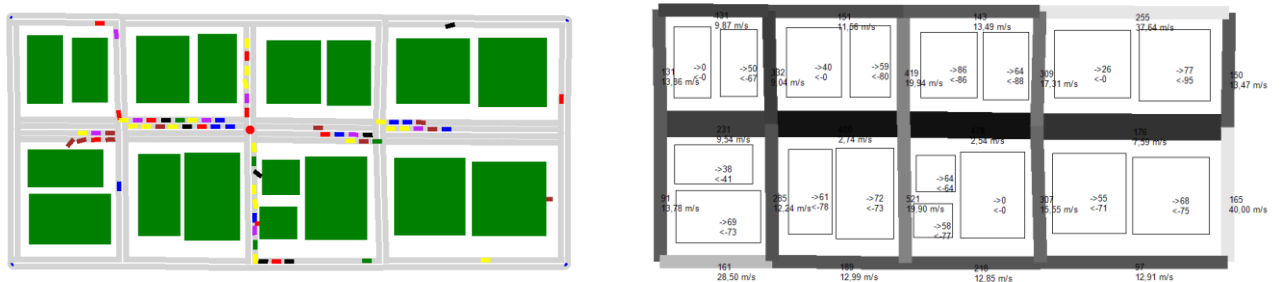


Рис. 4. Автошлях зі світлофорами.

Зменшимо цикл роботи світлофора до 30 с. При цьому зростає середня швидкість на всіх дорогах.

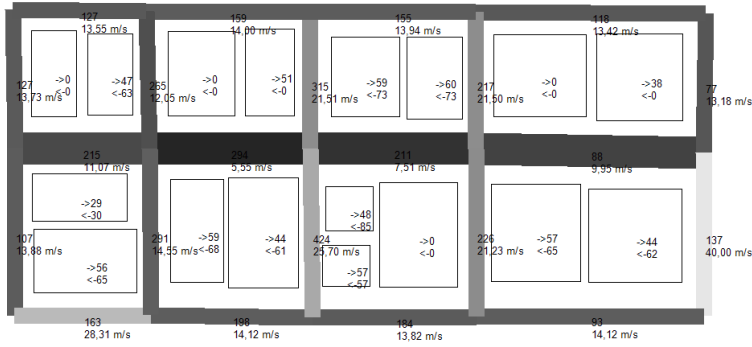


Рис. 5. Зростання пропускної здатності автошляху при зменшенні числа світлофорів.

Приклад 3

Маємо місто із досить активним рухом та низькою пропускною здатністю доріг (це характерно для старих частин багатьох сучасних міст).

