

О. ПОНОМАРЕНКО, В. ГОРБАЧОВ

ПРОГРАМНА ПЛАТФОРМА ДЛЯ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ АГРЕГАЦІЇ СТРУКТУРНОЇ МОДЕЛІ СКЛАДНИХ СИСТЕМ

Складні системи визначаються великою розмірністю, містять значну кількість елементів та зв'язків між ними. Мережі використовуються для подання складних систем. Через велику розмірність сучасних систем дослідники оцінюють запропоновані рішення з допомогою згенерованих мереж. Велика розмірність системи призводить до проблем у процесі моделювання та управління. Для вирішення цих питань потрібні методи зменшення розмірності складних систем. Агрегація структурної моделі системи полягає в об'єднанні її елементів у підсистеми, унаслідок чого зменшується розмірність системи й обчислювальна складність. **Предметом дослідження** є програмна платформа для оцінювання ефективності агрегації структурної моделі складних систем. **Мета роботи** – розроблення програмної платформи для оцінювання ефективності агрегації структурної моделі складних систем. **Актуальність статті** полягає в тому, що, застосовуючи програмну платформу, можна здійснити агрегацію структурної моделі систем із значною кількістю елементів, а також оцінити ефективність агрегації структурної моделі системи. У роботі передбачалося розв'язання таких **завдань**: розроблення програмної платформи, що складається з модуля генератора системи, модуля агрегації системи, модуля пошуку максимального потоку й модуля статистичного оброблення інформації; агрегація структурної моделі системи з використанням програмної платформи; визначення ефективності агрегації структурної моделі системи із застосуванням програмної платформи. **Результати дослідження**: створено програмну платформу, здійснено агрегацію структурної моделі системи та оцінено ефективність агрегації структурної моделі системи з використанням програмної платформи. **Висновки**: застосовуючи програмну платформу, можна згенерувати систему, здійснити агрегацію структурної моделі системи та виконати пошук максимального потоку; також програмна платформа дає змогу оцінити ефективність агрегації структурної моделі системи; значення максимального потоку є однаковим на двох рівнях системи, тому в цьому разі здійснюється агрегація структурної моделі системи зі збереженням коректності її параметрів.

Ключові слова: програмна платформа; агрегація структурної моделі; складні системи; генератор систем; максимальний потік.

Вступ

Складні системи складаються із значної кількості елементів і зв'язків між ними та спостерігаються в багатьох сферах, наприклад, соціальні, біологічні, екологічні та технологічні системи. Мережі використовуються як загальна модель складних систем [1, 2]. Збільшення розміру мереж призводить до розвитку нових аналітичних підходів до їх подання та оцінювання ефективності. У розробленні таких підходів дослідники стикаються з різними проблемами. Для мереж, що складаються навіть із декількох десятків вершин, досить просто намалювати мережу та відповісти на запитання про її структуру. Це один із простих та основних способів отримати уявлення про структуру мережі. Але це неможливо зробити для мереж із дуже великою кількістю елементів. Існує чимало інструментів візуалізації, що дають змогу структурувати та візуалізувати мережі [3].

Складні системи потребують спеціальних методів аналізу та проектування. Існують такі групи

завдань топологічного аналізу, вирішення яких потребує розвитку математичних методів і обчислювальних ресурсів: розроблення відповідного структурного опису системи для визначення елементів системи, підсистем та зв'язків між ними; створення підходів для визначення характеристик системи, наприклад, окреслення найкоротших шляхів і циклів; еквівалентні перетворення структури системи для зменшення розмірності системи; проектування структури системи [4].

Під час досліджень часто доводиться попередньо оцінювати ефективність топології мережі, здатність мережі витримувати високі навантаження та залишатися в робочому стані. Через великі масштаби сучасних систем створення реальної системи для експерименту практично неможливе. У цьому разі дослідники оцінюють запропоновані рішення за допомогою згенерованих мереж [5].

Існують методи для зменшення розмірності складних систем. Це зумовлено тим, що дуже велика розмірність системи призводить до проблем у процесі моделювання та управління. Агрегація структурної

моделі системи полягає в об'єднанні елементів системи в підсистеми, унаслідок чого зменшується розмірність системи та обчислювальна складність.

Робота присвячена створенню програмної платформи для оцінювання ефективності агрегації структурної моделі системи. Програмна платформа складається з модуля генератора системи, модуля агрегації системи, модуля пошуку максимального потоку й модуля статистичного оброблення інформації. Актуальність дослідження полягає в тому, що завдяки програмній платформі можна здійснити агрегацію структурної моделі системи із значною кількістю елементів, а також оцінити ефективність агрегації структурної моделі системи.

