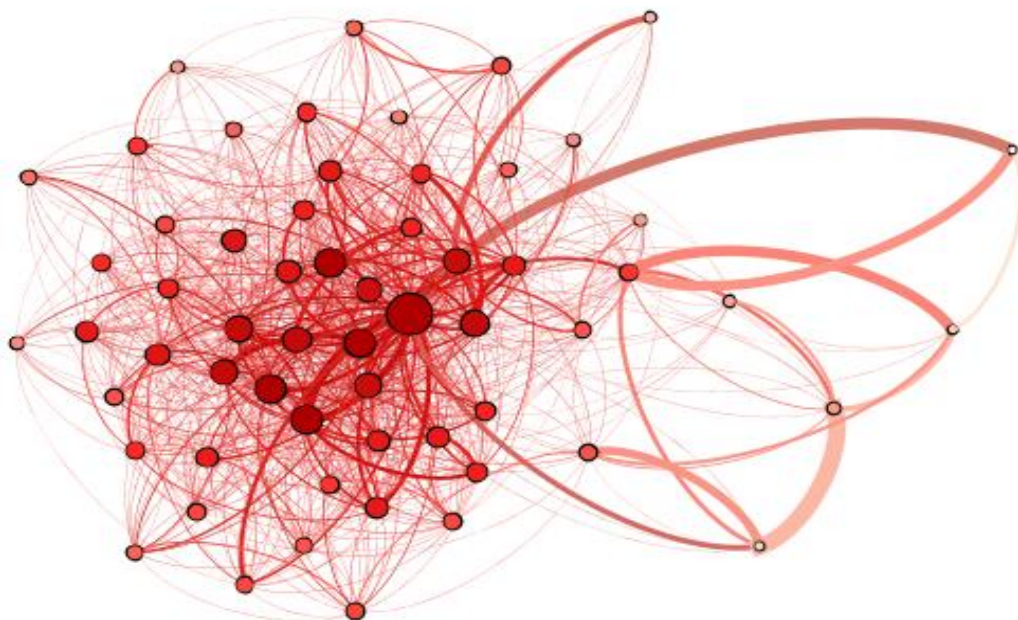


Д.В. Ланде, І.Ю. Субач

**ВІЗУАЛІЗАЦІЯ ТА АНАЛІЗ
МЕРЕЖЕВИХ СТРУКТУР**

Навчальний посібник



Київ – 2020

Міністерство освіти і науки України
Інститут спеціального зв'язку та захисту інформації
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»

Д.В. Ланде, І.Ю. Субач

ВІЗУАЛІЗАЦІЯ ТА АНАЛІЗ МЕРЕЖЕВИХ СТРУКТУР

*Розглянуто Вченою радою ІСЗЗІ КПІ ім. Ігоря Сікорського
для використання у навчальному процесі з підготовки
фахівців третього (наукового) рівня вищої освіти
зі спеціальності «Комп'ютерні науки»*

Київ
ІСЗЗІ КПІ ім. Ігоря Сікорського
2020

УДК 004.942
ББК 32.973я7
В 41

*Розглянуто Вченою радою
ІСЗЗІ КПІ ім. Ігоря Сікорського
(Протокол № 3 від 29.04.2020)*

Рецензенти:

*О.Г. Оксіюк, д.т.н., проф.
В.В. Вишнівський, д.т.н., проф.*

В 41

Д.В. Ланде, І.Ю. Субач. Візуалізація та аналіз мережевих структур.
Навчальний посібник. – К.: ІСЗЗІ КПІ ім. Ігоря Сікорського, 2020. – 79 с.

ISBN 978-966-2577-14-3

Сучасне інформаційне середовище не уявляється без даних, що утворюють складні мережеві структури. Для їх обробки та інтеграції необхідні нові методи та засоби. Доступна й проста у встановленні система Gephi може допомогти, як під час підготовки матеріалів наукових досліджень до доповідей і презентацій, так і для підтримки прийняття важливих рішень під час виконання завдань фахівцями у різних сферах людської діяльності. Для закріплення навчального матеріалу наприкінці кожної глави видання наведені питання для самостійної роботи.

Навчальне видання призначене слухачам інституту, що здобувають освіту за третім (науковим рівнем) за спеціальністю “Комп’ютерні науки” і вивчають дисципліну “Методологічні основи забезпечення кібернетичної безпеки держави” циклу професійної підготовки. Може бути також корисна всім, хто самостійно хоче засвоїти поняття та методи візуалізації та аналізу складних мережевих структур – курсантам, студентам, аспірантам, спеціалістам технічних спеціальностей та програмістам.

УДК 004.942
ББК 32.973я7

© Д.В. Ланде, 2020
© І.Ю. Субач, 2020
© ІСЗЗІ КПІ ім. Ігоря Сікорського, 2020

ISBN 978-966-2577-14-3

ЗМІСТ

ВСТУП	5
1. СКЛАДНІ МЕРЕЖІ. ОСНОВНІ ВІДОМОСТІ	7
1.1. Параметри мереж	7
1.2. Коефіцієнт глобальної ефективності	9
1.3. Коефіцієнт кластерності	10
1.4. Посередництво	11
1.5. Модулярність	11
1.6. Феномени мереж	12
1.7. Модель випадкової мережі Ердеша-Реньї	15
1.8. Модель випадкової мережі Барабаші-Альберт	16
1.9. Ранжування вузлів мереж	17
Питання для сомоперевірки	19
2. ІНСТРУМЕНТИ ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗУ МЕРЕЖ	20
2.1. <i>uDraw (Graph)</i>	20
2.2. <i>Graph Online</i>	22
2.3. <i>Graphviz</i>	24
2.4. <i>Social Network Visualizer</i>	30
2.5. Формат <i>GraphML</i>	35
Питання для сомоперевірки	40
3. ЗАГАЛЬНІ ВІДОМОСТІ ПРО GERHI. ІНТЕРФЕЙС КОРИСТУВАЧА	41
3.1. Загальні відомості про програму	41
3.2. Основні розділи	43
3.3. Вторинні вікна-вкладки	45
3.4. Створення графа в <i>Gephi</i>	46
3.5. Створення графа через інтерфейс «Граф»	48

3.6. Створення графа через інтерфейс «Лабораторія даних»	50
3.7. Експорт даних графа з зовнішнього файлу	53
Питання для сомоперевірки	54
4. КОМПОНОВКА ТА РАНЖУВАННЯ	55
4.1. Компонівка тестового графу	55
4.2. Укладка графу <i>Les Miserables</i>	56
4.3. Ранжування і статистика	59
4.4. Фільтрація мережі	61
Питання для самоперевірки.....	62
5. РЕАЛЬНІ МЕРЕЖІ. ВІЗУАЛІЗАЦІЯ ТА АНАЛІЗ	63
5.1. Тестові мережеві дані програми <i>Gephi</i>	63
5.2. Терористична мережа <i>Mali</i>	68
5.3. Мережа понять, пов'язаних з початковим, по <i>Wikipedia</i>	70
5.4. Мережа співавторства вчених	73
Питання для сомоперевірки	75
Література	76
Предметний покажчик	78

ВСТУП

Даний навчальний посібник присвячений вивченню сучасних засобів візуалізації та аналізу мережевих структур – настільних додатків, серед яких у теперішній час найбільш популярною є система *Gephi*.

Метою даного навчального посібника – є надання тим, хто навчається необхідних теоретичних відомостей та практичних навичок щодо обробки даних, які створюють мережеві структури. Сучасне інформаційне середовище не уявляється без таких даних. Для їх обробки та інтеграції необхідні нові методи та засоби. Доступна й проста у встановленні система *Gephi* може допомогти, як під час підготовки матеріалів до доповідей і презентацій, так і для підтримки прийняття важливих рішень.

Матеріал орієнтований на спеціалістів, що працюють у галузі інформаційних технологій, інформаційної та кібернетичної безпеки. Матеріал не перезавантажений абстрактними математичними формулами та розрахунками. Разом з тим, у сучасних системах аналізу та візуалізації мереж широко використовуються такі поняття, як степені вершин, ранжування, кластеризація, модульність, алгоритми укладки графів та ін.

Перша глава присвячена основним відомостям, поняттям та їх поясненням з теорії складних мереж (*Complex Networks*). Спостерігаємий у теперішній час перехід до обробки даних великих об'ємів (*Big Data*), обумовив перехід від теорії графів до концепції *Complex Networks*. Саме ця теорія стала фундаментом, на якому базуються алгоритми таких систем, як *Gephi*.

Друга глава присвячена опису програм, що є альтернативними *Gephi*, тобто інструментам аналізу та візуалізації мережевих структур, які є вільно доступними до використання. Наведено типових представників цього класу програм, описано їх можливості, тобто представлена «екосистема», з якої користувач може вибрати собі необхідний інструмент. Проте, сьогодні лідером у сфері настільних додатків є *Gephi*.

Третя глава присвячена викладенню загальної інформації про систему *Gephi*, опису можливостей та інтерфейсу користувача, створенню мережеских об'єктів у середовищі *Gephi*, основним режимам роботи у цьому програмному середовищі.

Четверта глава присвячена питанням візуалізації, упорядкування мережеских структур, статистичного аналізу мереж, алгоритмам укладки графів і фільтрації.

Остання, **п'ята глава** присвячена обробці даних, візуалізації та аналізу реальних мережеских структур. Також, надається короткий аналіз реальних мережеских структур, що наведено у відомих наборах даних (*Datasets*), які розміщено у мережі Інтернет.

1. СКЛАДНІ МЕРЕЖІ. ОСНОВНІ ВІДОМОСТІ

Останнім часом виділився окремий науковий і практичний напрям - аналіз соціальних мереж (*SNA, Social Networks Analysis*), який базується, з одного боку, на соціології, а з іншого - на теорії складних мереж (*Complex Networks*). У рамках теорії складних мереж вивчаються мережеві характеристики не лише з точки зору топології мереж, але і статистичні феномени, розподіл ваг окремих вузлів і ребер, ефекти протікання і провідності. Попри те, що в розгляд теорії складних мереж потрапляють різні мережі (електричні, транспортні, інформаційні), найбільший внесок у розвиток цієї теорії внесли дослідження саме соціальних мереж.

У теорії складних мереж виділяють три основні напрями:

- дослідження статистичних властивостей, які характеризують поведінку мереж;
- створення моделей мереж;
- прогнозування поведінки мереж при зміні структурних властивостей.

1.1. Параметри мереж

У прикладних дослідженнях найчастіше застосовуються такі типи для мережевого аналізу характеристики, як розмір мережі, мережева щільність, міра центральності та інші. При аналізі складних мереж, як і в теорії графів досліджуються:

- параметри окремих вузлів;
- параметри мережі в цілому;
- мережеві підструктури.

Для окремих вузлів виділяють наступні параметри:

- вхідна напівстепень вузла – кількість ребер графа, які входять у вузол;
- вихідна напівстепень вузла – кількість ребер графа, які виходять з вузла;
- середня відстань від цього вузла до інших;

- ексцентричність (eccentricity) – найбільша з геодезичних відстаней (мінімальних відстаней між вузлами) від цього вузла до інших;
- посередництво (betweenness), що показує, скільки найкоротших шляхів проходить через даний вузол;
- параметри центральності, наприклад, загальна кількість зв'язків даного вузла по відношенню до інших.

Для аналізу мережі в цілому використовують такі параметри, як:

- кількість вузлів;
- кількість ребер;
- середня відстань між вузлами;
- щільність – відношення кількості ребер в мережі до можливої максимальної кількості ребер $n(n-1)/2$ при даній кількості вузлів n ;
- кількість симетричних, транзитивних і циклічних тріад;
- діаметр мережі – найбільша геодезична відстань у мережі і т.д.

Важливою характеристикою мережі є функція розподілу ступенів вузлів $P(k)$ яка визначається як ймовірність того, що довільний вузол мережі i має ступінь $k_i = k$. Мережі, які характеризуються різними $P(k)$, демонструють різну поведінку, $P(k)$ в деяких випадках може бути розподілами Пуассона $(P(k) = e^{-m} m^k / k!)$, де m – математичне очікування), експоненціальним $(P(k) = e^{-k/m})$ або степеневим $(P(k) \sim 1/k^\gamma, k \neq 0, \gamma > 0)$.