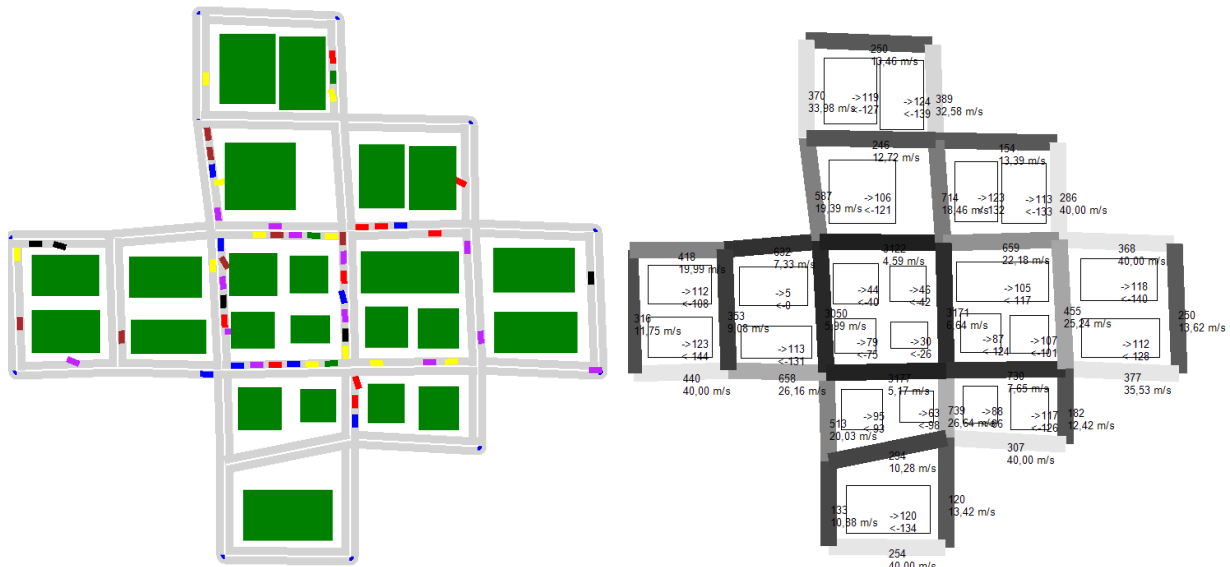


Рис. 6. Схема автошляхів без об'їзних доріг.

Побудуємо об'їзні дороги. Вони пропускають частину автомобілів, які раніше вимушені були рухатись через центр. При цьому ситуація в центрі міста поліпшується.

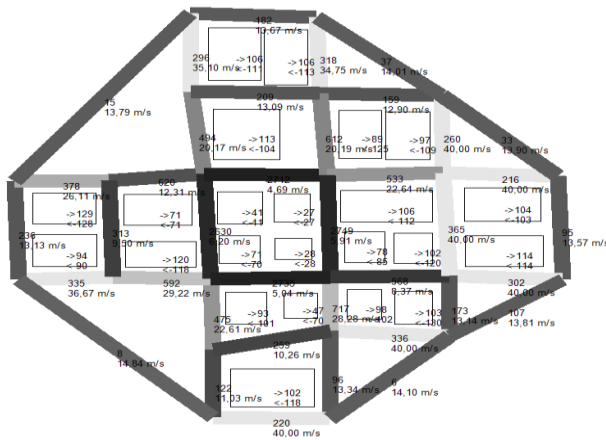


Рис. 7. Схема автошляхів з об'їзними дорогами.

ВИСНОВКИ

1. Проаналізовано існуючі методи моделювання міського трафіку, розроблено власну мережну модель міської транспортної системи та алгоритми комп'ютерного моделювання для неї.

2. Створено програмний продукт для комп'ютерного моделювання міського автомобільного руху. Він містить набір інструментів для створення моделі міста (редактор), моделювання міського трафіку та систему підтримки прийняття рішень для його оптимізації. Розроблено технічну документацію для підтримки роботи з продуктом.

Створена програма може бути використана місцевими органами самоврядування великих міст при проектуванні, розбудові та оптимізації транспортної системи міст або транспортними організаціями при прокладанні маршрутів. Також вона може використовуватись для теоретичних досліджень.

Перспективи розвитку теми роботи.

- Розгляд більш складних моделей (врахування міського транспорту, пішоходів, зміни інтенсивності та напрямку руху в залежності від часу доби, погодних умов тощо).
- Створення можливості автоматичної оптимізації транспортної системи міста, що моделюється.
- Застосування розробленої моделі для аналізу інших видів транспортних систем (повітряної, водної, залізничної) та споріднених явищ (передача інформації, міграційні процеси тощо).

ПОДЯКА

Автор глибоко вдячний науковому керівнику, кандидату фізико-математичних наук Ентіну Йосифу Абрамовичу за допомогу у виконанні роботи та Хенкіну Сергію Марковичу за постановку задачі та цінні рекомендації під час розробки програмного продукту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Семенов В.В. Математическое моделирование динамики транспортных потоков мегаполиса - Москва, Институт прикладной математики им. М.В. Келдыша – 2004. – 44с.
2. Трунин В. В. Компьютерное имитационное моделирование как способ решения транспортных проблем в городах. // Молодой ученый. — 2011. — №4. — С. 133-136.
3. Тымкив К. Хорошо стоим. Чем обернулся рост количества автомобилей в украинских мегаполисах. // Корреспондент. – 2012. – №48. – С.38-42.
4. Уизем Дж. Линейные и нелинейные волны. М.: Мир, 1977 – 624 с.
5. Чарльз Петцольд. Программирование для MicrosoftWindows на MicrosoftVisualBasic .NET – М.: Русская редакция – 2003. – В 2-х томах.
6. Окулов С.М. Программирование в алгоритмах – М.: БИНОМ – 2002 – 341с.
7. Климов А. Занимательное программирование на VisualBasic .NET – СПб.: БХВ-Петербург, 2005. – 528 с.
8. ДэнКларк. Объектно-ориентированное программирование в VisualBasic .NET. – 2003. – СПб.: Питер. – 352с.