Аналіз літератури

Мережі широко використовуються для подання систем, тому що за допомогою мереж можна проаналізувати функції, динаміку та еволюцію природних і штучних систем [6]. Часто систему не можна подати однією мережею, а замість цього мають бути змодельовані набори мереж, у яких різниця між ними, імовірно, є так само важливою, як і їх загальні риси. Такі випадки вказують на важливість методів для генерації мереж, що основані на даних та охоплюють як ключові характеристики системи, так і реалістичну мінливість, щоб використовувати їх у таких завданнях, як моделювання, аналіз та прийняття рішень. Протягом останніх двох десятиліть була запропонована значна кількість моделей з метою досягнення реалізму для згенерованих мереж.

Важливо, щоб генератор мережі відповідав двом основним критеріям – реалізму й різноманітності. Реалізм має зважати на будь-які властивості мережі, що управляють процесами, такими як функціонування системи, динаміка й еволюція. Реалізм може залежати як від структурних властивостей мережі, так і від особливостей мережі, що формуються. Наприклад, моделі соціальних мереж мають бути здатні відтворювати структурні особливості мережі та соціологічні явища, які виникають, зокрема взаємодія між людьми у співтоваристві, що зумовлено їх потребами й щоденною рутиною. Також час роботи та вимоги до пам'яті генератора мають бути допустимими для дуже великих наборів даних, що можуть містити значну кількість вузлів і дуг.

Автори роботи [6] розробляють та оцінюють швидкий генератор, що створює реалістичні та масштабовані копії складних мереж. У праці реалізм

описується двома способами: відповідність оригінальному графу за набором важливих структурних властивостей і відповідність поведінці під час виконання різних алгоритмів графа. Запропонований авторами генератор здатний генерувати мережі, які аналогічні оригінальній мережі з погляду важливих структурних показників та які призводять до аналогічного часу роботи багатьох алгоритмів аналізу графів і мереж. Використовуючи цей генератор, можна відтворити наявну мережу й масштабувати згенеровану версію, наприклад, щоб тестувати алгоритми на великих наборах даних, де вони не доступні.

У роботі [7] автори пропонують гнучкий та розширюваний фреймворк, щоб зрозуміти, як властивості різних типів мереж змінюються з різною кількістю дуг і вершин. Цей підхід дає змогу моделювати три класичні мережні моделі та налаштовувати параметри моделі й розмір мережі. У згаданому дослідженні автори аналізують, як поведуться різні властивості мережі залежно від її типу та як змінюються властивості залежно від розміру мережі. Також порівнюються властивості трьох мережних моделей: модель безмасштабної мережі, модель мережі малого світу та модель випадкової мережі. Автори вимірювали три типи мережних властивостей: центральність, середній найкоротший шлях і глобальний коефіцієнт кластеризації.

Методи моделювання систем використовуються для оцінювання надійності та продуктивності системи, оскільки вони імітують її роботу, дають змогу досліджувати вплив зовнішніх факторів на продуктивність системи, а також надають інструмент для аналізу ризиків і прийняття рішень [8].

У зв'язку зі збільшенням розмірів систем, моделі, що описують поведінку мережних систем, здатні мати дуже високу розмірність. Це призводить до проблем під час моделювання та управління через обмежені обчислювальні можливості та ємність сховища. Окреслені проблеми стимулюють розроблення методів зменшення розмірності мережних систем. За останні десятиліття з'явилося чимало теорій і методів зменшення розмірності моделей систем [9].

Агрегація структурної моделі системи полягає в об'єднанні елементів у підсистеми, унаслідок чого створюється другий рівень системи. Завдяки агрегації структурної моделі системи зменшується розмірність системи й обчислювальна складність.

У роботі [10] застосовується агрегація елементів мережі масового обслуговування для того, щоб зменшити розмірність мережі та час моделювання. Для моделювання великих систем необхідні методи з відповідним балансом точності та швидкості, оскільки моделювання реальної системи може потребувати багато часу.

У праці [11] агрегація використовується для пошуку найкоротших шляхів у графі. Це може застосовуватися для супутникової навігації, щоб реагувати на оновлення дорожнього трафіку в режимі реального часу.

Мета статті – розроблення програмної платформи для оцінювання ефективності агрегації структурної моделі складних систем. У дослідженні розв'язуються такі завдання: розроблення програмної платформи, що складається з модуля генератора системи, модуля агрегації системи, модуля пошуку максимального потоку й модуля статистичного оброблення інформації; здійснення агрегації структурної моделі системи з використанням програмної платформи; оцінювання ефективності агрегації структурної моделі системи із застосуванням програмної платформи.

Програмна платформа

У попередній роботі [12] здійснена агрегація структурної моделі системи з незначною кількістю елементів. Але для систем із великою кількістю елементів неможливо здійснити агрегацію структурної моделі системи без застосування програмних інструментів. Тому була створена програмна платформа мовою програмування C#, що складається з чотирьох модулів: модуля генератора системи, модуля агрегації системи, модуля пошуку максимального потоку та модуля статистичного оброблення інформації.

1. У модулі генератора створюється система із заданими параметрами.