Мережі зі степеневим розподілом ступенів вузлів називаються безмасштабними (*Scale Free*). Саме безмасштабні розподіли часто спостерігаються в реальних мережах соціального характеру. При степеневому розподілі можливе існування вузлів з дуже високим ступенем, що практично не спостерігається в мережах з пуассоновським розподілом. Відстань між вузлами визначається як кількість ребер, через які можна

добратися від одного вузла до іншого. Найкоротшим шляхом d_{ij} між вузлами i та j називається найменша відстань між ними. Для усієї мережі можна ввести поняття середнього шляху, як середнього по усіх парах вузлів найкоротшої відстані між ними:

$$l = \frac{2}{n(n+1)} \sum_{i \geq j} d_{ij},$$

де n – кількість вузлів, d_{ij} – найкоротша відстань між вузлами i та j .

Угорськими математиками П. Ердешем (P. Erdős) і А. Реньї (A. Rényi) було показано, що середня відстань між двома вершинами у випадковому графові (модель Ердеша-Реньї) росте як логарифм від числа його вузлів.



П. Ердеш (1913-1996)

1.2. Коефіцієнт глобальної ефективності

Мережа може виявитися незв'язною, тобто знайдуться вузли, відстань між якими виявиться нескінченною. Відповідно, середній шлях, відповідно до приведеної вище формули, буде також нескінченним. Для урахування таких випадків вводиться поняття середнього інверсного шляху між вузлами (його ще називають "глобальною ефективністю мережі"), що розраховується за формулою:

$$il = \frac{2}{n(n-1)} \sum_{i > j} \frac{1}{d_{ij}}.$$

Зворотна величина глобальної ефективності – середнє гармонійне геодезичних відстаней: $h = 1 / il$.

Один із способів знайти критичні компоненти мережі – пошук найбільш уразливих вузлів. Уразливість мережі від вузла може бути визначена як спад глобальної ефективності мережі в разі видалення вузла і всіх суміжних йому ребер з мережі:

$$V_i = (il - il_i) / il,$$

де il – глобальна ефективність вихідної мережі, а il_i – глобальна ефективність після видалення вузла i та всіх суміжних йому ребер.

Упорядкований розподіл вузлів щодо цієї величини пов'язаний зі структурою всієї мережі. Таким чином, вузол, який найбільш сильно впливає на вразливість мережі, займає найвищу позицію в мережевий ієрархії. Міра уразливості мережі – максимальна уразливість від всіх вузлів:

$$V = \max V_i.$$

1.3. Коефіцієнт кластерності

Д. Уоттс (*D. Watts*) і С. Строгатц (*S. Strogatz*) у 1998 році визначили такий мережевий параметр, як коефіцієнт кластерності, який характеризує рівень зв'язності вузлів в мережі, тенденцію до утворення груп взаємопов'язаних вузлів, так званих клік (*clique*).

Для конкретного вузла коефіцієнт кластеризації показує, скільки найближчих сусідів даного вузла є також найближчими сусідами один для одного.



Д. Уоттс



С. Строгатц

Коефіцієнт кластерності може визначатися як для кожного вузла, так і для усієї мережі. Для мережі коефіцієнт кластерності визначається як нормована по кількості вузлів сума відповідних коефіцієнтів для окремих вузлів.

Коефіцієнт кластерності для окремого вузла мережі визначається таким чином. Нехай з вузла виходить k ребер, які з'єднують його з k іншими вузлами, найближчими сусідами. Якщо припустити, що всі найближчі сусіди з'єднані безпосередньо один з одним, то кількість ребер між ними становила б $1/2 \cdot k(k-1)$ – це число, яке відповідає максимально можливій кількості ребер, якими могли б з'єднуватися найближчі сусіди обраного вузла. Відношення реальної кількості ребер, які сполучають найближчих сусідів цього вузла i до максимально можливого (такому, при якому усі найближчі сусіди цього вузла були б сполучені безпосередньо один з одним) називається називається коефіцієнтом кластерності вузла – $C(i)$.

1.4. Посередництво

Посередництво (*betweenness*) – це параметр вузла, що показує, скільки найкоротших шляхів проходить через нього. Ця характеристика відображає роль даного вузла у встановленні зв'язків у мережі. Вузли з найбільшим посередництвом грають головну роль у встановленні зв'язків між іншими вузлами в мережі. Посередництво b_m вузла m визначається за формулою:

$$b_m = \sum_{i \neq j} \frac{B(i, m, j)}{B(i, j)},$$

де $B(i, j)$ – загальна кількість найкоротших шляхів між вузлами i та j , $B(i, m, j)$ – кількість найкоротших шляхів між вузлами i та j , що проходять через вузол m .

1.5. Модулярність

Модулярність — один з мережевих параметрів, який був введений для вимірювання ступеня розбиття мережі на модулі (кластери, кліки).

Він обчислюється як різниця між часткою ребер всередині кластера в розглянутій мережі і очікуваної часткою ребер всередині кластера в мережі, в якій вершини мають ту ж ступінь, що і у вихідній, але ребра розподілені випадково.

Для розрахунку модулярності використовується поняття матриці суміжності. Матриця суміжності A складається з елементів A_{vw} , значення яких дорівнюють 0, якщо вузол v не сполучений з вузлом w , і вазі зв'язку між v і w , якщо ці вузли сполучені між собою.

Модульність мережі можна виразити формулою:

$$Q = \frac{1}{2m} \sum_{v,w} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w),$$

де A_{vw} – елемент матриці суміжності A , m – кількість ребер у графі, k_v , k_w – ступеня вузлів v і w відповідно, δ – дельта Кронекера (показує, чи знаходяться вузли v і w в одному модулі).

Отже, модульність – це міра якості кластеризації, на основі якої будується широкий клас алгоритмів виявлення груп в мережах.

1.6. Феномени мереж

Про "структуру спільноти" можна говорити тоді, коли існують групи вузлів (кластери), які мають високу щільність ребер між собою, притому, що щільність ребер між окремими групами – низька. Традиційний метод для виявлення структури спільноти – кластерний аналіз. Існують численні методи кластерного аналізу, які базуються на різних вимірах відстаней між вузлами. Для великих соціальних мереж наявність структури спільнот виявилася невід'ємною властивістю.

Значення «слабких зв'язків»

До властивостей реальних соціальних мереж відносяться і, так звані, «слабкі» зв'язки. Аналогом слабких соціальних зв'язків є, наприклад, відносини з далекими знайомими і колегами. У деяких випадках ці зв'язки

виявляються більш ефективними, ніж «сильні» зв'язки. Так, нещодавно було отримано концептуальний висновок в області мобільного зв'язку, який полягає в тому, що «слабкі» соціальні зв'язки між індивідуумами виявляються найбільш важливими для існування соціальної мережі.

Було виявлено, що саме слабкі соціальні зв'язки зв'язують воєдино велику соціальну мережу. Якщо ці зв'язки проігнорувати, то мережа розпадеться на окремі фрагменти, тобто зв'язність мережі порушиться (рис. 1). Виявилося, що саме слабкі зв'язки є тим феноменом, який пов'язує мережу в єдине ціле.

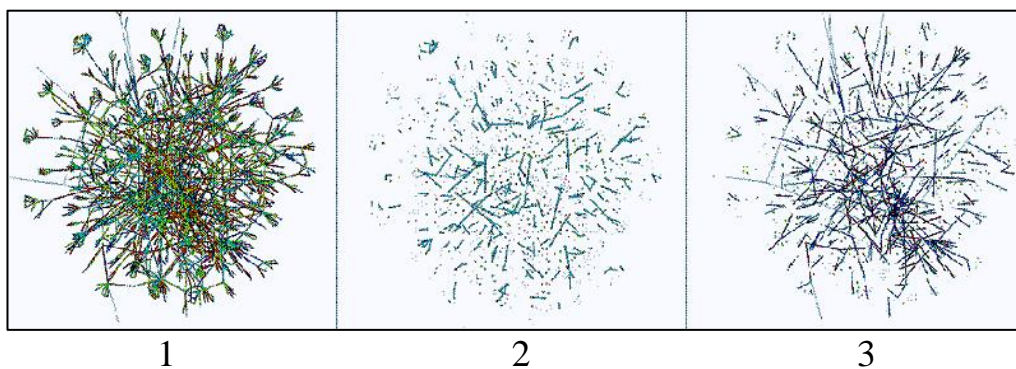


Рис. 1 – Структура мережі:

- 1) повна карта мережі соціальних комунікацій;
- 2) соціальна мережа, з якої вилучено слабкі зв'язки;
- 3) мережа, з якої вилучено сильні зв'язки: структура зберігає зв'язність.

Малий світ (Small World)

Незважаючи на величезні розміри деяких соціальних мереж у багатьох із них існує порівняно короткий шлях між двома будь-якими вузлами – геодезична відстань. У 1967 р. психолог С. Мілграм в результаті виконаних масштабних експериментів обчислив, що існує ланцюжок знайомств, в середньому довжиною шість ланок, практично між двома будь-якими громадянами США.

Д. Уоттс і С. Строгатц виявили феномен, характерний для багатьох реальних мереж, названий ефектом малих світів (*Small Worlds*). При дослідженні цього феномену ними була запропонована процедура побудови

наочної моделі мережі, якій притаманний цей феномен. Три стани цієї мережі представлені на рис. 2: регулярна мережа – кожен вузол якої з'єднаний з чотирма сусідніми, тієї ж самої мережі, у якій деякі «ближні» зв'язки випадковим чином замінені «далекими» (саме в цьому випадку виникає феномен «малих світів») і випадкова мережа, в якій кількість подібних замін перевищила певний поріг.

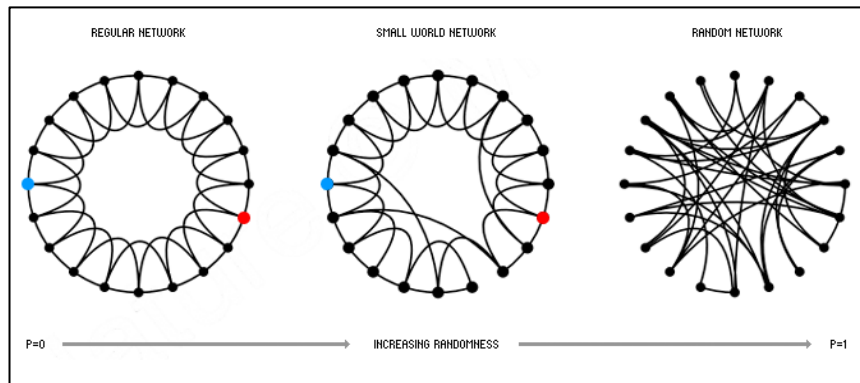


Рис. 2 – Модель Уоттса-Строгатца

У реальності виявилося, що саме ті мережі, вузли яких мають водночас кілька локальних і випадкових «далеких» зв'язків, демонструють одночасно ефект малого світу і високий рівень кластеризації.

На рис. 3 наведені графіки зміни середньої довжини шляху і коефіцієнта кластеризації штучної мережі Д. Уоттса і С. Строгатца від ймовірності встановлення «далеких зв'язків» (у напівлогарифмічній шкалі).

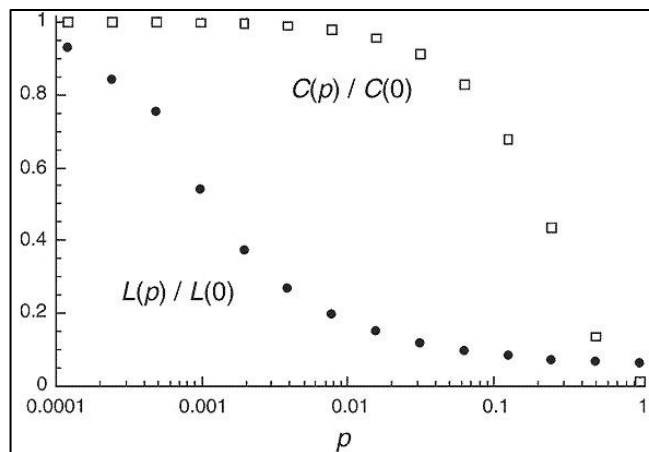


Рис. 3 – Динаміка зміни довжини шляху і коефіцієнта кластерності в моделі Уоттса-Строгатца

Феномен клубу багатих

WWW є мережею, для якої також підтверджений феномен малих світів. Аналіз топології веб, проведений Ши Жоу (*S. Zhou*) і Р. Дж. Мондрагоном (*R.J. Mondragon*) з Лондонського університету показав, що вузли з великим ступенем вихідних гіперпосилань мають більше зв'язків між собою, ніж з вузлами з малим ступенем, тоді як останні мають більше зв'язків з вузлами з великим ступенем, ніж між собою. Цей феномен був названий "клубом багатих" (*Rich-Club Phenomenon*). Дослідження показало, що 27% всіх з'єднань мають місце між всього 5% найбільших вузлів, 60% припадає на з'єднання інших 95% вузлів з 5% найбільших і тільки 13% – це з'єднання між вузлами, які не входять до провідних 5%.

