

ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ МОДЕЛЮВАННЯ ТА КЕРУВАННЯ РОБОТОЮ ІНФОКОМУНІКАЦІЙНИХ МЕРЕЖ

© Дронюк І., Кошулінський Р., 2007

Розроблено кросплатформний програмний комплекс для моделювання та оптимізації транспортних потоків в інфокомунікаційних мережах. Докладно описано структуру та принцип роботи з ним. Зазначено сфери застосування та перспективи його подальшого розвитку.

A cross-platform application for infocommunication networks modelling and traffic optimization was developed. The structure of an application and principles of work are thoroughly described. Perspectives of its further development and current usage are mentioned.

Вступ

Проблемі підвищення ефективності роботи інфокомунікаційних мереж присвячено багато робіт українських та зарубіжних вчених. Для вирішення цієї проблеми використовується багато різних математичних підходів. Методи та засоби підвищення ефективності роботи оптичних транспортних систем як основи інфокомунікаційних систем на основі аналізу топологічних властивостей мережевих структур розроблено у монографії [1]. Ланцюгові моделі теорії масового обслуговування для вирішення даної проблеми запропоновано у [2]. Задачі математичного моделювання інфокомунікаційних мереж та алгоритми оптимізації потоків на цих мережах розглядаються в монографії [3]. Розроблення математичних моделей та на їх основі побудова імітаційних моделей, створених на ЕОМ, які дають змогу прораховувати оптимальні режими роботи мережі – один із сучасних напрямів розв'язання цієї задачі. Такий підхід представлений у монографії [4]. Один із способів моделювання інфокомунікаційних мереж – використання теорії графів. Саме цей спосіб активно використовують автори під час побудови математичних моделей, покладених в основу розробленого кросплатформного програмного комплексу. Комп'ютерній реалізації алгоритмів на графах присвячено монографію [5]. Ужиток схожого спрямування описується у статті [6].

Застосування сучасних інформаційних технологій відкриває нові можливості для імітаційного моделювання різноманітних мереж та реалізації класичних алгоритмів оптимізації транспортних потоків. Для побудови комп'ютерної системи моделювання інфокомунікаційних мереж вибрано мову програмування С#. Написання програмного продукту здійснювалось у середовищі Visual Studio 2005 з використанням об'єктно-орієнтованого підходу. До розв'язання цієї задачі використано інтерфейс абстрактних типів даних (АТД). Для моделювання транспортної мережі застосовано математичний апарат теорії графів. Для візуального зображення інфокомунікаційної мережі були написані власні компоненти. Для розрахунків оптимальних шляхів та транспортних потоків реалізовано власні алгоритми, які ґрунтуються на класичних алгоритмах пошуку мінімальних шляхів на графах і максимальних потоків.

Загальна характеристика проекту

Проект Graph .NET – це програмний комплекс, орієнтований на платформу Microsoft .NET Framework, який дає змогу моделювати інфокомунікаційні мережі з використанням теорії графів та розв'язувати задачі оптимізації та управління транспортними потоками. Сьогодні поширені не тільки персональні комп'ютери, а й кишенькові (КПК) та смартфони. Тому програмний код розроблявся з урахуванням його подальшого використання на цих пристроях. Розроблений

програмний комплекс є кросплатформний, у випадку звичайних персональних комп'ютерів та ноутбуків він працює під операційними системами сімейства Windows (2000/XP/Vista) після встановлення .NET Framework версії 2.0 і вище. У випадку КПК і смартфонів необхідний Windows Mobile 2003 і вище при встановленні .NET Compact Framework 2.0 і вище. Варто також зазначити, що для коректної роботи на смартфоні необхідний сенсорний екран, оскільки більшість маніпуляцій з графами (додавання/вилучення вершин, побудова шляху і т.п.) для зручності користувача розроблені так, що реалізуються сенсорно, без допомоги апаратних клавіш. Схематично апаратні платформи та операційні системи, які підтримуються ужитком, зображено на рис. 1.