2. У модулі агрегації здійснюється агрегація структурної моделі системи. Елементи системи об'єднуються в підсистеми, створюються фіктивні контакти підсистем і зв'язки між підсистемами. Отже, унаслідок роботи модуля агрегації створюється другий рівень системи.

3. У модулі пошуку максимального потоку здійснюється пошук максимального потоку для рівня системи. Якщо виконати це завдання для різних рівнів системи, то можна побачити, чи змінився

максимальний потік після агрегації структурної моделі системи, тобто чи вплинула агрегація на параметри системи.

4. У модулі статистичного оброблення інформації аналізуються згенеровані системи та їх параметри для подальшої генерації систем зі схожими параметрами.

Структура програмної платформи зображена на рис. 1.

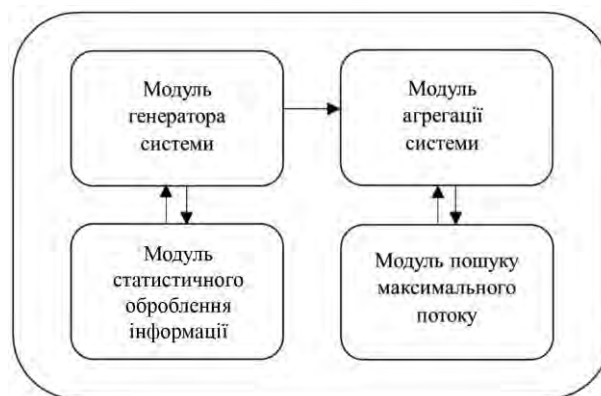


Рис. 1. Структура програмної платформи

Модуль генератора системи

У модулі генератора системи створюється система, подібна до тих, що проаналізовані раніше в роботах [12, 13]. У генератор системи передається кількість її елементів. Потім генерується граф, що має такі вершини: джерело, C_1, C_2, \dots, C_n , стік, де n – кількість елементів системи, джерело та стік – це нульовий елемент системи, що є зовнішнім середовищем.

У циклі перебираються вершини графа та створюються дуги. Для генерації графа виконуються такі умови: мінімум одна дуга виходить з вершини; мінімум одна дуга входить у вершину; мінімум одна дуга виходить із джерела; мінімум одна дуга входить у стік; дуга не може входити в джерело; дуга не може виходити зі стоку; дуга не може виходити та входити в ту саму вершину.

На етапі агрегації елементи системи будуть об'єднані в підсистеми. З огляду на це в модулі генератора системи виконується така умова: має бути зв'язок між елементами всередині підсистеми.

Далі визначається, чи є такі вершини графа, до яких не входить дуга. Якщо такі вершини існують, то генеруються дуги, що входять у ці вершини. Отже, формується колекція дуг графа *Edges*.

На наступному етапі створюється об'єкт класу *SystemLevel*, що є початковою системою, або перший рівень системи. У циклі *foreach* здійснюється перебір дуг графа й виконується такий алгоритм:

1) якщо ще не існує елемента в системі з таким номером, як номер вершини, до якої входить дуга, то створюється елемент системи й додається до колекції *Components* класу *SystemLevel*;

2) знаходиться елемент системи з таким номером, як номер вершини, до якої входить дуга, та створюється вхідний контакт елемента;

3) якщо ще не існує елемента в системі з таким номером, як номер вершини, з якої виходить дуга, то створюється елемент системи та додається до колекції *Components* класу *SystemLevel*;

4) знаходиться елемент системи з таким номером, як номер вершини, з якої виходить дуга, і створюється вихідний контакт елемента;

5) створюється зв'язок між елементами системи й додається до колекції *Connections* класу *SystemLevel*.

Отже, формується об'єкт класу *SystemLevel*, що є початковою системою, або перший рівень системи та містить колекцію елементів системи *Components* і колекцію зв'язків між елементами системи *Connections*.

Модуль агрегації системи

У модуль агрегації системи передається початкова система, отримана в модулі генератора системи. Система подана класом *SystemLevel* і є першим рівнем системи. Клас *SystemLevel* містить властивості *Components* і *Connections*. *Components* – це колекція елементів системи; *Connections* – колекція зв'язків між елементами системи.

Елементи в системі об'єднуються таким чином: $S_{\mu 0} = \{C_0\}$, $S_{\mu 1} = \{C_1, C_2\}$, ..., $S_{\mu m} = \{C_{n-1}, C_n\}$, де n – кількість елементів у системі; m – кількість підсистем.

Елементи об'єднуються по два до кожної підсистеми, нульовий елемент додається в окрему підсистему. Якщо в системі непарна кількість елементів, то останній елемент додається один у підсистему.

Підсистема подана класом *Subsystem* і є елементом другого рівня системи й водночас системою, що містить декілька елементів. Клас *Subsystem* наслідується від класу *Component* і містить

властивість *InternalComponents*, що є колекцією елементів, які входять у цю підсистему.