Ці дослідження дають підстави вважати, що залежність WWW від великих вузлів є значно істотнішою, ніж передбачалося раніше, тобто вона ще чутливіша до зловмисних атак. З концепцією "малих світів" пов'язаний також практичний підхід, що називається "мережевою мобілізацією", яка реалізується над структурою "малих світів". Зокрема, швидкість поширення інформації завдяки ефекту "малих світів" в реальних мережах зростає на порядки в порівнянні з випадковими мережами, адже більшість пар вузлів реальних мереж з'єднані короткими шляхами.

1.7. Модель випадкової мережі Ердеша-Реньї

Існує дві моделі класичного випадкового графа: в першій вважається, що M ребер розподілені довільно і незалежно між парами з N вершин графа; в другій моделі фіксується ймовірність m , з якою може об'єднуватися кожна з пар вершин. При $m \rightarrow \infty$ і $N \rightarrow \infty$ для обох варіантів розподіл ступенів вузлів k визначається формулою Пуассона:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!},$$

де середнє значення ступеня вузла: $\langle k \rangle = 2M / N$ для першої моделі і $\langle k \rangle = mN$ для другої. При цьому середня довжина найкоротшого шляху для мережі Ердеша-Реньї становить $\langle l \rangle = \ln(N) / \ln(\langle k \rangle)$, а коефіцієнт кластерності: $C \sim \langle k \rangle / N$.

1.8. Модель випадкової мережі Барабаші-Альберт

Сценарій побудови мережі Барабаші-Альберт базується на двох механізмах – зростанні і переважному приєднанні (*preferential attachment*). Модель використовує алгоритм: зростання мережі відбувається починаючи з невеликої кількості вузлів n_0 , до яких на кожному часовому кроці додається новий вузол з $n < n_0$ зв'язками, які приєднуються до вже існуючих вузлів; переважне приєднання полягає в тому, що ймовірність приєднання $P(k_i)$ нового вузла до вже існуючого вузла i залежить від ступеня k_i вузла i :

$$P(k_i) = \frac{k_i}{\sum_j k_j}.$$

У знаменнику підсумовування ведеться по всіх вузлах. Як комп'ютерні моделі, так і аналітичні рішення дають ступеневу асимптотику розподілу ступенів вузлів з показником γ , близьким за значенням до числа 3.

Одним з напрямків аналізу соціальних мереж є візуалізація, яка має важливе значення, оскільки часто дозволяє робити важливі висновки щодо характеру взаємодії суб'єктів-вузлів, не прибігаючи до точних методів аналізу. При відображенні моделі мережі може виявитися доцільним:

- розміщення вузлів мережі в двох вимірах;
- просторове впорядкування об'єктів в одному вимірі відповідно до деяких кількісних властивостей;
- використання загальних для всіх мережевих діаграм методів для відображення кількісних і якісних властивостей об'єктів і відносин.

1.9. Ранжування вузлів мереж

Два найвідоміших алгоритми ранжування вузлів мереж *HITS* (*Hyperlink Induced Topic Search*) и *PageRank* були розроблені в 1996 році в *IBM* Дж. Клейнбергом (*J. M. Kleinberg*) і в Стенфордському університеті – С. Бріном (*S. Brin*) і Л. Пейджем (*L. E. Page*), відповідно.

Алгоритм *HITS*

Алгоритм *HITS* забезпечує вибір з мережі кращих «авторів» (вузлів, на які ведуть посилання) і «посередників» (вузлів, від яких йдуть посилання цитування).

Вузол є хорошим посередником, якщо він містить посилання на цінні джерела, і навпаки, вузол є хорошим автором, якщо він згадується хорошими посередниками.



Дж. Клейнберг, автор алгоритму HITS

Для кожного вузла $d_j \in D$ рекурсивно обчислюється його значимість як автора $a(d_j)$ і посередника $h(d_j)$ за формулами:

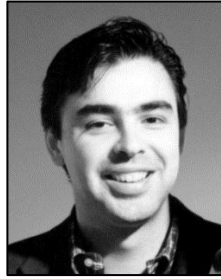
$$a(d_j) = \sum_{i \rightarrow j} h(d_i), \quad h(d_j) = \sum_{j \rightarrow i} a(d_i).$$

При цьому, в першій формулі в сумі беруть участь значення посередництва лише тих вузлів, які посилаються на вузол з індексом j .

Відповідно, в другій формулі в сумі беруть участь значення авторства лише тих вузлів, на які посилається вузол з індексом j . На кожному кроці алгоритму значення $a(d_j)$ і $h(d_j)$ нормуються.

Алгоритм *PageRank*

Алгоритм *PageRank* застосовується для довільного вузла з урахуванням кількості посилань від інших вузлів на даний вузол. При цьому *PageRank* як і *HITS*, на відміну від літературного індексу цитування, не вважає всі посилання рівнозначними.



Л. Пейдж і С. Брін, автори PageRank

Принцип розрахунку рангу вузлів *PageRank* ґрунтується на моделі «випадкового блукання» користувача за наступним алгоритмом: він відкриває випадковий вузол (веб-сторінку), з якої переходить за випадково обраним посиланням. Потім він переміщається на іншу веб-сторінку і знову активізує випадкове посилання і так далі, постійно переходячи від сторінки до сторінки, ніколи не повертаючись. Іноді, коли з деякою ймовірністю $1 - \delta$ йому таке блукання набридає, або на сторінці немає посилань на інші сторінки, він знову переходить на випадкову веб-сторінку – не за посиланням, а набравши вручну деякий *URL*. Передбачається, що ймовірність того, що блукаючий в мережі користувач перейде на деяку певну веб сторінку – це її ранг. Вочевидь, *PageRank* вузла тим вище, чим більше інших вузлів посилається на неї, і чим ці вузли є популярнішими.

Нехай є n вузлів $\{d_1, d_2, \dots, d_n\}$, які посилаються на даний вузол (веб-сторінку A), а $C(A)$ – загальне число посилань з вузла A на інші вузли. Визначається деяке фіксоване значення δ , як ймовірність того, що користувач, переглядаючи яку-небудь веб-сторінку з множини D , перейде на вузол A за посиланням, а не набираючи її *URL* в явному вигляді.

У рамках моделі вірогідність продовження цим користувачем веб-серфінгу по мережі з N веб-сторінок без використання посилань, шляхом ручного введення адреси (URL) з випадкової сторінки складе $1-\delta$ (альтернатива переходу за посиланнями). Індекс *PageRank* $PR(A)$ для вузла A розглядається як ймовірність того, що користувач опиниться в деякий випадковий момент часу на цьому вузлі:

$$PR(A) = (1 - \delta) / N + \delta \sum_{i=1}^n \frac{PR(d_i)}{C(d_i)}.$$

За цією формулою індекс вузла розраховується простим ітераційним алгоритмом.

Незважаючи на відмінності *HITS* і *PageRank*, в цих алгоритмах загальне те, що авторитетність (вага) вузла залежить від ваги інших вузлів, а рівень "посередника" залежить від того, наскільки є авторитетними вузли, на які він посилається.

Питання для самоперевірки

1. Назвіть типові характеристики, що застосовуються у прикладних дослідженнях для мережевого аналізу.
2. Які параметри виділяють при аналізі складних мереж: для окремих вузлів, мережі у цілому?
3. Як визначається функція розподілу ступенів вузлів мережі?
4. Дайте означення безмасштабованої мережі.
5. Дайте означення найкоротшого та середнього шляху мережі.
6. Що таке «глобальна ефективність мережі»?
7. Як визначається уразливість мережі вузла?
8. Розкрийте суть поняття «коефіцієнт кластерності вузла». Як він визначається?
9. Поясніть параметр вузла – «посередництво». Як він визначається?
10. Що таке модульність мережі та як вона розраховується?
11. Поясніть суть феноменів «малий світ» та «клуб богатих».
12. У чому суть моделі Ердеша-Реньї?
13. Розкрийте суть моделі Барабаші-Альберт.
14. Поясніть суть алгоритмів *HITS* та *PageRank*.

2. ІНСТРУМЕНТИ ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗУ МЕРЕЖ

Перш ніж перейти до розгляду мети даного навчального посібника – системі *Gephi* – розглянемо, які є ще вільно поширювані, безкоштовні програми, інструменти візуалізації та аналізу складних мереж доступні сьогодні користувачам-аналітикам.

У теперішній час існує досить багато таких програм, які забезпечують візуалізацію найпростіших мережових структур – невеликих графів. Серед таких програм можна назвати, наприклад, систему *uDraw (Graph)*, розроблену в Німеччині в Бременському університеті (рис. 4).

2.1. *uDraw (Graph)*

uDraw(Graph) – спеціальна програма, що забезпечує візуалізацію різноманітних графів (<http://www.informatik.uni-bremen.de/uDrawGraph/en/>).

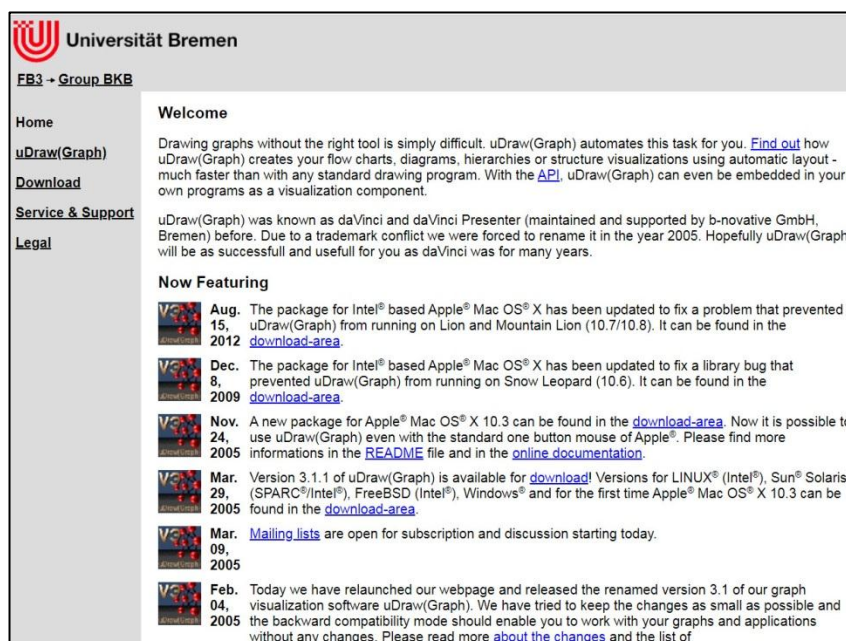


Рис. 4 – Стартова сторінка програми *uDraw (Graph)*

На рис. 5 наведено граф, створений в інтерфейсі програми *uDraw (Graph)*.

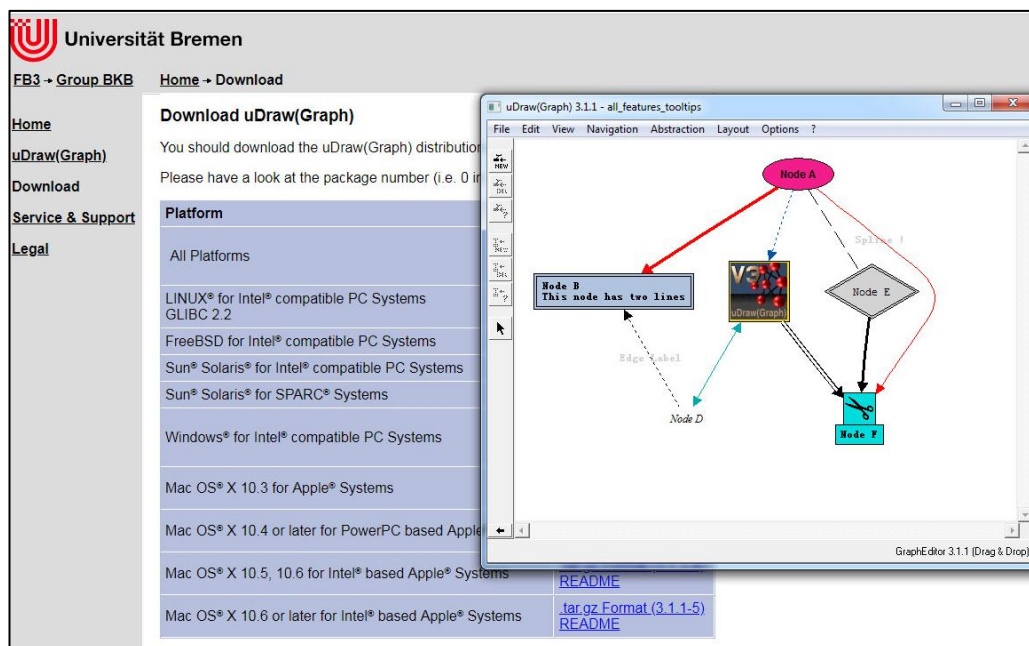


Рис. 5 – Приклад графа, створеного за допомогою *uDraw (Graph)*

На рис. 6 приведена панель для забезпечення навігації, за допомогою якої забезпечується «переміщення» по вузлах графу. При побудові графа є можливість вибору фігур, якими зображуються вузли, змінювати кольори фону, вузлів і ребер. Побудований граф можна зберігати у вигляді зображень, представлених в форматах *GIF*, *TIF*, *JPEG*, *PNG* (режим «Експорт», рис. 7).