ДОДАТКИ

Додаток А. Посібник користувача

Призначення та загальний вигляд програми

VirtualCity – це програмний продукт, призначений для комп'ютерного моделювання міського автомобільного руху. Він містить набір інструментів для створення моделі міста, моделювання міського трафіку та систему підтримки прийняття рішень для його оптимізації.

Головне вікно програми складається з меню та робочої області. Також програма має рухомі вікна інструментів та властивостей.

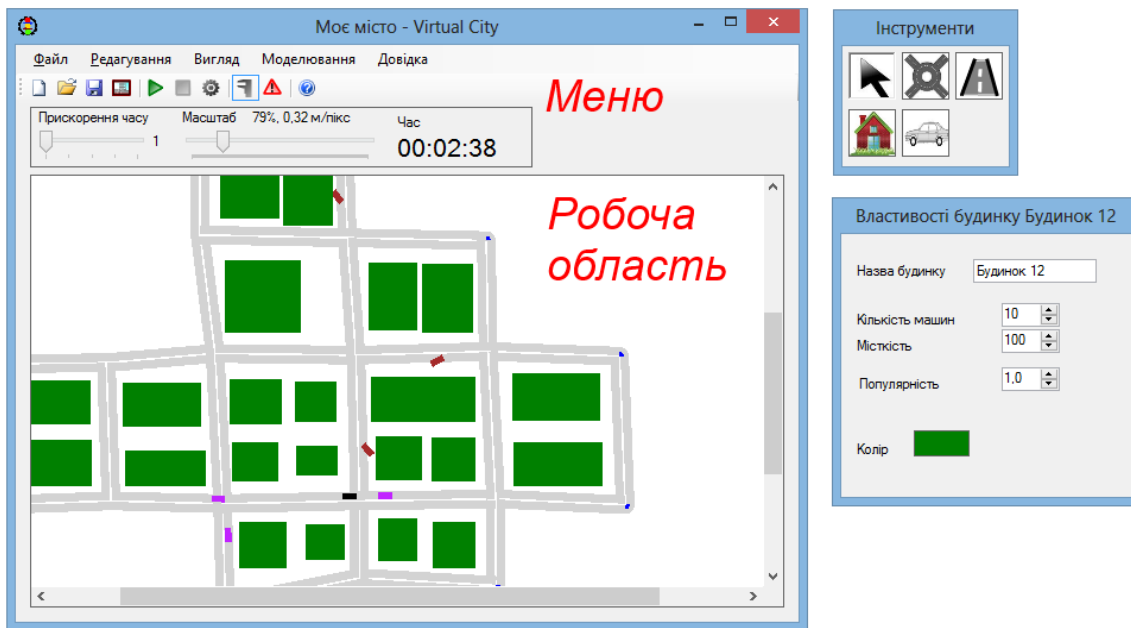






Рис. Д1. Робоча область та головне меню проекту.

Робота з містом

Програма **VirtualCity** зберігає всі дані про досліджуване місто у текстових файлах з розширенням ***.vcity**, з яких можна пізніше завантажувати ці дані. Файл з даними про місто у програмі називається «Місто».

Створення міста. Виберіть пункт меню **Файл|Нове місто**, або натисніть кнопку , або натисніть **Ctrl+N**. Тоді з'явиться діалогове вікно. Введіть до нього назву міста, його ширину та висоту й натисніть **OK**. Після цього ви зможете редагувати представлення створеного міста в робочій області.





Завантаження/збереження міста. Для завантаження міста виберіть пункт меню **Файл|Завантажити**, або натисніть кнопку , або натисніть **Ctrl+O**. Після цього виберіть файл та натисніть **OK**. Для збереження міста до файлу виберіть пункт меню **Файл|Зберегти** або натисніть кнопку , або натисніть **Ctrl+S**. Після цього виберіть місце збереження файлу, введіть його ім'я та натисніть **OK**.


Експорт зображення міста. Ви можете зберегти поточний вигляд міста до графічного файлу. Для цього виберіть пункт меню **Файл|Експорт зображення** або натисніть кнопку .