На етапі створення підсистеми формуються вхідні та вихідні контакти підсистеми. У циклі *foreach* здійснюється перебір елементів, що входять у підсистему, та за таким алгоритмом створюються вихідні контакти підсистеми:

– знаходиться вихідний контакт елемента та зв'язок на першому рівні системи, який виходить з цього контакту;

– якщо цей зв'язок входить до елемента, розміщеного в цій підсистемі, то вихідний контакт підсистеми не створюється;

– якщо цей зв'язок входить до елемента, що знаходиться в іншій підсистемі, то створюється вихідний контакт підсистеми.

Далі в цьому циклі створюються вхідні контакти підсистеми за таким алгоритмом:

– знаходиться вхідний контакт елемента та знаходиться зв'язок на першому рівні системи, який входить до цього контакту;

– якщо цей зв'язок виходить з елемента, розміщеного в цій підсистемі, то вхідний контакт підсистеми не створюється;

– якщо цей зв'язок виходить з елемента, розміщеного в іншій підсистемі, то створюється вхідний контакт підсистеми.

На наступному етапі створюються зв'язки між підсистемами. У циклі *foreach* здійснюється перебір підсистем і виконується такий алгоритм:

– знаходиться вихідний контакт цієї підсистеми, знаходиться зв'язок першого рівня системи, що відповідає цьому вихідному контакту підсистеми, та знаходиться вхідний контакт елемента першого рівня, до якого входить цей зв'язок;

– потім серед усіх вхідних контактів підсистем знаходиться такий вхідний контакт підсистеми, що відповідає вхідному контакту елемента першого рівня.

Отже, створюється зв'язок між вихідним контактом підсистеми та знайденим вхідним контактом іншої підсистеми й додається до колекції *Connections*.

Унаслідок роботи модуля створюється об'єкт класу *SystemLevel*, що є перетвореною системою, або другий рівень системи. Властивість *Components* класу *SystemLevel* містить колекцію підсистем; властивість *Connections* містить колекцію зв'язків між підсистемами.

Модуль пошуку максимального потоку

Унаслідок агрегації структурної моделі системи здійснюється перетворення структури системи, тому треба перевірити, чи збереглися параметри системи на другому рівні системи. Для цього можна розв'язати задачу про максимальний потік. Сутність задачі полягає в тому, щоб отримати значення максимального потоку мережі для заданої топології [14]. Задача про максимальний потік формулюється таким чином: необхідно знайти максимально можливе для цієї мережі значення сумарного потоку між джерелом і стоком за умови заданих пропускних здатностей дуг мережі. Для вирішення задачі про пошук максимального потоку використовується теорема, згідно з якою максимально можливе значення сумарного потоку на кінцевих дугах дорівнює мінімальній пропускній здатності обраного розрізу [15]. Під пропускною здатністю розрізу мається на увазі сума пропускних здатностей дуг, що утворюють розріз.

У модуль пошуку максимального потоку передається об'єкт класу *SystemLevel*, що є системою та містить колекцію елементів системи й колекцію зв'язків між елементами системи. Колекція зв'язків між елементами системи перетворюється у двовимірний масив, що подає граф, та відповідно до алгоритму розв'язується задача про максимальний потік.

Алгоритм Форда – Фалкерсона передбачає такі кроки:

- пошук на графі ланцюга з джерела в стік;
- для кожної дуги ставиться максимально можливе значення потоку з джерела в стік, значення потоку не може бути більшим, ніж вага дуги;
- якщо значення потоку дорівнює вазі дуги, то така дуга є насиченою та крізь неї вже не можна пройти під час вибору ланцюгів у графі;
- алгоритм завершується тоді, коли перехід із джерела в стік стане неможливим.

Після завершення дії алгоритму значення максимального потоку дорівнює сумі потоків усіх дуг, інцидентних стоку графа.

Унаслідок роботи модуля маємо значення максимального потоку. Якщо значення максимального потоку буде однаковим для першого й другого рівня системи, то можна зробити висновок, що в цьому разі здійснюється агрегація структурної моделі системи зі збереженням коректності її параметрів.

Процес пошуку максимального потоку зображено на прикладі згенерованої системи (рис. 2).

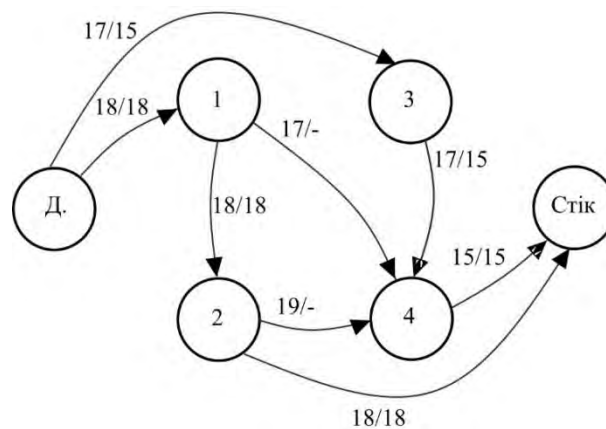


Рис. 2. Обхід графа згенерованої системи

Результат обходу графа поданий у табл. 1.