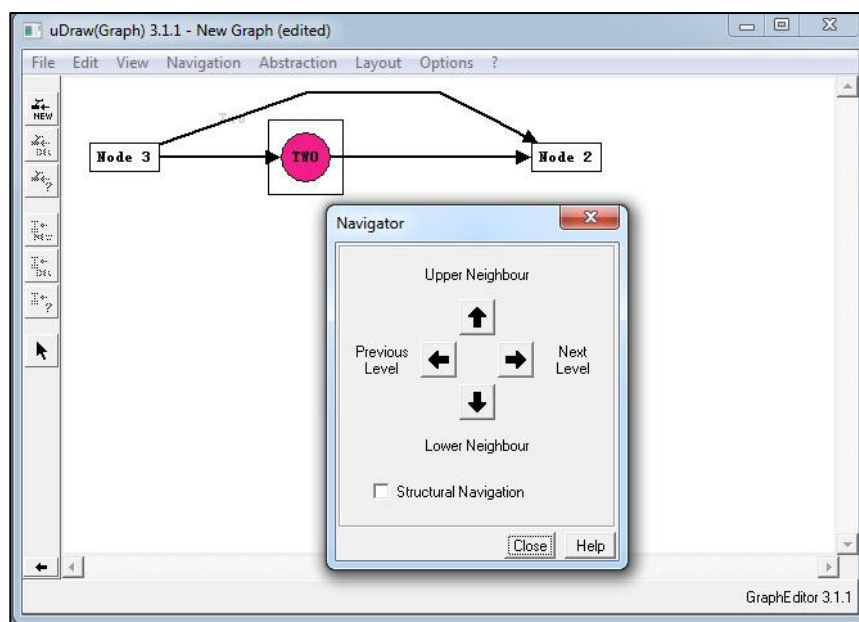


Рис. 6 – Панель навігації

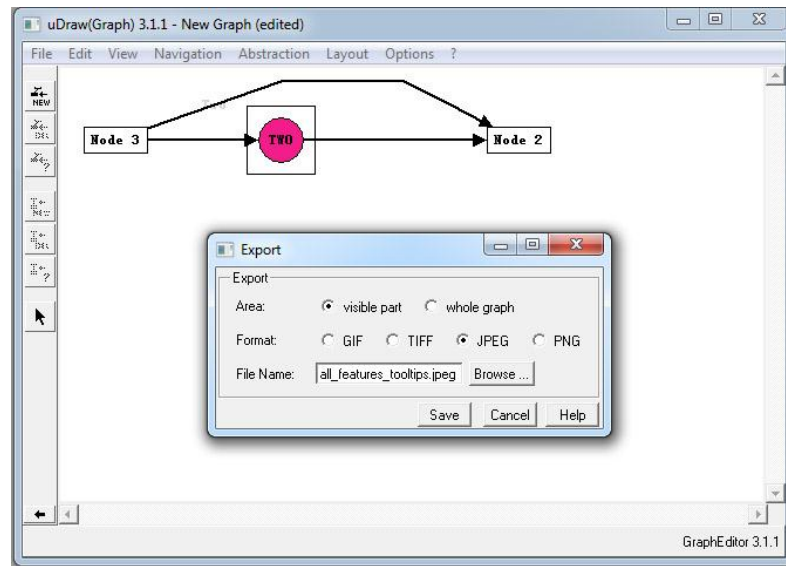


Рис. 7 – Панель експорту графа, створеного за допомогою uDraw (Graph)

2.2. Graph Online

Для практичної роботи з графовими структурами користувачу не обов'язково потрібно встановлювати програмне забезпечення на своєму комп'ютері. Існує безліч мережевих сервісів, серед яких можна назвати сервіс *Graph Online*.

Graph Online (<https://graphonline.ru/en/>) – простий і функціональний інструмент для створення, дослідження і візуалізації невеликих графів в режимі онлайн. Забезпечує експорт файлів в формати *GraphML*, *CSV*, вивід інформації у вигляді матриць, а також визначення різних параметрів графу, а також реалізацію таких алгоритмів, як пошук найкоротшого шляху, компонент зв'язності, діаметру графу і т.п.

У режимі онлайн за допомогою простих маніпуляцій користувач може сформувати вузли і з'єднати їх ребрами (неспрямованими, спрямованими, зваженими, див. рис. 8).

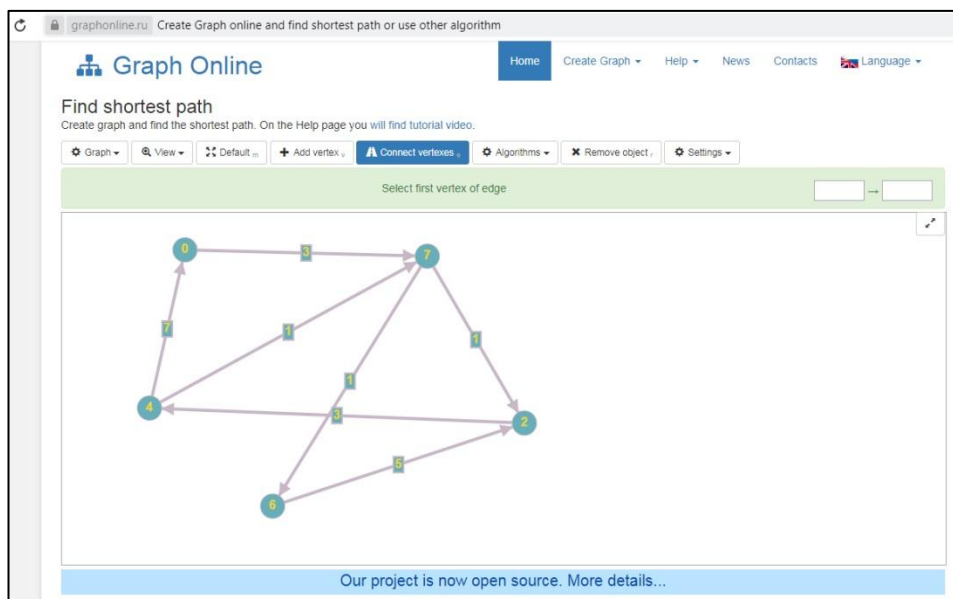


Рис 8 – Приклад графа, створеного за допомогою Graph Online

Graph Online дозволяє відобразити в текстовому вигляді (практично в форматі CSV) матрицю суміжності, редагувати її, змінюючи вже створений файл (рис. 9).

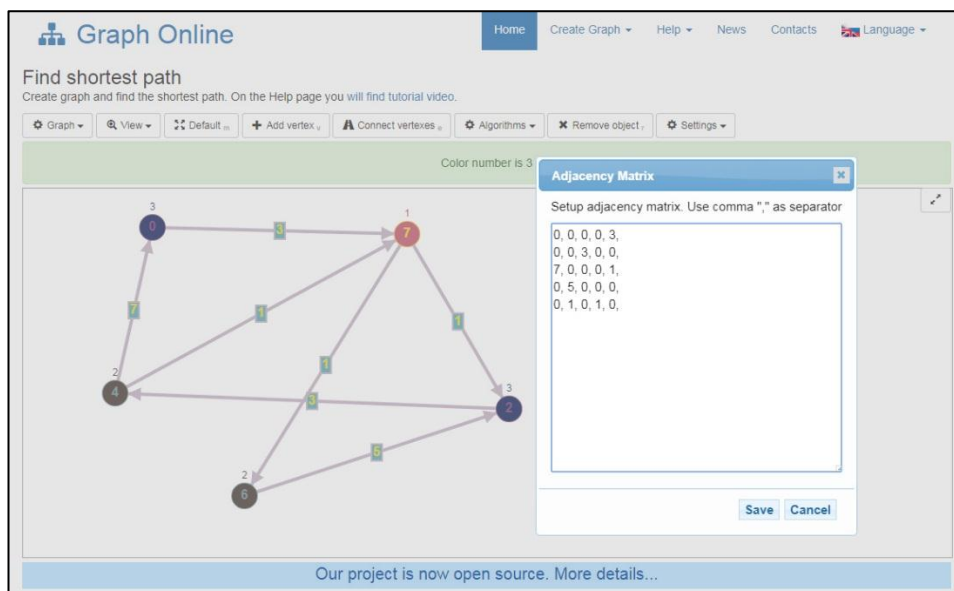


Рис. 9 – Матриця суміжності графа, створеного за допомогою *Graph Online*

Сервіс *Graph Online* дозволяє знаходити найкоротші шляхи за алгоритмами Дейкстри, Флойда-Уоршелла, Ейлерові шляхи і цикли, визначати більшість параметрів побудованого графа (рис. 10).

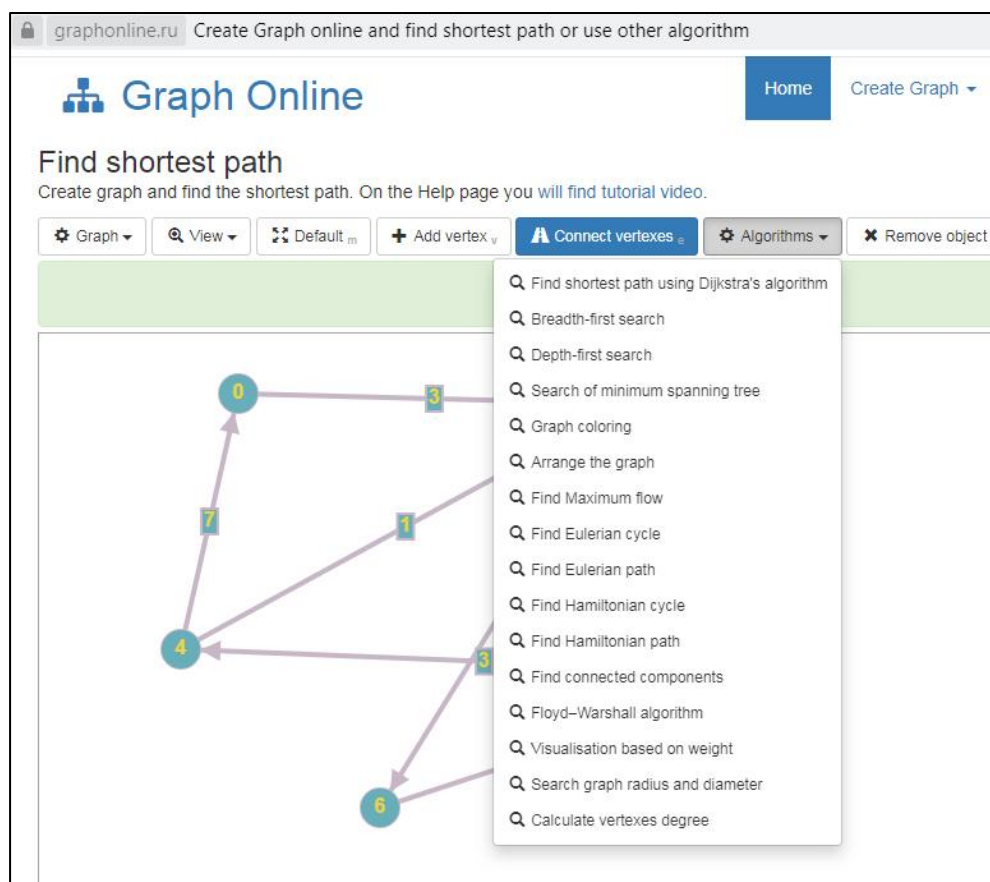


Рис. 10 – Панель розрахунку параметрів графа за допомогою Graph Online

2.3. Graphviz

Зовсім інший рівень сервісу візуалізації і аналізу мережевих структур надають системи, створені великими колективами розробників, зокрема, система *Graphviz* (*Graph Visualization Software*). Система *Graphviz* розроблена лабораторією *AT&T* і надана у вільний доступ з відкритими вихідними файлами за ліцензією *EPL* (*Eclipse Public License*). *Graphviz* працює в середовищі таких операційних систем, як *Linux*, *Mac OS*, *Microsoft Windows* (рис. 11).