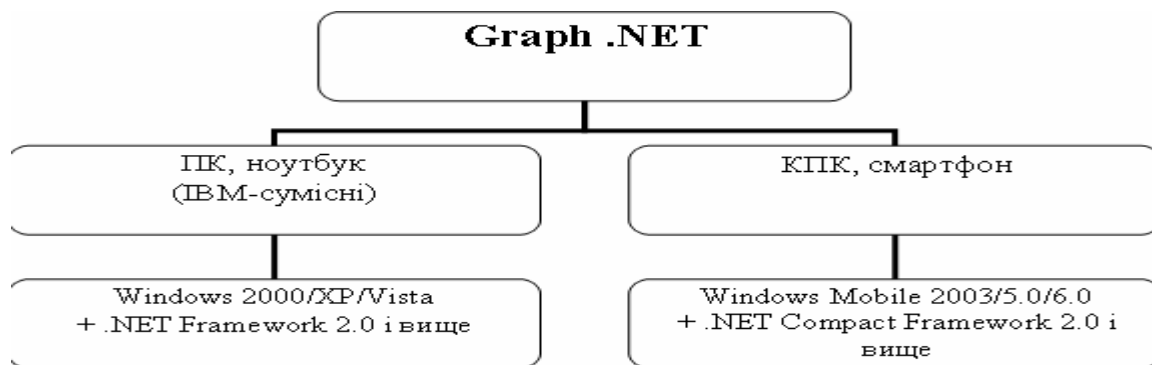


Рис. 1. Апаратні та програмні конфігурації, які підтримує програмний комплекс Graph.NET

Програмний комплекс складається із двох частин, які компілюються окремо. Структуру проекту зображено на рис.2. Перша частина – це ядро, яке містить всі основні класи для зберігання графів і роботи над ними, і компілюється в бібліотеку GraphUtils.dll. Воно містить реалізації алгоритмів на графах із урахуванням особливостей мереж, елемент управління для візуалізації графа, за допомогою якого моделюємо мережу, а також засоби для серіалізації даних в XML-файли. Друга частина містить інтерфейс користувача і компілюється у виконавчий файл Graph.net.exe.

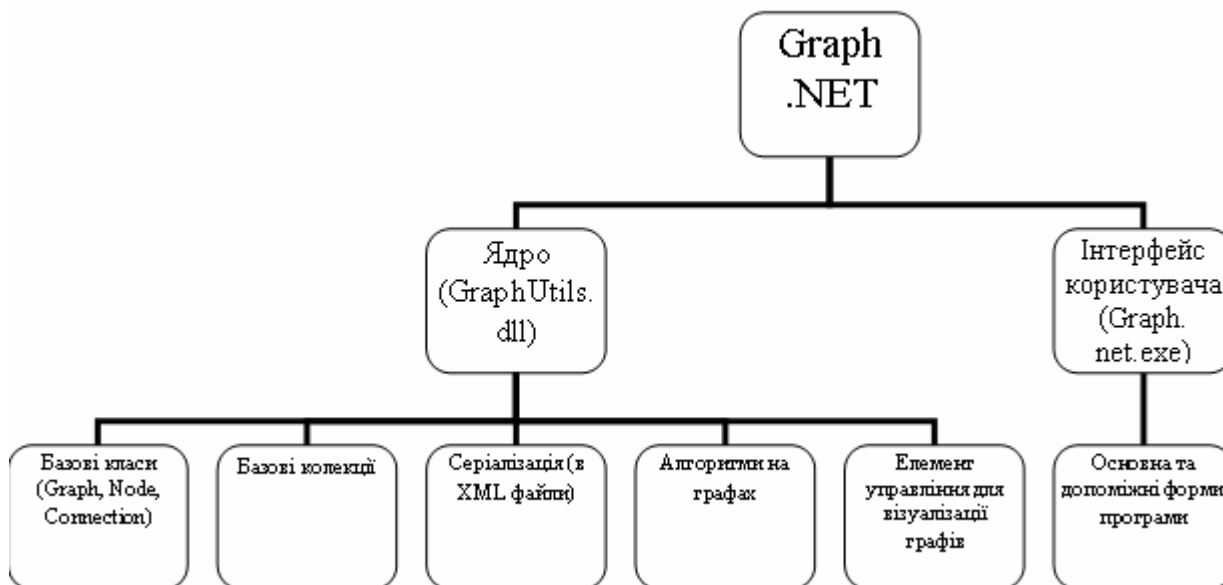


Рис. 2. Структура програмного комплексу Graph.NET

Перевага такого підходу полягає у тому, що надалі, не вникаючи в особливості реалізації тих чи інших алгоритмів, можна написати свій користувацький інтерфейс, і навпаки, можна додавати нові алгоритми, не змінюючи при цьому користувацького інтерфейсу.