Таблиця 1. Результат обходу графа

№	Шлях	Значення потоку
1	джерело – 1 – 2 – стік	18
2	джерело – 3 – 4 – стік	15

Значення максимального потоку дорівнює сумі потоків усіх дуг, інцидентних стоку графа, що в наведеному прикладі дорівнює 33.

Тестування модулів програмної платформи

Було проведено тестування програмної платформи на прикладі систем із незначною кількістю елементів за допомогою *Unit Tests*.

Для тестування модулів платформи використовувались системи, для яких вже були здійснені агрегація структурної моделі системи та пошук максимального потоку, наприклад, система, розглянута в роботі [12]. У тесті така система у вигляді об'єкта класу *SystemLevel* передавалася в модуль агрегації. Перетворена система, отримана внаслідок роботи модуля, має збігатися із системою, отриманою після агрегації для цього прикладу раніше.

Для тестування модуля пошуку максимального потоку в тесті в цей модуль передавались перший і другий рівні системи. Унаслідок роботи модуля пошуку максимального потоку значення максимального потоку на першому та другому рівнях системи мають

збігатися з розрахунками, проведеними для цієї системи раніше.

Було виміряно час роботи модулів програмної платформи для систем із різною кількістю елементів, наприклад, що містять 50, 250 та 500 елементів. Також виміряно загальний час роботи програмної платформи.

Оцінювання ефективності агрегації структурної моделі системи

Був проведений експеримент із використанням програмної платформи, основною метою якого є оцінювання ефективності агрегації структурної моделі системи та ефективності самої програмної платформи. Критеріями ефективності агрегації структурної моделі системи є зниження обчислювальної складності системи та підвищення швидкодії.

Алгоритм експерименту передбачає такі етапи:

- модуль генератора системи: генерація системи із заданою кількістю елементів;
- модуль пошуку максимального потоку: розв'язання задачі пошуку максимального потоку для згенерованої системи;
- модуль агрегації системи: агрегація структурної моделі системи;
- модуль пошуку максимального потоку: вирішення задачі пошуку максимального потоку для перетвореної системи.

Був проведений експеримент для систем, що містять 50, 250 та 500 елементів. У процесі експерименту алгоритм був виконаний 100 разів. Значення максимального потоку на першому та другому рівнях системи було однаковим.

Середнє значення часу роботи в секундах модуля генератора системи, модуля агрегації системи, модуля пошуку максимального потоку на першому рівні системи та модуля пошуку максимального потоку на другому рівні системи зображено на рис. 3 та 4.

За результатами експерименту можна зробити висновок, що програмна платформа дає змогу згенерувати систему із заданою кількістю елементів, здійснити агрегацію структурної моделі системи та розв'язати задачу про пошук максимального потоку для кожного рівня системи. У процесі експерименту значення максимального потоку було однаковим для згенерованої та перетвореної системи, тому впливає висновок, що в цьому разі

здійснюється агрегація структурної моделі системи зі збереженням коректності її параметрів.

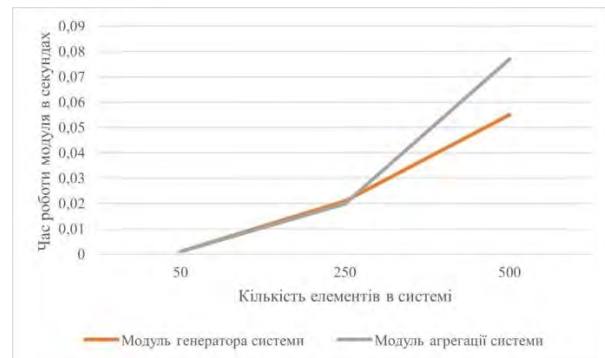


Рис. 3. Час роботи модуля генератора й модуля агрегації

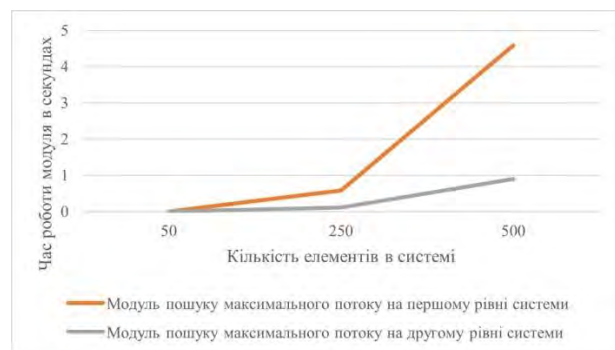


Рис. 4. Час роботи модуля пошуку максимального потоку на першому та другому рівні системи

Середнє значення часу роботи в секундах програмної платформи подано на рис. 5.