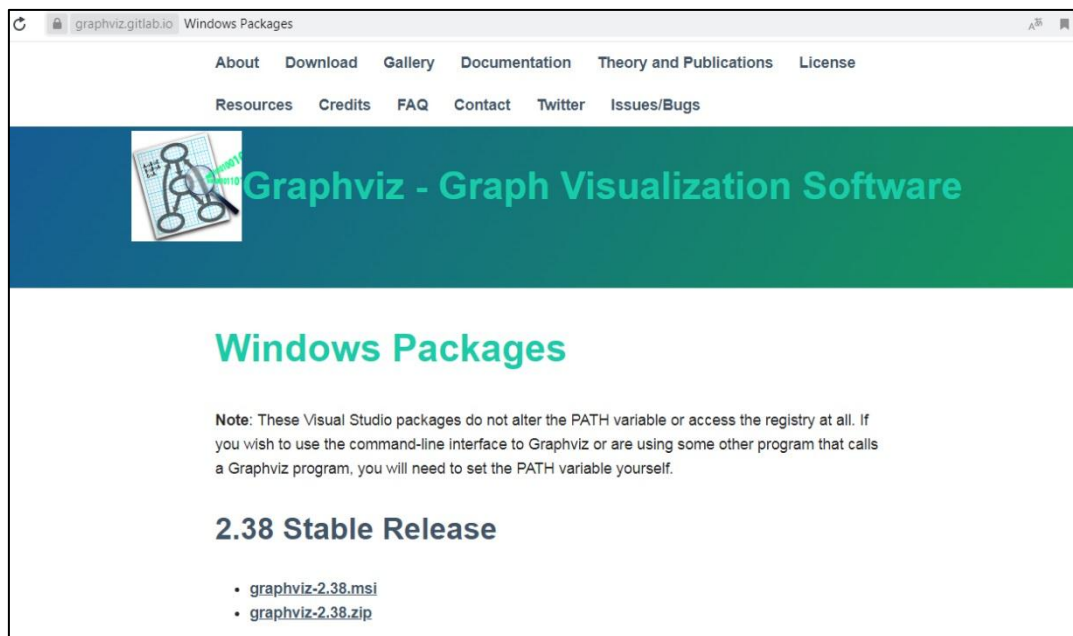


Рис. 11 – Стартова сторінка системи *Graphviz*

Система *Graphviz* (<http://graphviz.org/>) являє собою пакет утиліт з візуалізації графів, заданих у вигляді опису мовою *DOT*, а також додаткових текстових і графічних програм, віджетів і бібліотек, які використовуються при розробці програмного забезпечення для візуалізації структурованих даних.

Graphviz складається з набору утиліт командного рядка і програм з графічним інтерфейсом, здатних обробляти файли мовою *DOT*, а також віджетів і бібліотек, що полегшують створення графів і програм для побудови графів. Вона включає наступні:

- *dot* — інструмент створення багаторівневого графу з можливістю виведення зображення результуючого графу у множині форматів (*PNG*, *PDF*, *PostScript*, *SVG* і низки інших);
- *neato* — інструмент для створення графу на основі «пружинної» моделі («*spring model*», «*energy minimised*»);
- *twopi* — інструмент для створення графу на основі «радіальної» моделі;
- *circo* — інструмент для створення графау на основі «кругової» моделі;
- *fdp* — інструмент створення неспрямованого графу на основі *fdp*-моделі;

- *dotty* — графічний інтерфейс для створення графів;
- *lefty* — програмований графічний віджет (мовою *EZ*).

До пакету утиліт входить програмний модуль «*dot*» — автоматичний візуалізатор орієнтованих графів, який приймає на вхід текстовий файл мовою *DOT* з поданням графа у вигляді суміжних списків, а на виході формує граф у вигляді графічного, векторного або текстового файлу.

Для побудови графа в системі *Graphviz* достатньо в режимі *Edit* задати його опис спеціальною мовою опису *DOT*, а потім в режимі *Graph* вибрати вкладку *Layout*, щоб візуалізувати граф (рис. 12).

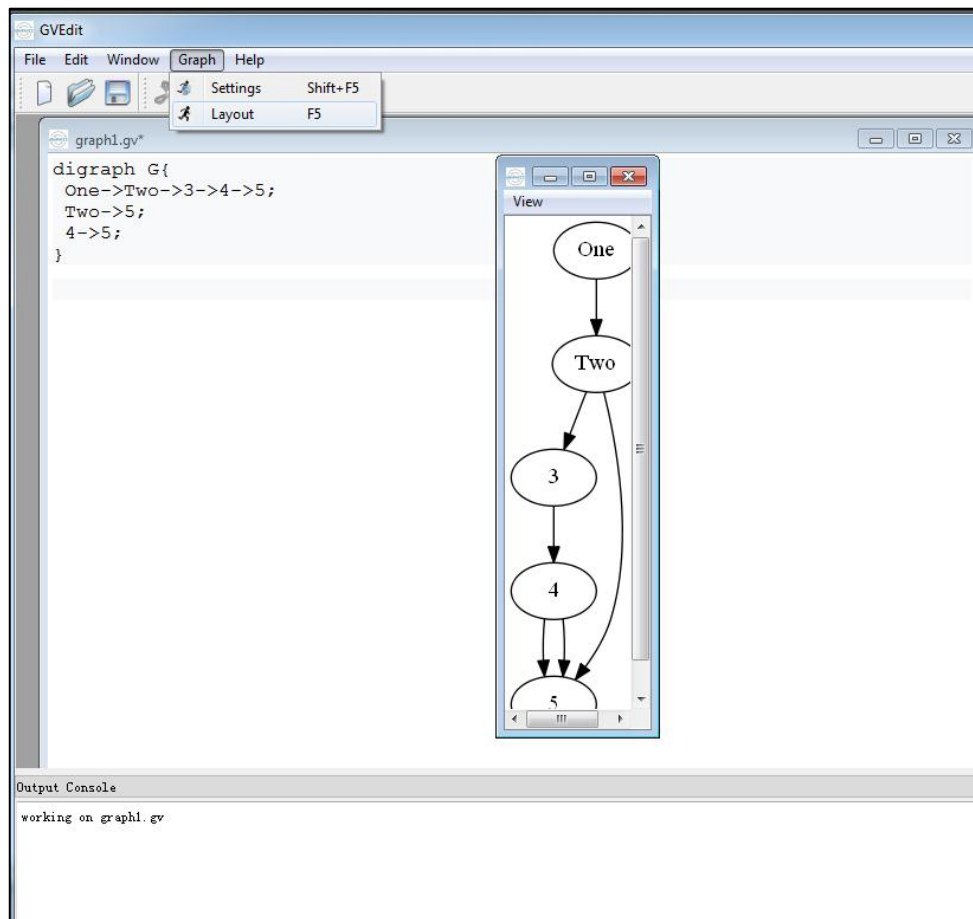


Рис. 12 – Опис графа мовою *DOT*, візуалізація і вікно діагностики

У режимі *Graph* → *Setting* можна змінювати параметри графу вручну без необхідності безпосереднього опису мовою *DOT*, наприклад, змінювати форму вузлів, колір фону, колір вузлів і т.п. (див. рис. 13).

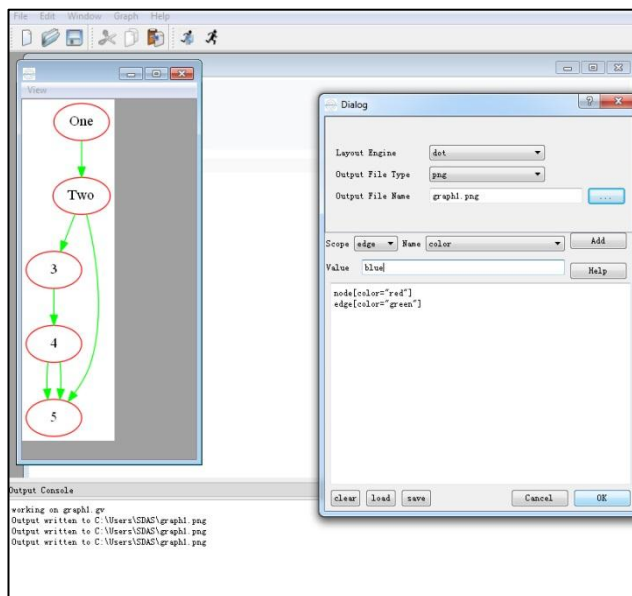


Рис. 13 – Режим *Graph* → *Setting*

Структура *DOT*-файлу дуже проста. На рис. 14 наведено відповідний приклад. Програма «dot» сама розпізнає всі зв'язки графу і впорядковує його таким чином, щоб була найменша кількість перетинів.

Структура графа на мові *DOT* описується у вигляді списку субграфів, кожен з яких має вигляд:

```
graph %имя_графа% {  
}
```

У цьому випадку в фігурних дужках { } знаходяться коментарі та інструкції, що описують окремий субграф. Інструкції описують вузли і ребра цільового графу і розділяються крапкою з комою.

Мова *DOT* підтримує коментарі в стилі мов *C* і *C++* — // і /**/, а також символ # в якості першого символу однорядкового коментаря.

Припускається подання неорієнтованих або орієнтованих графів.

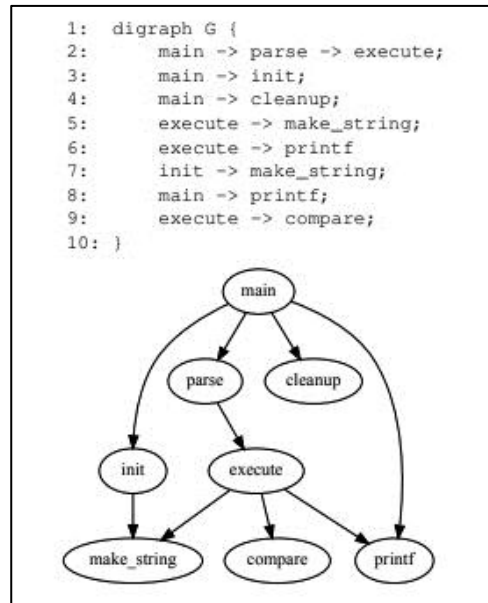


Рис. 14 – Опис мовою DOT і відповідний граф

Неорієнтований граф мовою DOT описується списком вузлів і ребер, представлених назвою вузлів і подвійним тире (--) між пов'язаними вузлами, наприклад:

```

graph graphname {
    a;
    b;
    c;
    d;
    a -- b;
    b -- c;
    b -- d;
}

```

Припустимо скорочений опис:

```

graph graphname {
    a -- b -- c;
    b -- d;
}

```

Орієнтований граф мовою *DOT* описується списком вузлів і ребер, представлених назвою вузлів і стрілкою (->) між пов'язаними вузлами, наприклад:

```
digraph graphname {  
    a -> b -> c;  
    b -> d;  
}
```

Припустимий і надлишковий опис:

```
digraph graphname {  
    a;  
    b;  
    c;  
    d;  
    a -> b;  
    b -> c;  
    b -> d;  
}
```

При описі графів мовою *DOT* можна використовувати атрибути, що визначають колір, форму і стиль вузлів і ребер. Атрибути описуються парами ключ=значення, укладених в квадратні дужки ([ключ= значення]).

Для кожного елемента графу може бути визначено декілька атрибутів, відокремлених пробілом.

```
graph graphname {  
    // label - видима назва вершини  
    a [label="Foo"];  
    // shape - визначення форми вершини  
    b [shape=box];  
    // color - визначення кольору ребра  
    a -- b -- c [color=blue];  
    // style - визначення стилю ребра  
    b -- d [style=dotted];  
}
```

Інтерпретатори мови *DOT* при візуалізації розташовують елементи автоматично. Для коригування візуального представлення використовуються графічні редактори, серед яких і представлений в даному пункті *Graphviz*.

2.4. Social Network Visualizer

Мабуть, найкращим настільним користувацьким додатком для візуалізації та аналізу мережових структур (можливо, тільки за винятком *Gephi*) у даний час є *Social Networks Visualizer (SocNetV)*. Цей безкоштовний кросплатформний програмний додаток для аналізу і візуалізації соціальних мереж, розроблений мовою *C++*. *SocNetV* є вільним програмним забезпеченням, ліцензованим під *GNU General Public License 3 (GPL3)*.