Розроблений програмний комплекс має дружній користувачський інтерфейс. Основне вікно містить головне меню 1, панель інструментів 2, елемент управління для роботи з графом 3 та панелі додаткових налаштувань 4, 5 (рис. 3).

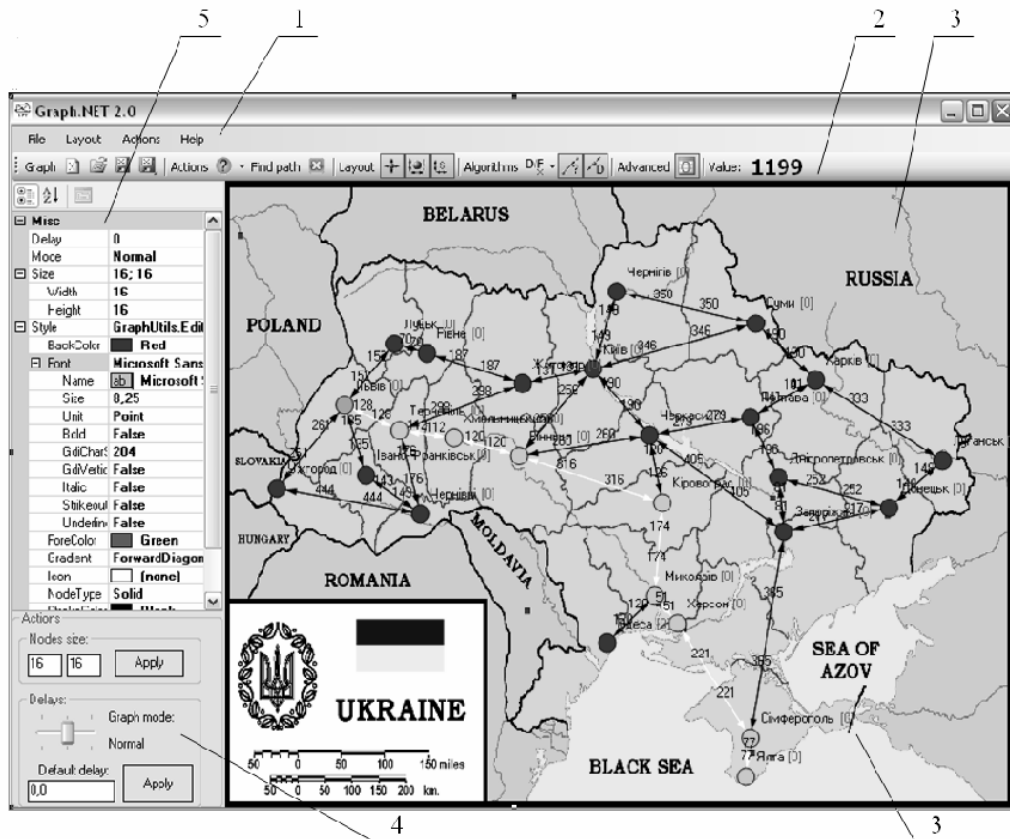












Рис. 3. Основне вікно програмного комплексу GRAPH.NET: 1 – головне меню; 2 – панель інструментів; 3 – елемент управління для роботи з графом; 4 – панель додаткових налаштувань для графу; 5 – панель додаткових налаштувань для вершин і ребер графу

Головне меню містить такі опції:

- File – містить опції New, Save, Save As, Load, які дають змогу створити новий граф, зберегти його чи завантажити з файла відповідно. Пункт Exit відповідає за вихід з ужитку.
- Layout – містить набір операцій для впорядкування вершин графу. Підпункти Random і Rectangle відповідають за розміщення вершин випадковим чином або у вигляді прямокутника. Операції Layout nodes, Freeze nodes moving, Save graph size відповідають за зміну розміщення вершин графу при зміні розмірів основного вікна, заборону пересувати вершини мишкою, зберігання розміру та координат графу на елементі управління у поточному файлі відповідно.
- Actions – відповідає за зміну режимів роботи програми. Підпункт Delete відповідає за видалення виділеного попередньо мишкою вершини чи ребра графу.
- Help – містить вказівки користувачу, інформацію про попередні версії та розробників програми.