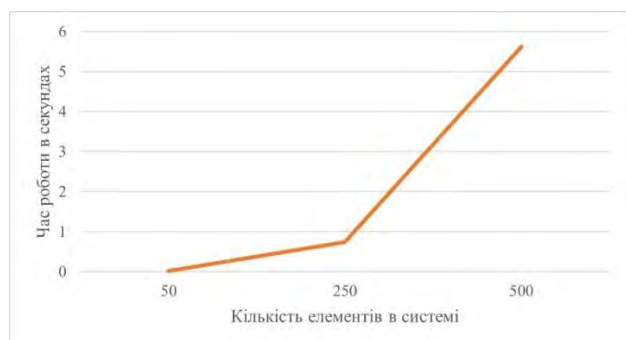


Рис. 5. Час роботи програмної платформи

Час роботи модуля генератора системи, модуля агрегації системи та модуля пошуку максимального потоку на другому рівні системи менший, ніж час роботи модуля пошуку максимального потоку на першому рівні системи. Це зумовлено тим, що алгоритм пошуку максимального потоку працює

повільніше для системи із значною кількістю елементів. Час роботи модуля генератора для системи, що містить 500 елементів, дорівнює 0,055 с, час роботи модуля агрегації – 0,077 с.

Висновки та перспективи подальшого розвитку

У роботі розроблено програмну платформу мовою програмування C# та оцінено ефективність агрегації структурної моделі системи з використанням програмної платформи. Програмна платформа містить модуль генератора системи, модуль агрегації системи, модуль пошуку максимального потоку та модуль статистичного оброблення інформації. У модуль генератора системи передається кількість елементів системи. Унаслідок роботи модуля формується початкова система, або перший рівень системи. Початкова система передається в модуль агрегації системи, у якому здійснюється агрегація структурної моделі системи. Елементи системи об'єднуються в підсистеми, формуються контакти підсистем і зв'язки між підсистемами. За результатами роботи модуля отримується перетворена система, або другий рівень

системи. У модуль пошуку максимального потоку передається система, і внаслідок роботи модуля отримується значення максимального потоку.

Проведено дослідження із використанням програмної платформи для оцінювання ефективності агрегації структурної моделі системи та ефективності самої програмної платформи. Експеримент проведено для систем, що містять 50, 250 та 500 елементів. За результатами дослідження впливає висновок, що, застосовуючи програмну платформу, можна згенерувати систему, здійснити агрегацію структурної моделі системи та розв'язати задачу про пошук максимального потоку. Також програмна платформа дає змогу оцінити ефективність агрегації структурної моделі системи. Значення максимального потоку є однаковим для початкової та перетвореної системи, тому можна зробити висновок, що в цьому разі здійснюється агрегація структурної моделі системи зі збереженням коректності її параметрів.

Подальша робота полягає у виконанні таких завдань: дослідження систем із кількістю вершин від 500 до 1000; оцінювання ефективності самої програмної платформи; використання агрегації структурної моделі системи у сфері комп'ютерних мереж.

Список літератури

1. Newman M. Networks: An Introduction, Oxford University Press. 2010. 1042 p.
URL:[https://www.scirp.org/\(S\(351jmbntvnsjtladkpozje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1892198](https://www.scirp.org/(S(351jmbntvnsjtladkpozje))/reference/ReferencesPapers.aspx?ReferenceID=1892198)
2. Jinhu Lu, Guanrong Chen, Maciej Ogorzalek, Ljiljana Trajkovic Theory and Applications of Complex Networks: Advances and Challenges. *Proceedings of the IEEE International Symposium on Circuits and Systems*. 2013. P. 184–198. DOI:10.1109/ISCAS.2013.6572335
3. Hackl J. Tikz-network: a LaTeX library for vizualizing complex networks. 6th International Conference on Complex Networks & Their Applications. Lyon, France. 2017. URL: <https://github.com/hackl/tikz-network>
4. Gorbachov V., Batiia A. K., Ponomarenko O., Romanenkov Y. Formal transformations of structural models of complex network systems. *Proceedings of the IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018*. Kyiv, Ukraine. 2018. P. 473–477. DOI: <https://doi.org/10.1109/DESSERT.2018.8409175>
5. Ponomarenko O., Gorbachov V., Batiia A. K., Kotkova O. The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies. *IEEE East-West Design & Test Symposium (EWDTS), Conference proceedings*. Batumi, Georgia. 2019. P. 513–516. DOI: <https://doi.org/10.1109/EWDTS.2019.8884421>
6. Staudt C., Hamann M., Gutfraind A., Safro I., Meyerhenke H. Generating realistic scaled complex networks. *Applied Network Science*. 2017. Vol. 2(36). DOI: <https://doi.org/10.1007/s41109-017-0054-z>
7. Ashraf A., Budka M., Musial K. NetSim – The framework for complex network generator / A. Ashraf et al. *Procedia Computer Science*. 2018. Vol. 126. DOI: <https://doi.org/10.1016/j.procs.2018.07.289>
8. George-Williams H., Santhosh T. V., Patelli E. Simulation Methods for the Analysis of Complex Systems. In: Aslett L. J. M., Coolen F. P. A., De Bock J. Uncertainty in Engineering, *SpringerBriefs in Statistics*, Springer. 2022. P. 95–113. DOI: https://doi.org/10.1007/978-3-030-83640-5_7
9. Cheng X., Scherpen J. Model Reduction Methods for Complex Network Systems. *Annual Review of Control Robotics and Autonomous Systems*. 2020. Vol. 4. P. 425–453. DOI: <https://doi.org/10.1146/annurev-control-061820-083817>
10. Shortle J. F., Mark B. L., Gross D. Reduction of closed queueing networks for efficient simulation. *ACM Transactions on Modeling and Computer Simulation*. 2009. Vol. 19. № 3. Article 10. P. 1–22. DOI: <https://doi.org/10.1145/1540530.1540531>