Розробник системи – Дімітріс Каламарас (*Social Network Analysis and Visualization Software*). Вихідний програмний код програми, пакети і виконувані файли для *Windows*, *Linux* і *MacOS* доступні на веб-сайті проекту: <http://socnetv.org> (рис. 15).



Рис. 15 – Стартова сторінка системи SocNetV

У *SocNetV* є можливості ручного введення мережі (графу) або завантаження вже існуючого мережевого набору даних, представленого в форматах *GraphML*, *UCINET*, *Pajek* і т. д., обчислення статистики, значень центральності, застосування різних алгоритмів компонентування, заснованих на центральності або посередництві (*Betweenness*) вузлів або динамічних моделей.

Базові можливості *SocNetV*:

- створення випадкових мереж з використанням різних моделей мереж нового покоління (*Barabási–Albert Scale-Free*, *Erdős–Rényi*, *Watts-Strogatz Small-World*, *d-regular* та ін.);
- імпорт з мережових форматів (*GraphML*, *Adjacency*, *Pajek*, *UCINET*, списки і т. д.);
- завантаження і редагування мережі;
- швидкий розрахунок індексів центральності, зв'язності, щільності, геодезичної відстані, ступенів вузлів, ексцентриситету, коефіцієнту кластерності та ін.;
- розрахунок додаткових метрик для аналізу соціальних мереж, таких як індекси центральності і значущості;
- розрахунок *PageRank*;
- матричні обчислення: визначення графіку суміжності, матриці Лапласа, цитування та ін.;
- аналіз структурної еквівалентності з використанням ієрархічної кластеризації, подібності профілів зв'язків, коефіцієнтів Пірсона;
- сканування сайтів для автоматичного створення "соціальних мереж" з посилань, знайдених по заданих адресах;
- наявність швидких алгоритмів для виявлення спільнот, таких як визначення тріад, клік та ін.;
- наявність різних моделей компоновання (укладання) вузлів, заснованих на індексах положення (кругові, рівневі та вузлові) за шкалою центральності, або на силовому розміщенні (алгоритми *Eades Spring Embedder*, *Fruchterman-Reingold* і т. д.) для візуалізації соціальних мереж (див. рис. 16);
- можливість експорту у формати *GraphML*, *Pajek*, *Adjacency*.

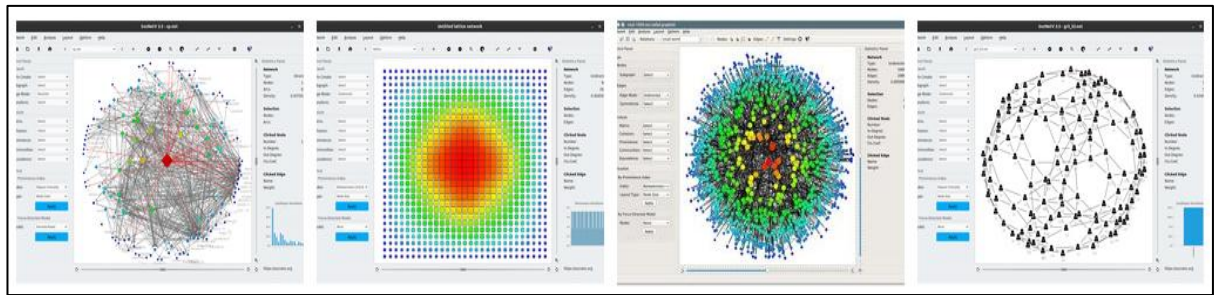


Рис. 16 – Варіанти компоновки мережі в системі SocNetV

На Рис. 17 представлений приклад завантаження в систему SocNetV файлу в форматі CSV з роздільниками комами (режим *Network* → *Import* → *Adjacency Matrix*). Після завантаження відображається невпорядкований граф (рис. 18), який за допомогою радіального компонування (рис. 19), розфарбування та завдання алгоритму розрахунку центральності може бути представлений в наочному вигляді (рис. 20).