Панель інструментів 2 (рис. 3), розміщена безпосередньо під головним меню, призначена для швидкого доступу до всіх основних функцій ужитку за допомогою графічних піктограм. Всі піктограми мають спадні підказки.

Охарактеризуємо панель інструментів докладніше.     – це піктограми для створення нового файла з графом, завантаження існуючого чи зберігання графу у файл.  – піктограма, що відображає режими роботи програми і дає змогу їх змінювати.  –

ця піктограма по суті дублює підпункт Delete пункту Action головного меню.  – ці піктограми відповідають за зміну розташування вершин графу при зміні розміру основного вікна, заборону перетягувати вершини мишкою та зберігання розміру графу у файл відповідно і фактично дублюють аналогічні пункти з головного меню.  – ці піктограми відповідають за налаштування алгоритмів пошуку найкоротших з'єднань. Перша піктограма відповідає за алгоритм і дає змогу його вибрати, друга – за попередню перевірку існування шляху, за допомогою третьої – враховують затримки на вузлах.  – за допомогою цієї піктограми можна показувати та ховати панелі додаткових налаштувань 4 і 5 (рис. 3).  – тут виводиться результат обчислень. У випадку реалізації алгоритму пошуку найкоротшого з'єднання – це величина обчисленого оптимального з'єднання. Якщо реалізується алгоритм максимального потоку – це величина обчисленого максимального потоку. Розмірність залежить від вихідних даних.

Ужиток має різні режими роботи:

- Додавання вершин (Add node). У цьому режимі натисканням мишки в полі елемента управління створюється нова вершина. За допомогою мишки її можна перетягувати, а за допомогою панелі 5 (рис.3) змінювати її налаштування.
- Зв'язування вершин (Add connection). У цьому режимі можна зв'язувати між собою вершини графу. Для цього спочатку мишкою виділяють вершину, яку потрібно зв'язати, потім перетягують утворене сполучення до потрібної вершини.
- Пошук найшвидшого з'єднання (Find path). Реалізує модифікований алгоритм розрахунку найкоротшого з'єднання між заданими двома вузлами. Для оптимізації необхідно клацнути мишкою на початковому вузлі, а потім двічі клацнути на кінцевому. Розрахований оптимальний шлях буде підсвічений іншим кольором, а числове значення відобразиться у полі Value на панелі інструментів 2.
- Пошук максимального потоку (Get flow). Реалізує розрахунок максимального потоку між двома заданими вузлами. Початковий та кінцевий вузли вибирають аналогічно до режиму пошуку найкоротших з'єднань. Розрахована структура оптимального потоку буде підсвічена іншим кольором. Обчислене числове значення буде зображено у полі Value на панелі інструментів 2.
- Перевірка існування з'єднання між вузлами (Path existence). У цьому режимі перевіряється існування з'єднання між двома заданими вершинами. Вершини задаються так само, як і у попередніх двох режимах.

Для врахування особливостей роботи мережі панелі налаштувань 4 і 5 містять налаштування розмірів вершин графу, затримок на вузлах, які приймаються за замовчуванням та повзунок вибору режиму роботи мережі – легкий, нормальний або навантажений. Режимми роботи впливають на затримки сигналу на вузлах.

Окрім згаданих вище налаштувань, на панелі 5 розміщується таблиця значень, що генерується залежно від контексту. Якщо користувач вибрав вершину графу, то таблиця містить властивості вибраної вершини графу, якщо ребро – то властивості цього ребра (з'єднання). Опишемо їх детальніше. Вершини графу (або вузли в інфокомунікаційній мережі) мають такі властивості: Delay – затримка сигналу на вузлі; Mode – режим роботи вибраного вузла; Size – розміри вибраного вузла на екрані (у пікселях); Text – містить текст надпису над вершиною; Style – містить інформацію про вигляд вузла на екрані. Це кольори в різних станах, шрифт, яким робляться надписи, присутність піктограмки, кольорове заповнення вершини (суцільне або градієнтне).

Ребра графу (або з'єднання в інфокомунікаційній мережі) мають такі властивості: Capacity – пропускна здатність вибраного каналу; Distance – віддаль між вузлами, які цей канал з'єднує; Enabled – вказує на доступність каналу в певний момент часу; Flow – показує значення потоку, який проходить вибраним каналом у певний момент часу.