11. MacKay R. S. Hierarchical aggregation of complex systems. *Proceedings of the ECCS'11*. Vienna, Austria. 2011.
12. Пономаренко О. Є., Горбачов В. О. Агрегація структурної моделі складних мережних систем. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2023. Т. 1. № 71. С. 138–144. DOI: <https://doi.org/10.26906/SUNZ.2023.1.138>
13. Gorbachov V., Sytnikov D., Ryabov O., Batiaa A. K., Ponomarenko O. Dimension Reduction for Network Systems Using Structure Model Aggregation. *International Journal of Design & Nature and Ecodynamics*. 2020. Vol. 15. No. 1. P. 13–23. DOI: <https://doi.org/10.18280/ijdne.150103>
14. Cormen T., Leiserson C., Rivest R., Stein C. Introduction to algorithms. 2009. 3rd ed. 1313 p. URL: https://pd.daffodilvarsity.edu.bd/course/material/book-430/pdf_content
15. Ford L., Fulkerson D. Maximal Flow Through a Network. *Canadian Journal of Mathematics*. 1956. Vol. 8. P. 399–404. URL: <https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/maximal-flow-through-a-network/5D6E55D3B06C4F7B1043BC1D82D40764>

References

1. Newman M. Networks: An Introduction, Oxford University Press. 2010. 1042 p.
URL: [https://www.scirp.org/\(S\(351jmbntvnsjt1aadkpozje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1892198](https://www.scirp.org/(S(351jmbntvnsjt1aadkpozje))/reference/ReferencesPapers.aspx?ReferenceID=1892198)
2. Lu, J., Chen, G., Ogorzalek, M., Trajkovic, L. (2013), "Theory and Applications of Complex Networks: Advances and Challenges", *Proceedings of the IEEE International Symposium on Circuits and Systems*. P. 184-198. DOI:10.1109/ISCAS.2013.6572335
3. Hackl, J. "Tikz-network: a LaTeX library for vizualizing complex networks", 6th International Conference on Complex Networks & Their Applications, Lyon, France. 2017. available at: <https://github.com/hackl/tikz-network>
4. Gorbachov, V., Batiaa, A. K., Ponomarenko, O., Romanenkov, Y. (2018), "Formal transformations of structural models of complex network systems", *Proceedings of the IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018*, Kyiv, Ukraine. P. 473–477. DOI: <https://doi.org/10.1109/DESSERT.2018.8409175>
5. Ponomarenko, O., Gorbachov, V., Batiaa, A. K., Kotkova, O. (2019), "The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies", *IEEE East-West Design & Test Symposium (EWDTS), Conference proceedings*, Batumi, Georgia. P. 513–516. DOI: <https://doi.org/10.1109/EWDTS.2019.8884421>
6. Staudt, C., Hamann, M., Gutfraind, A., Safro, I., Meyerhenke, H. (2017), "Generating realistic scaled complex networks", *Applied Network Science*, Vol. 2(36). DOI: <https://doi.org/10.1007/s41109-017-0054-z>
7. Ashraf, A., Budka, M., Musial, K. (2018), "NetSim – The framework for complex network generator", *Procedia Computer Science*, Vol. 126. DOI: <https://doi.org/10.1016/j.procs.2018.07.289>
8. George-Williams, H., Santhosh, T. V., Patelli, E. (2022), "Simulation Methods for the Analysis of Complex Systems", *Uncertainty in Engineering, SpringerBriefs in Statistics*, Springer. P. 95–113. DOI: https://doi.org/10.1007/978-3-030-83640-5_7
9. Cheng, X., Scherpen, J. (2020), "Model Reduction Methods for Complex Network Systems", *Annual Review of Control Robotics and Autonomous Systems*, Vol. 4. P. 425–453. DOI: <https://doi.org/10.1146/annurev-control-061820-083817>
10. Shortle, J. F., Mark, B. L., Gross, D. (2009), "Reduction of closed queueing networks for efficient simulation", *ACM Transactions on Modeling and Computer Simulation*, Vol. 19, No. 3, Article 10. P. 1–22. DOI: <https://doi.org/10.1145/1540530.1540531>
11. MacKay, R. S. (2011), "Hierarchical aggregation of complex systems", *Proceedings of the ECCS'11*, Vienna, Austria.
12. Ponomarenko, O., Gorbachov, V. (2023), "Aggregation of structural model of complex network systems", ["Ahrehatsiia strukturnoi modeli skladnykh merezhnykh system"], *Control, Navigation and Communication Systems. Academic Journal*, Vol. 1 (71), P. 138–144. DOI: <https://doi.org/10.26906/SUNZ.2023.1.138>
13. Gorbachov, V., Sytnikov, D., Ryabov, O., Batiaa, A. K., Ponomarenko, O. (2020), "Dimension Reduction for Network Systems Using Structure Model Aggregation", *International Journal of Design & Nature and Ecodynamics*, Vol. 15, No. 1, P. 13–23. DOI: <https://doi.org/10.18280/ijdne.150103>
14. Cormen, T., Leiserson, C., Rivest, R., Stein, C. "Introduction to algorithms", 3rd ed. 2009. 1313 p. available at: https://pd.daffodilvarsity.edu.bd/course/material/book-430/pdf_content
15. Ford, L., Fulkerson, D. (1956), "Maximal Flow Through a Network", *Canadian Journal of Mathematics*, Vol. 8, P. 399–404. available at: <https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/maximal-flow-through-a-network/5D6E55D3B06C4F7B1043BC1D82D40764>