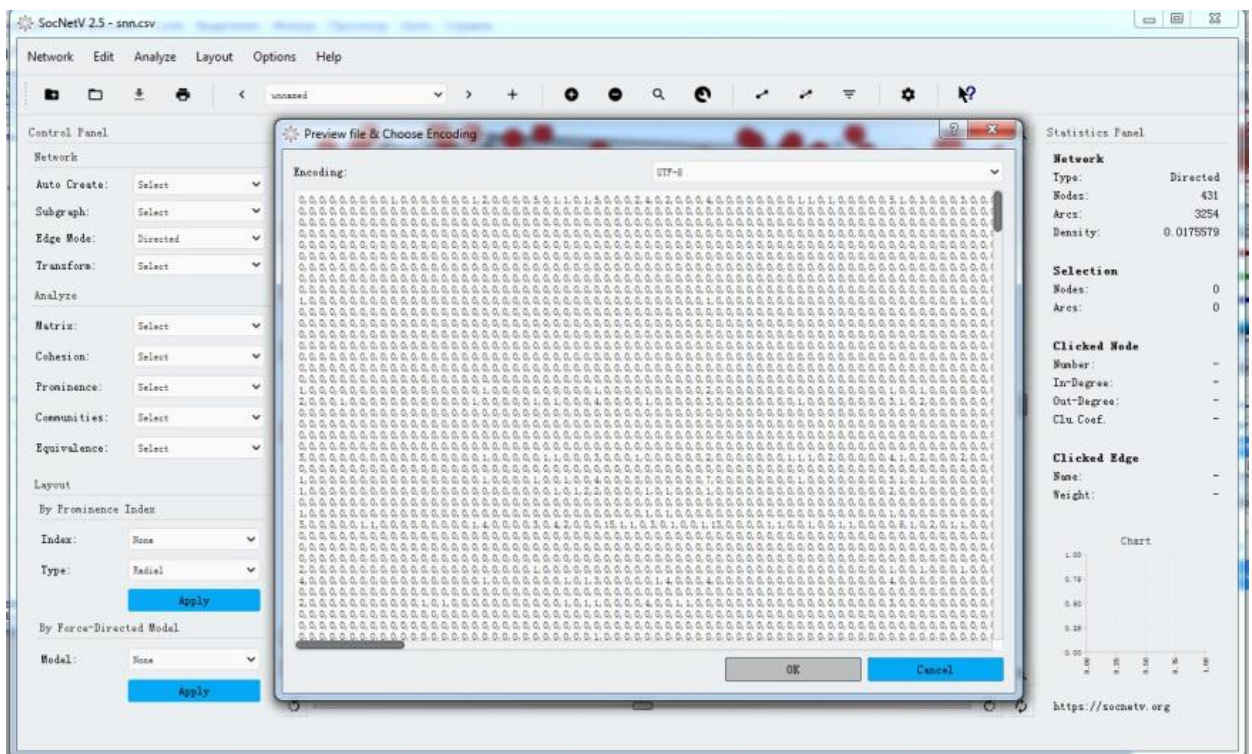


Рис.17 – Завантаження в систему SocNetV файлу в форматі CSV

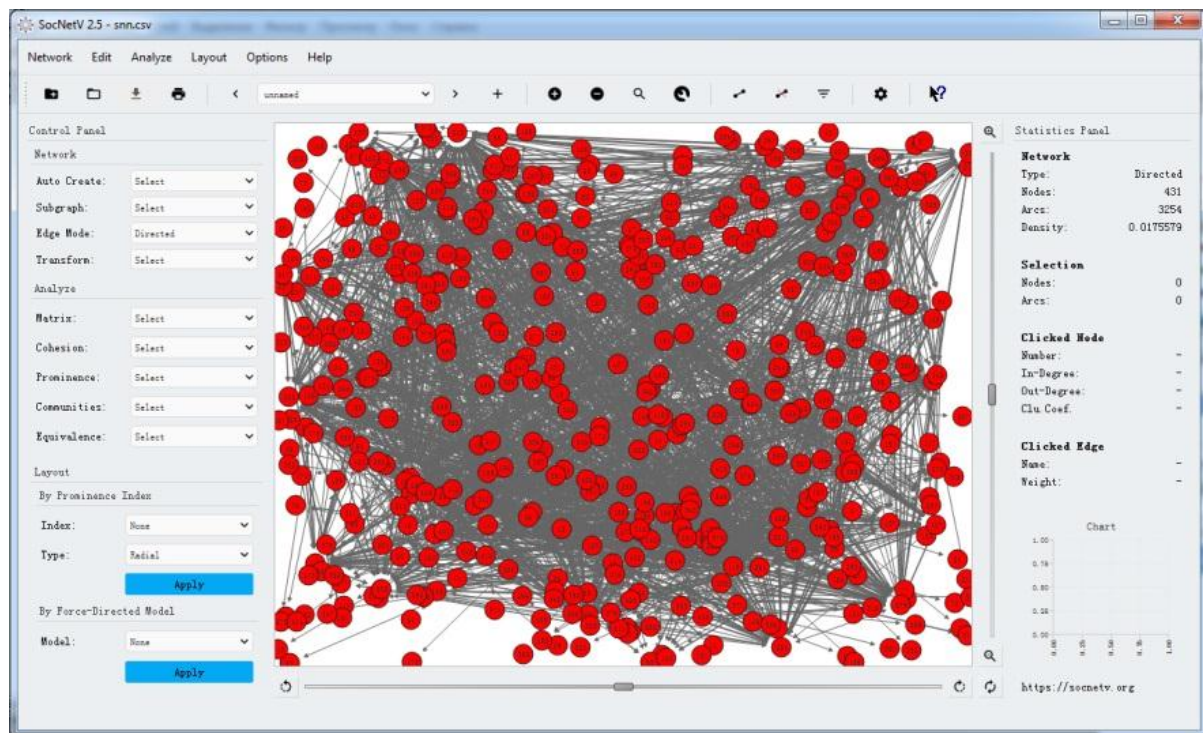


Рис.18 – Відображення невпорядкованого графа

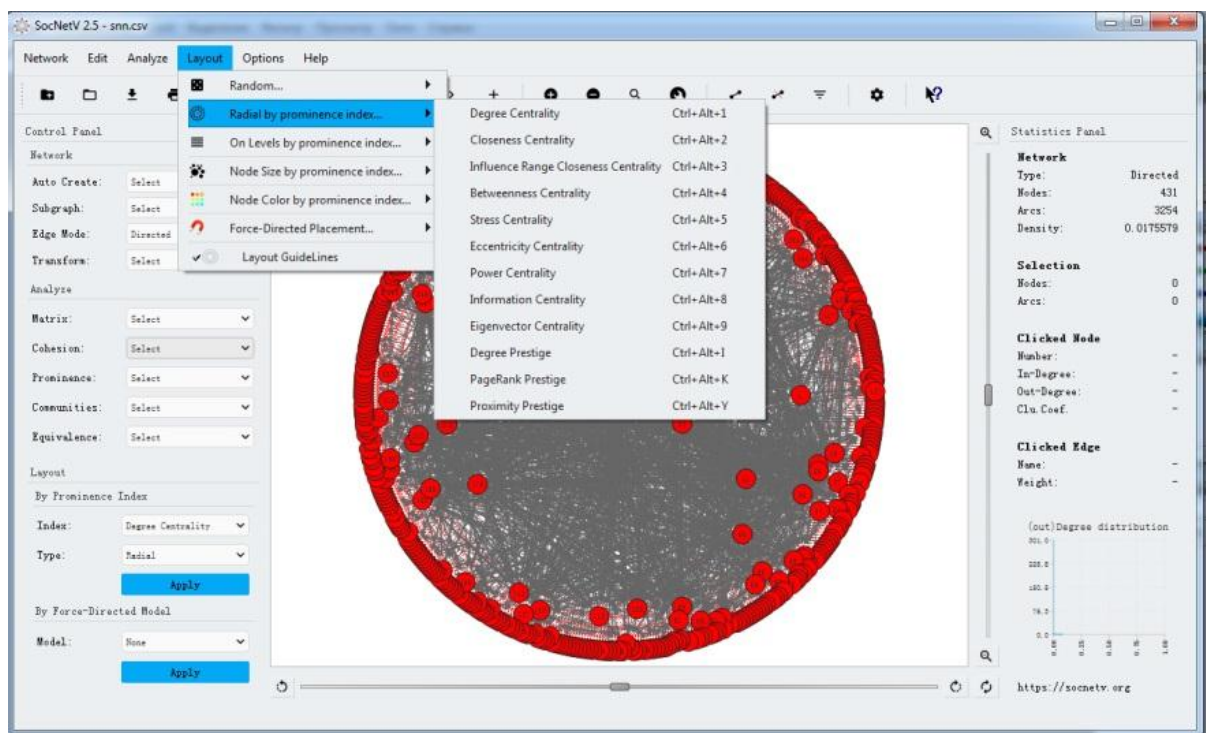


Рис. 19 – Вибір радіальної комоновки і виду центральності

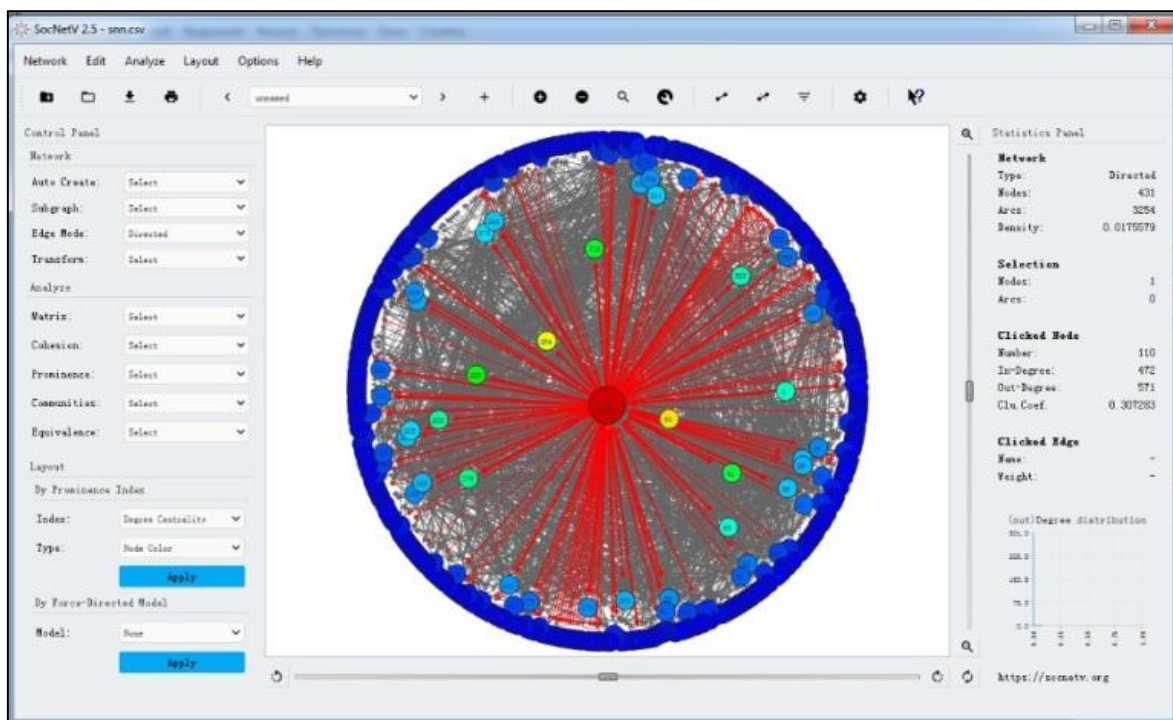


Рис. 20 – Відображення графу в радіальному компонуванні

На рис. 21 і 22 наведено відображення того ж графу в компонуванні *Node* (вузловий) і *Force-Directed*.

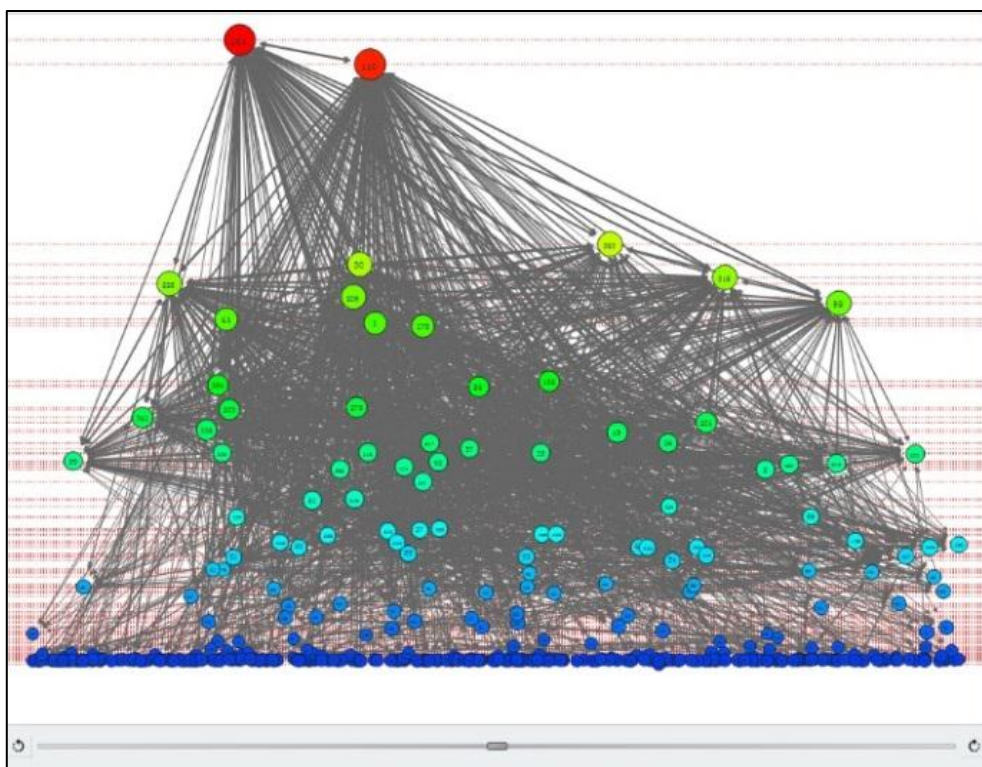


Рис. 21 – Відображення графу у вузловому компонуванні

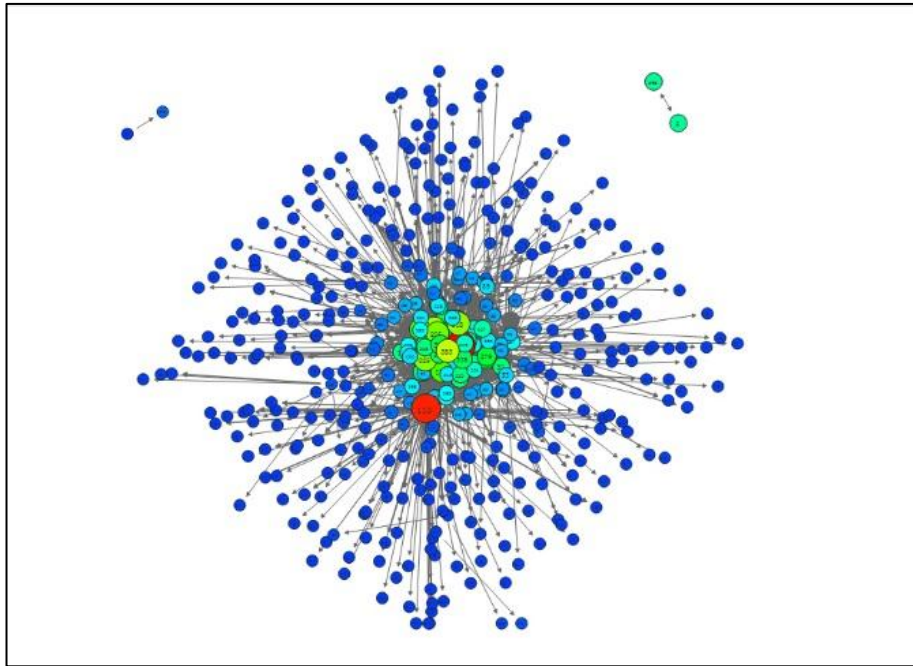


Рис. 22 – Відображення графу в компонуванні *Force-Directed*

2.5. Формат GraphML

Чимало програм візуалізації і аналізу графів (*Graph Online*, *SocNetV*, *Gephi*) використовують мову представлення даних як універсальну мову, що задовольняє специфікаціям *XML*.

Зокрема, в *SocNetV* і в *Gephi* є можливості завантаження вже існуючого, і, відповідно, вивантаження набору даних, представленого в форматі *GraphML*.

Призначення файлу в форматі *GraphML* (*GraphML*-документа) – визначення графу. Для початку розглянемо простий граф показаний на наведеному нижче рис. 23. Цей граф містить 11 вузлів і 12 ребер.

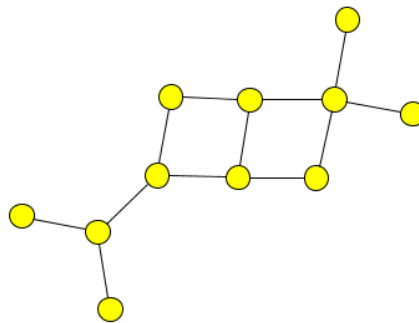


Рис. 23 – Простий граф

Мовою *GraphML* цей простий граф описується в такий спосіб:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
    http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="undirected">
    <node id="n0"/>
    <node id="n1"/>
    <node id="n2"/>
    <node id="n3"/>
    <node id="n4"/>
    <node id="n5"/>
    <node id="n6"/>
    <node id="n7"/>
    <node id="n8"/>
    <node id="n9"/>
    <node id="n10"/>
    <edge source="n0" target="n2"/>
    <edge source="n1" target="n2"/>
    <edge source="n2" target="n3"/>
    <edge source="n3" target="n5"/>
    <edge source="n3" target="n4"/>
    <edge source="n4" target="n6"/>
    <edge source="n6" target="n5"/>
    <edge source="n5" target="n7"/>
    <edge source="n6" target="n8"/>
    <edge source="n8" target="n7"/>
    <edge source="n8" target="n9"/>
    <edge source="n8" target="n10"/>
  </graph>
</graphml>
```

GraphML-документ складається із заголовку, елемента *graphml* і ряду піделементів: *graph*, *node*, *edge*. Наприкінці розділу розглянемо перераховані елементи детальніше, і покажемо, як вони визначають граф.

Заголовок з посиланням на *XML*-схему має вигляд:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
    http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  ...
</graphml>
```

Перший рядок документу – визначає те, що документ є підмножиною стандарту *XML* 1.0 і виконаний у кодуванні *UTF-8*.

Другий рядок містить кореневий елемент *GraphML*-документу: *graphml*. Елемент *graphml*, також як і всі інші елементи мови *GraphML*, належить іменному простору <http://graphml.graphdrawing.org/xmlns>. Наступні два *XML*-атрибути визначають стандартну *XML*-схему даного документу, розташовану на сервері graphdrawing.org. Перший атрибут, *xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance"* – визначає, *xsi* як префікс іменного простору *XML*-схеми. Другий атрибут, *xsi: schemaLocation = "http://graphml.graphdrawing.org/xmlns"* визначає місцезнаходження *XML*-схеми для елементів іменного простору *GraphML*.

Граф описується за допомогою елемента *graph*. Елементи, які розташовані всередині елемента *graph*, забезпечують оголошення вузлів і ребер. Вузол оголошується за допомогою елемента *node*, а ребро за допомогою елемента *edge*.

Визначення графу:

```
<graph id="G" edgedefault="directed">
  <node id="n0"/>
  <node id="n1"/>
  ...
  <node id="n10"/>
  <edge source="n0" target="n2"/>
  <edge source="n1" target="n2"/>
  ...
  <edge source="n8" target="n10"/>
</graph>
```

У *GraphML* не встановлено порядок проходження елементів *node* і *edge* в описі.

Граф у *GraphML* може містити спрямовані і неспрямовані ребра одночасно. Спрямованість ребер, що привласнюється за замовчуванням, задається *XML*-атрибутом *edgedefault* елемента *graph*. Цей *XML*-атрибут може приймати одне з двох значень: *directed* і *undirected*.

Додатково, за допомогою атрибута *id*, графу може бути присвоєно ідентифікатор, коли на даний граф потрібно організувати посилання.

Вузол у графі оголошується за допомогою елемента *node*. Кожен вузол має унікальний (в межах даного документа) ідентифікатор, який задається за допомогою *XML*-атрибута *id*.

Ребро у графі оголошується за допомогою елемента *edge*. Кожне ребро має дві кінцеві точки, що задаються за допомогою *XML*-атрибутів *source* і *target*. Значення атрибутів *source* і *target* повинні містити ідентифікатори вузлів, визначених у тому ж документі що і ребро.

Додатковий *XML*-атрибут *directed* визначає спрямованість ребра, задану в явному вигляді. Значення *true* задає спрямоване ребро, а *false* – неспрямоване. Якщо спрямованість в явному вигляді не задана, то застосовується спрямованість задана за замовчуванням при оголошенні графу.