Зберігання даних

Оскільки граф є ієрархічною структурою, було розроблено такі абстрактні типи даних, кожен з яких, крім найелементарнішого, містить сукупність інших і відповідає окремому класу. Дані про граф зберігаються в XML-файлі, а також можуть бути з нього завантажені. У програмному комплексі передбачено можливість підвантажувати зображення географічних карт, на які накладаються моделі відповідних мереж. XML-файли, згенеровані в попередніх версіях програми, повністю підтримуються її наступними версіями.

Структуру графу, який зберігається у файлі, наведено на рис.4.

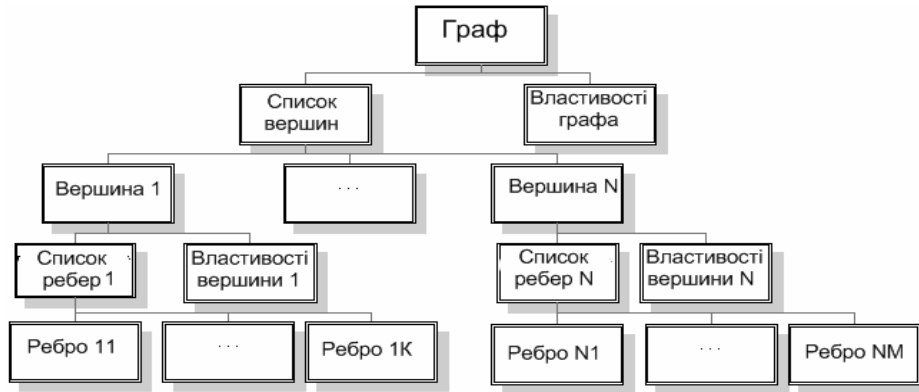


Рис. 4. Структура графу, який зберігається у XML-файлі

Ще одна перевага зберігання даних про граф у XML-файлах полягає у тому, що користувач може редагувати граф за допомогою звичайного текстового редактора. Але при цьому треба знати порядок розміщення і призначення елементів XML-файла, що створюється, оскільки після некоректного редагування файл не зможе бути завантаженим.

Висновки

1. Розроблено кросплатформний програмний комплекс для моделювання транспортних мереж з застосуванням об'єктно-орієнтованого підходу мовою C# на основі математичних моделей, побудований за допомогою математичного апарату теорії графів.

2. Програмний комплекс має зручний користувацький інтерфейс і дає змогу розв'язувати оптимізаційні задачі для транспортних та інфокомунікаційних мереж, зокрема знаходити оптимальний маршрут проходження трафіку між вузлами та структуру потоку максимальної пропускної здатності. Розроблений ужиток можна застосовувати для моделювання та оптимізації роботи різноманітних мереж, не тільки інфокомунікаційних. Разом з тим розроблений ужиток враховує особливості побудови саме інфокомунікаційної мережі, наприклад, три режими навантаження протягом доби та інші.

3. За допомогою розробленого програмного комплексу можна здійснювати перерозподіл транспортних потоків у режимі реального часу на випадок аварійних ситуацій, оскільки у моделі підтримується можливість виведення того чи іншого сполучення або вузла у неробочий стан.

4. Проект постійно розвивається, причому паралельно удосконалюються версії для настільних ПК та КПК і смартфонів. У наступних версіях будуть додані нові можливості для моделювання різних мереж та удосконалені вже існуючі оптимізаційні алгоритми.

1. Захарченко М.В., Климаш М.М. *Методи та засоби підвищення ефективності оптичних транспортних систем.* – Львів, 2007. – 186 с. 2. Czachorski T. *Modele kolejowe w ocenie efektywnosci sieci I systemow komputerowych.* – Gliwice. – 336 s. 3. *Основи математичної теорії інфокомунікаційних систем / Під ред. В.В. Поповського.* – Харків: СМІТ-Компанія, 2006. – 564с. 4. Зайченко Е.Ю. *Сети АТМ: Моделирование, анализ и оптимизация.* – К., 2003. – 216 с. 5. Седжвик Р. *Фундаментальные алгоритмы на C++. Часть 5. Алгоритмы на графах.* – К.: Диасофт, 2004. 6. Зайченко Ю.П., Зайченко Е.Ю., Поспелов И.В. *Комплекс программ анализа и синтеза структуры региональных и глобальных вычислительных сетей // Управляющие системы и машины.* – 2000. – №5/6. – С.71–87.