Масштабування та прокрутка. За допомогою пункту меню «Масштаб» можна задати потрібний масштаб міста. Якщо при цьому все місто не вміщується до робочої області, ви можете прокручувати його за допомогою смуг прокрутки у нижній та правій частинах робочої області.

Редагування міста

Додавання нових об'єктів. В концепції **VirtualCity** місто складається з перехресть, доріг, автомобілів та будинків.

Спочатку виберіть інструмент **Перехрестя** () та за допомогою лівої кнопки миші (ЛКМ) відмітьте перехрестя в необхідних місцях. Потім виберіть інструмент **Дорога** () та з'єднайте деякі перехрестя, натискаючи на них по черзі. Далі виберіть інструмент **Будинок** () та відмітьте будинки. Для цього наведіть курсор на ліву верхню точку будинку, затисніть ЛКМ, протягніть його до нижньої правої точки будинку та відпустіть. Для того, щоб додати автомобілі, виберіть інструмент **Автомобіль** () та натисніть ЛКМ в потрібних місцях доріг.

Редагування об'єктів. Для редагування міста користуйтеся інструментом вибору (). Для вибору об'єкту натискайте на ньому ЛКМ.

Щоб перемістити перехрестя, затисніть на ньому ліву кнопку миші (ЛКМ), протягніть до необхідної точки та відпустіть. Якщо при цьому дороги будуть проходити крізь будинки чи перетинатися між собою, переміщення не відбудеться.

Для того, щоб змінити розмір будинку, наведіть курсор на його межу, затисніть ЛКМ та перемістіть цю межу.

Для редагування параметрів міста (назва, ширина, висота та колір фону) натисніть ЛКМ на порожній ділянці міста та задайте відповідні параметри у вікні властивостей.

Налаштування об'єктів міста. За допомогою вікна властивостей можна задавати параметри об'єктів міста. Для цього виберіть необхідний об'єкт та введіть відповідні параметри. Для автомобілів можна встановлювати максимальну швидкість, для будинків – кількість машин у них, місткість, популярність та колір, для доріг – кількість рядів в кожному напрямку (від нуля до шести). Також для всіх об'єктів можна задавати назву.

Властивості будинків. У вікні властивостей для кожного будинку задаються такі параметри, як місткість, кількість машин та популярність.



Місткість будинку – це максимальна кількість машин, які можуть знаходитись у ньому. Кількість машин задається в інтервалі від нуля до місткості. Вона змінюється у процесі моделювання. В режимі моделювання не можна змінювати ці два параметри. Відношення кількості автомобілів до місткості визначає частоту виїзду автомобілів.



Популярність – це коефіцієнт, який визначає інтенсивність потоку машин до даного будинку відносно інших будинків. У скільки разів популярність одного будинку більша за популярність іншого, у стільки разів більше машин (в середньому) в'їжджатиме до нього за одиницю часу.

Візуалізація властивостей будинків. На будинках при увімкненій опції **Вигляд|Показувати відмітки** відображуються властивості будинків. На кожному будинку зображується коло та квадрат. Колір кола вказує на міру заповненості будинку (відношення кількості машин в ньому до місткості). Колір квадрата вказує на популярність. Кольори змінюються від зеленого (мінімальне значення) до червоного (максимальне значення).

Способи регулювання перехресть. В програмі існують три види перехресть – нерегульоване, з головною дорогою та зі світлофором. Їх можна задавати у вікні властивостей. На нерегульованому перехресті машини перетинають перехрестя в тому порядку, в якому під'їжджають, на перехресті з головною дорогою пріоритет надається машинам з головної дороги. Щоб задати головну дорогу, треба вказати дві дороги, що підходять до даного перехрестя і утворюють головну дорогу. На перехресті зі світлофором у певні моменти часу машини можуть рухатись лише з заданих доріг на інші задані дороги. Щоб створити програму світлофора, необхідно задати його цикл C у секундах. Потім необхідно задати набір режимів. Режим має чотири параметри: час початку t_1 , час закінчення t_2 ($0 \leq t_1, t_2 \leq C$), дорогу-в'їзд r_1 та дорогу-виїзд r_2 . Такий режим означає, що через дане перехрестя машини з дороги r_1 на дорогу r_2 можуть рухатись лише в проміжки часу міста $[C \cdot n + t_1; C \cdot n + t_2]$, де $n = 0, 1, 2, \dots$. Щоб задати режим, натисніть кнопку «Додати» у вікні властивостей та введіть необхідні параметри. Щоб видалити режим, натисніть «Видалити». На перехресті з чотирма дорогами можна встановити стандартну програму, яка по черзі пропускатиме машини з під'їздів, які утворюють одну дорогу. Для цього натисніть кнопку «Авто».