Відомості про авторів / About the Authors

Пономаренко Ольга Євгенівна – Харківський національний університет радіоелектроніки, аспірант кафедри електронних обчислювальних машин, Харків, Україна; e-mail: olha.ponomarenko@nure.ua; ORCID ID: <https://orcid.org/0009-0002-4634-6552>

Горбачов Валерій Олександрович – кандидат технічних наук, професор, Харківський національний університет радіоелектроніки, професор кафедри електронних обчислювальних машин, Харків, Україна; e-mail: valeriy.gorbachov@nure.ua; ORCID ID: <https://orcid.org/0000-0003-3423-2371>

Ponomarenko Olha – Kharkiv National University of Radio Electronics, postgraduate student at the Department of Electronic Computers, Kharkiv, Ukraine.

Gorbachov Valeriy – PhD (Engineering Sciences), Professor, Kharkiv National University of Radio Electronics, Professor at the Department of Electronic Computers, Kharkiv, Ukraine.

THE SOFTWARE PLATFORM FOR EVALUATION OF THE EFFECTIVENESS OF STRUCTURAL MODEL AGGREGATION OF COMPLEX SYSTEMS

Complex systems have a large dimension, consist of a large number of elements and connections between them. Networks are used to represent complex systems. Due to the high dimension of modern systems, researchers evaluate proposed solutions using generated networks. The high dimension of the system leads to problems in modeling and control. To solve these problems, methods for reducing the dimension of complex systems are required. Aggregation of the structural model of the system consists in combining elements of the system into subsystems and as a result dimension of the system and computational complexity are reduced. **The subject matter of research** is a software platform for evaluation of the effectiveness of structural model aggregation of complex systems. **The goal of the work** is to develop the software platform for evaluation of the effectiveness of structural model aggregation of complex systems. **The relevance of the work** lies in the fact that using the software platform it is possible to make the structural model aggregation of systems with a large number of elements and it is also possible to evaluate the effectiveness of structural model aggregation of the system. **The following tasks were solved** in the work: development of the software platform which consists of a system generator module, a system aggregation module, a maximum flow searching module and a statistical data processing module; structural model aggregation of the system using the software platform; evaluation of the effectiveness of structural model aggregation of the system using the software platform. **As a result of the work** the software platform was created, the structural model aggregation of the system was made and the effectiveness of structural model aggregation of the system using the software platform was evaluated. The studies allow us to **conclude**: using the software platform it is possible to generate a system, make the structural model aggregation of the system and solve the problem of the maximum flow searching; the software platform also allows to evaluate the effectiveness of structural model aggregation of the system; the value of the maximum flow is the same at two levels of the system, so in this case the structural model aggregation of the system is made while preserving the correctness of its parameters.

Keywords: software platform; structural model aggregation; complex systems; system generator; maximum flow.

Бібліографічні описи / Bibliographic descriptions

Пономаренко О. Є., Горбачов В. О. Програмна платформа для оцінювання ефективності агрегації структурної моделі складних систем. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 3 (25). С. 79–87. DOI: <https://doi.org/10.30837/ITSSI.2023.254.079>

Ponomarenko, O., Gorbachov, V. (2023), "The software platform for evaluation of the effectiveness of structural model aggregation of complex systems", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (25), P. 79–87. DOI: <https://doi.org/10.30837/ITSSI.2023.254.079>