Рекомендації щодо налаштування міста. Програма розрахована на моделювання ситуацій, наближених до реальності. Тому рекомендується створювати такі моделі міст, які могли б існувати в реальності. Не робіть перехрестя, з яких виходить більше ніж п'ять доріг, не задавайте дуже велику ймовірність виїзду машин із будинків, не робіть програми світлофорів, які не пропускають машини в якихось напрямках. Транспортна система має бути зв'язаною (тобто з будь-якої дороги можна доїхати до іншої). У моделі не можна задавати перехрестя, з яких виходить лише одна дорога.




Допоміжні відмітки. Якщо в місті наявні мости (тобто дороги, при видаленні яких місто розпадається на незв'язані компоненти), біля них з'явиться позначка моста () . Якщо на деякій дорозі має місце низька середня швидкість, біля неї з'явиться позначка вузької дороги (). Біля


регульованих перехресть відображається позначка світлофора (). Ці позначки можна вмикати та вимикати за допомогою пункту меню **Вигляд|Показувати відмітки** або кнопки  .

Додаткові можливості редагування. В пункті меню «**Редагування**» наявні наступні інструменти:

- **Місто.**
 - **Створити сітку кварталів** – створити систему із перехресть і доріг, яка задає класичний вигляд міста із квадратними кварталами.
 - **Очистити** – видалити всі об’єкти.
 - **Вирівняти по сітці** – перемістити перехрестя та вершини будинків до найближчих вузлів уявної сітки.
- **Автомобілі.**
 - **Видалити всі автомобілі.**
- **Будинки.**
 - **Задати кількість автомобілів** – задати для всіх будинків однакову місткість та кількість автомобілів.
 - **Задати один колір.**
 - **Задати випадкові кольори.**

Моделювання

Моделювання в реальному часі. Для того, щоб розпочати моделювання, виберіть пункт меню **Моделювання|Почати**, натисніть кнопку  або клавішу F5. Для того, щоб припинити моделювання, виберіть пункт меню **Моделювання|Зупинити**, натисніть кнопку  або клавішу F6. Моделювання відбуватиметься в реальному часі, але для його прискорення можна змінювати прискорення руху внутрішнього годинника програми за допомогою пункту меню «**Прискорення часу**» від 1 до 5. Під час моделювання не рекомендується редагувати місто. Якщо під час моделювання на деяких дорогах з’являтимуться затори, ці дороги будуть обведені рамкою, колір якої змінюватиметься від жовтого до червоного в залежності від «важкості» затору. Щоб увімкнути/вимкнути цю опцію, використовуйте пункт меню **Вигляд|Показувати відмітки** або кнопку  . Під час моделювання в головному меню відображається поточний час міста у форматі ГГ:ХХ:СС.

Автоматичне моделювання. В програмі передбачена можливість автоматичного моделювання. Для цього виберіть пункт меню **Моделювання|Автоматичне моделювання**, натисніть кнопку  або клавішу F7. Після цього з’явиться діалогове вікно:

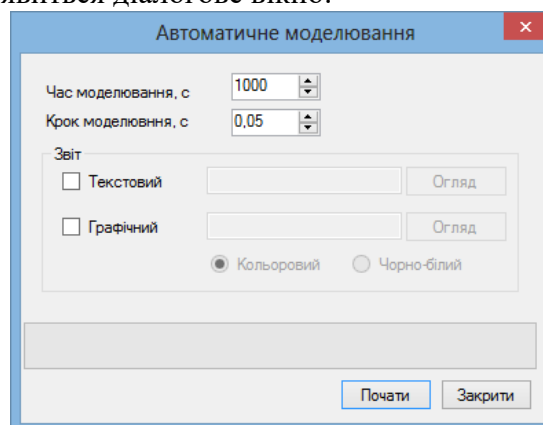


Рис. Д2. Діалогове вікно «Автоматичне моделювання».

В ньому необхідно задати час моделювання та крок моделювання (від 0,05с до 1с). Після натискання кнопки «**Почати**» програма буде моделювати рух автомобілів за вказаний час. Статус моделювання буде відображати заповнення смуги. Після завершення моделювання буде відображений стан міста на момент завершення.

Звіти автоматичного моделювання. Програма може створювати звіти, в яких відображаються параметри, що характеризують місто під час моделювання (для будинків – кількість автомобілів, що вїхали та виїхали; для доріг – кількість автомобілів, що проїхали та середня швидкість проїзду). Текстовий звіт – це текстовий файл, де ці параметри представлені у вигляді таблиці. Графічний звіт – зображення міста (будинки, дороги та параметри, написані на них). Крім того, швидкість проїзду машин відображена за допомогою кольорів доріг (на кольоровому звіті – відтінками червоного й жовтого, на чорно-білому – відтінками сірого). Ці звіти допомагають проаналізувати стан транспортної системи даного міста протягом тривалого часу.

Для того, щоб створити звіт, помітьте відповідні опції у вікні моделювання перед моделюванням та вкажіть місце збереження звіту. По закінченню моделювання можна відкрити ці звіти за допомогою кнопки «Відкрити».

Додаток Б. Приклади програмного коду

Метод класу **City**, який моделює рух в місті за малий проміжок часу **dt**.

```
PublicSubModelStep(ByValdtAsSingle)
    'Пересуваємовсімашини
    Fori = 1 ToCarsCounter
        IfCars(i).ExistsThenCars(i).Move(dt, Me)
    Next
    'Пересуваємомашининаперехрестях
    Fori = 1 ToISCounter
        IfIntScs(i).ExistsThenIntScs(i).Move(dt, Me)
    Next
    'Випадковийвиїздізбудинків
    Fori = 1 ToHsCounter
        IfRnd() <Building.MaxOutPossibility * (Houses(i).CarsNum / Houses(i).MaxCarsNum) _
            * dtThenHouses(i).CreateCar(Me)
    Next
    'Збільшуємочасміста
    Time += dt
EndSub
```

Функція класу **Car**, яка визначає прямокутник, що зображує машину.

```
PublicFunctionGetRect(ByRef CT AsCity) AsPointF()
    Dimpnts(4) AsPointF
    'Перпендикуляр до напрямку орієнтації машини
    Dimang1 AsSingle = TrueRot - PI / 2
    Dim p3 AsPointF = Pos 'Передня точка машини
    'Задняточкамашини
    Dim p4 AsPointF=NewPointF(p3.X - Length * Cos(TrueRot), p3.Y - Length * Sin(TrueRot))
    'Визначаємокоординативершинпрямокутника
    pnts(0) = NewPointF(p3.X - 0.5 * Width * Cos(ang1), p3.Y - 0.5 * Width * Sin(ang1))
    pnts(1) = NewPointF(p3.X + 0.5 * Width * Cos(ang1), p3.Y + 0.5 * Width * Sin(ang1))
    pnts(3) = NewPointF(p4.X - 0.5 * Width * Cos(ang1), p4.Y - 0.5 * Width * Sin(ang1))
    pnts(2) = NewPointF(p4.X + 0.5 * Width * Cos(ang1), p4.Y + 0.5 * Width * Sin(ang1))
    pnts(4) = pnts(0)
    Returnpnts
EndFunction
```

Метод класу **Car**, який «протягує» машину на відстань **dist** під кутом **MovRot** за передню точку.

```
'Pos - координата передньої точки, MovRot - напрям руху, TrueRot - орієнтація
PublicSubDrag(ByValdistAsSingle)
    'Зміщуємопереднюточку
    Pos = NewPointF(Pos.X + dist * Cos(MovRot), Pos.Y + dist * Sin(MovRot))
    'Визначаємонобуорієнтаціямашини
    Dim l AsSingle = Me.Length
    DimangAsSingle = TrueRot - MovRot
    Dim x AsSingle = Sqrt(l * l + dist * dist + 2 * l * dist * Cos(ang))
    TrueRot -= Asin(dist / x * Sin(ang))
```

EndSub

Метод класу **Car**, який моделює зміщення машини за **Time**секунд.

PublicSubMove(ByValTimeAsSingle, ByRef CT AsCity)

'Відстань, яку має пройти машина

DimdistAsSingle = Speed * Time

'Якщо машина виїжджає із будинку на дорогу

IfNotAtRoadAndGoToRoadThen

GR_Pos += dist

Drag(dist)

IfGR_Pos>GR_LenThen *'Якщо машина виїхала*

Me.GoToRoad = False

Me.AtRoad = True

Me.MovRot = CT.Roads(Me.Road).Rotat

IfMe.Line> 0 ThenMovRot += PI

CT.Houses(GR_House).GR_IsCarGoingOut = False

Me.GR_House = 0

EndIf

ExitSub

EndIf

'Якщо машина в'їжджає з дороги до будинку

IfGoFromRoadThen

GR_Pos += dist

Drag(dist)

IfGR_Pos>= GR_LenThen

CT.Houses(GR_House).Analyse_InComeCounter += 1

Me.Exists = False

EndIf

ExitSub

EndIf

'Якщо машина перестроюється

IfChangingLineThen

CL_Pos += dist

Drag(dist)

IfCL_Pos>= CL_LenThen

ChangingLine = False

Me.MovRot = CT.Roads(Me.Road).Rotat

IfMe.Line< 0 ThenMe.MovRot += PI

Me.TrueRot = Me.MovRot

EndIf

ExitSub

EndIf

'З'їзд із дороги

IfLastRoadOfRouteAndPosAtLine + dist>= CT.Houses(TargetBuilding).Targ_PosAtLineThen

dist = CT.Houses(TargetBuilding).Targ_PosAtLine - PosAtLine

Me.GoFromRoad = True

Me.RouteDetermined = False

Me.MovRot = Geometry.AngleBy2Pt(Me.Pos, CT.Houses(TargetBuilding).Targ_InPoint)

Me.GR_Len = Geometry.Dist(Me.Pos, CT.Houses(TargetBuilding).Targ_InPoint)

Me.GR_Pos = 0

Me.GR_House = TargetBuilding

PosAtLine += dist

Pos = GetPos(CT)

ExitSub

EndIf

```

'Якщо попереду є машина, зменшуємо швидкість
DimIsCarForwardAsBoolean = False
DimSafeDistAsSingle = City.SafeDist
Fori = 1 ToCT.CarsCounter
  IfCT.Cars(i).ExistsAndNot (i = Me.Number) AndCT.Cars(i).Road = Me.Road _
  AndCT.Cars(i).Line = Me.LineThen
    DimCrBackAsSingle = CT.Cars(i).PosAtLine - CT.Cars(i).Length
    IfCrBack>PosAtLineAndCrBack - SafeDist<PosAtLineThen
      dist = 0
      IsCarForward = True
    ElseIfCrBack>PosAtLineAndCrBack<PosAtLine + distThen
      dist = CrBack - SafeDist - PosAtLine
      IsCarForward = True
    EndIf
  EndIf
Next

'Спроба перестроювання
DimLineChOkAsBoolean = False
IfIsCarForwardAnd _
Me.PosAtLine<CT.Roads(Me.Road).Length - IntZone(CT) - Me.LengthThen
  IfMe.Line> 0 Then
    IfCT.Roads(Me.Road).n1 >Me.LineThen
      TryStartChangeLine(Me.Line + 1, CT, LineChOk)
      IfMe.Line> 1 ThenTryStartChangeLine(Me.Line - 1, CT, LineChOk)
    EndIf
  Else
    IfCT.Roads(Me.Road).n2 >Abs(Me.Line) Then
      TryStartChangeLine(Me.Line - 1, CT, LineChOk)
      IfMe.Line< -1 ThenTryStartChangeLine(Me.Line + 1, CT, LineChOk)
    EndIf
  EndIf
EndIf
IfLineChOkThenExitSub

'Якщо машина доїхала до перехрестя
IfPosAtLine + dist>CT.Roads(Road).Length - IntZone(CT) Then
  dist = CT.Roads(Road).Length - IntZone(CT) - PosAtLine
  IST_NewRoad = GetNextRoad(CT)
  DimNextIntersectionAsInteger = NextInsc(CT)
  CT.IntScs(NextIntersection).RequestPass(Number, IST_NewRoad, CT)
EndIf

'Зміщуємо машину
PosAtLine += dist
Drag(dist)
Pos = GetPos(CT)
EndSub

```