Додатково, за допомогою *XML*-атрибути *id*, може бути заданий ідентифікатор ребра, якщо необхідно організувати посилання на дане ребро.

За допомогою механізму розширення, який називається *GraphML*-атрибути, для елементів графа може бути задана додаткова інформація.

Для оптимізації синтаксичного розбору документа за допомогою парсера використовуються спеціальні метадані, які можуть бути додані до деяких *GraphML*-елементів за допомогою *XML*-атрибутів.

GraphML підтримує вкладені графи, тобто графи в яких вузли ієрархічно впорядковані. Ієрархія виражається через структуру *GraphML*-документа. Вузол в *GraphML*-документі може мати елемент *graph*, який містить вузли ієрархічно вкладені у даний вузол. Нижче наводиться приклад вкладеного графу і відповідний йому *GraphML*-документ (рис. 24).

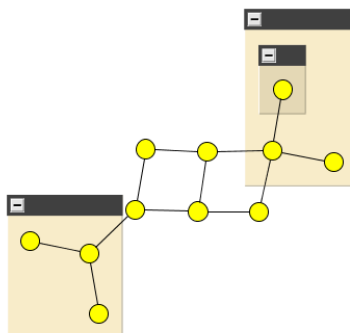


Рис. 24 – Вкладений граф.

Представленому графу відповідає *GraphML*-документ:

GraphML-документ з вкладеними графами:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="undirected">
    <node id="n0"/>
    <node id="n1"/>
    <node id="n2"/>
    <node id="n3"/>
    <node id="n4"/>
    <node id="n5">
      <graph id="n5:" edgedefault="undirected">
        <node id="n5::n0"/>
        <node id="n5::n1"/>
        <node id="n5::n2"/>
        <edge id="e0" source="n5::n0" target="n5::n2"/>
        <edge id="e1" source="n5::n1" target="n5::n2"/>
      </graph>
    </node>
    <node id="n6">
      <graph id="n6:" edgedefault="undirected">
        <node id="n6::n0">
          <graph id="n6::n0:" edgedefault="undirected">
            <node id="n6::n0::n0"/>
          </graph>
        </node>
        <node id="n6::n1"/>
        <node id="n6::n2"/>
        <edge id="e10" source="n6::n1" target="n6::n0::n0"/>
        <edge id="e11" source="n6::n1" target="n6::n2"/>
      </graph>
    </node>
    <edge id="e2" source="n5::n2" target="n0"/>
    <edge id="e3" source="n0" target="n2"/>
  </graph>
</graphml>
```



```
<edge id="e4" source="n0" target="n1"/>
<edge id="e5" source="n1" target="n3"/>
<edge id="e6" source="n3" target="n2"/>
<edge id="e7" source="n2" target="n4"/>
<edge id="e8" source="n3" target="n6::n1"/>
<edge id="e9" source="n6::n1" target="n4"/>
</graph>
</graphml>
```

Ребра, що з'єднують два вузли, що знаходяться у вкладеному графі, повинні бути оголошені в графі, який є предком обох вузлів в ієрархії.

GraphML підтримує поняття гіперребер. Гіперребро – це смислове об'єднання ребер яке не тільки пов'язує дві кінцеві точки, а й описує залежність між довільним числом кінцевих точок (наприклад, опис найкоротшого шляху – примітка перекладача). Гіперребра оголошуються за допомогою елемента *hyperedge*. Кожній кінцевій точці яка входить у дане гіперребро відповідає елемент *endpoint*. Елемент *endpoint* повинен мати *XML*-атрибут *node*, який містить ідентифікатор вузла в документі.

Вузли можуть містити різні логічні точки підключення ребер і гіперребер. Такі точки підключення називаються портами.

Порти вузла оголошуються за допомогою елементів *port*, які є дочірніми по відношенню до відповідного елементу *node*. Порти можуть бути вкладеними. Кожен елемент *port* повинен мати *XML*-атрибут *name*, який ідентифікує цей порт. Елемент *edge* має необов'язкові *XML*-атрибути *sourceport* і *targetport* які задають для ребра вихідний і вхідний порти вузла, відповідно. Аналогічно елемент *endpoint* має необов'язковий *XML*-атрибут *port*.

Питання для самоперевірки

1. Назвіть найбільш популярні інструменти візуалізації та аналізу мереж. Дайте їм порівняльну характеристику.
2. Що таке гіперребро?
3. Як називаються точки підключення ребер та гіперребер?

3. ЗАГАЛЬНІ ВІДОМОСТІ ПРО *GEPhi*. ІНТЕРФЕЙС КОРИСТУВАЧА

3.1. Інформація про програму

Gephi (<https://gephi.org/>) – це в даний час найпопулярніша програма візуалізації і аналізу мереж та графів («мережевих графів»). *Gephi* забезпечує швидку компоновку, ефективну фільтрацію та інтерактивне дослідження даних, а також є одним з кращих варіантів для візуалізації великомасштабних мереж. *Gephi* – це мультиплатформне програмне забезпечення яке розповсюджується з відкритим кодом згідно з ліцензіями *CDDL 1.0* і *GNU General Public License v3*. За адресою <https://gephi.org/> (рис. 25) доступні версії для *Mac OS X*, *Windows* і *Linux* вихідних кодів. Для роботи програми потрібна Java версія 7 і вище. У теперішній час програма локалізується для таких мов: англійська, французька, португальська, російська, китайська, чеська та німецька.

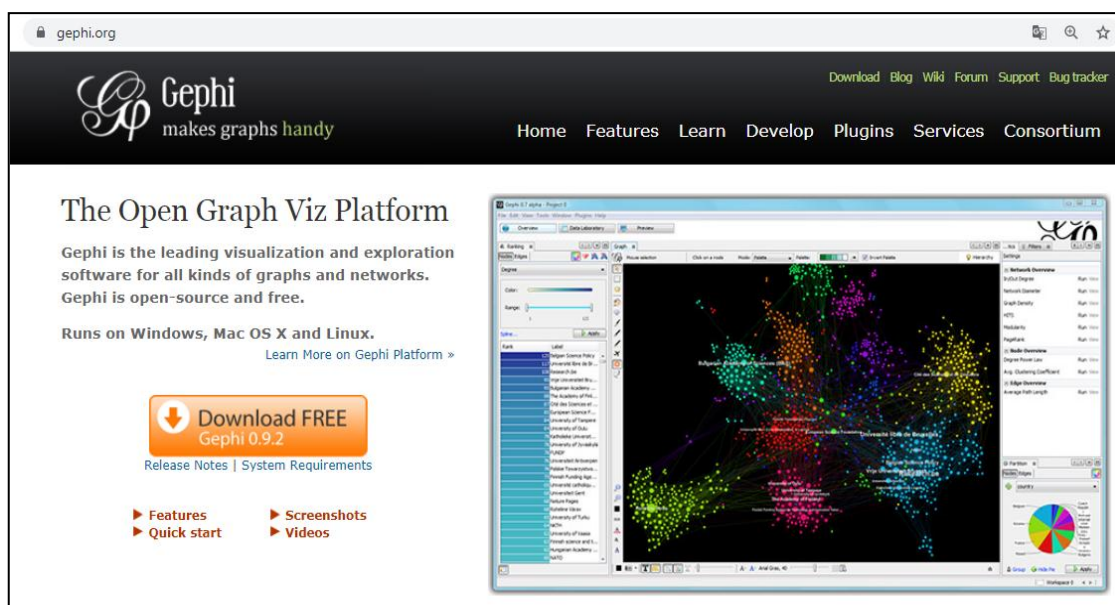


Рис. 25 – Стартове вікно *Gephi* – кнопка завантаження останньої версії

Розробники *Gephi* описують цю програму як "як *Photoshop*, але для даних".

Gephi дозволяє завантажувати дані мереж в форматах *GEXF*, *GDF*, *GML*, *GraphML*, *Pajek* (*NET*), *GraphViz* (*DOT*), *CSV*, *UCINET* (*DL*), *Tulip*

(*TPL*), *Netdraw* (*VNA*) і таблиць *Excel*. Крім того, *Gephi* дозволяє експортувати дані мереж в форматах *JSON*, *CSV*, *Pajek* (*NET*), *GUESS* (*GDF*), *Gephi* (*GEFX*), *GML* та *GraphML*. Завдяки цьому *Gephi* може взаємодіяти з іншими системами аналізу і візуалізації графів.

Програма включає в себе множини різних алгоритмів компонування (укладання графіків на площині) і дозволяє налаштувати кольори, розміри і мітки в графах. *Gephi* є інтерактивним програмним забезпеченням і надає засоби для виявлення спільнот, а також надається можливість розрахунку найкоротших шляхів або відносної відстані від будь-якого вузла до даного вузла. Плагіни від *Gephi* дозволяють розширювати її функціональність і додавати нові алгоритми, макети та інструменти вимірювань. *Gephi* має багатопотокову схему обробки даних, і таким чином, дозволяє виконувати кілька видів аналізу одночасно.

Інтерфейс користувача системи *Gephi* включає три основні розділи (вікна):

- «Лабораторія даних»: тут зберігаються всі вихідні дані про мережі, а також додаткові розрахункові значення;
- «Обробка даних»: тут відбувається велика частина операцій користувача, зокрема, ручне редагування мереж, тестування макетів, встановлення фільтрів;
- «Попередній перегляд»: тут уточнюється форма виводу графу, як правило, за допомогою набору інструментів граф допрацьовується, в тому числі, і з естетичної точки зору. У цьому ж вікні реалізований виклик експорту графа в формати *PDF*, *PNG* і *SVG*.

Ці три основні розділи охоплюють множину вкладок, які дозволяють користувачу реалізовувати окремі функції. Нижче розглядається кожне з основних і вторинних вікон – розділів і вкладок.

3.2. Основні розділи

«Лабораторія даних»

Усі дані, на основі яких створюються мережеві графи, зберігаються в «Лабораторії даних» – сховищі графічних даних. Хоча лабораторія даних може мати вигляд електронної таблиці, її функціональність не слід плутати з *Excel* або *Google Spreadsheet*. Деякі операції по обробці даних можуть бути виконані саме тут, проте, краще всього, щоб базові дані про мережу були підготовлені ще до їх імпорту в *Gephi*. Простіше використовувати інструменти електронних таблиць, коли є необхідність створювати різні масиви великих об'ємів. Аналогічно, значення полів, які ґрунтовані на певній схемі сортування, краще всього створювати поза *Gephi*.

Сказане не означає, що дані, які зберігаються в лабораторії, повністю статичні. Наприклад, всі статистичні обчислення і обчислення кластеризації будуть автоматично додавати нові значення до кожного вузла під час запуску процесу. Також є можливість додавати стовпці до таблиці, копіювати дані з одного стовпчика в інший, видаляти стовпці і так далі. Проте, внесення масових змін на рівні вузлів або ребер може виявитися дуже трудомістким, особливо якщо досліджуваний набір даних про мережу складається з тисяч значень.

Вікно «обробки даних»

Усі дані про мережу спочатку переглядаються у вікні обробки даних, в якому *Gephi* забезпечує первинне представлення досліджуваної мережі. Первісний вигляд мережі може бути примітивним, але потім з цим поданням проводиться спеціальна обробка. Усі додатки, що пов'язані з укладанням мережі, фільтрацією, сегментуванням, розфарбуванням і будь-які інші налаштування макета, видно в першу чергу саме в цьому вікні.

Вікно графіку прилягає до кількох панелей інструментів, кожна з яких містить множину функцій. Функціональність кожного з цих варіантів, як правило, інтуїтивно зрозуміла.

Вікно «попереднього перегляду»

Вікно попереднього перегляду *Gephi* дозволяє користувачеві налаштувати різні атрибути, які були створені у вихідному вікні графіку. Тут можна налаштувати мітки вузлів, вибрати шрифт, його розмір, колір, контури, мітки і т.п. Ці рішення можуть бути прийняті на основі щільності і складності графу: в щільних великих графах рекомендується позначати тільки найбільші вузли.

Зовнішній вигляд вузла також задається вказанням параметрів ширини границі, кольору границі і прозорості. При цьому завжди можна переключитися у вікно обробки даних, щоб виконати чимало з налаштувань в *Gephi*, а потім перейти у вікно попереднього перегляду і оновити відображення графу.

Для налаштування зовнішнього вигляду ребер графу передбачені такі опції, як налаштування товщини ребер, кольору, прозорість, можливість викривлених ребер, установки міток. Для орієнтованих ребер можна налаштувати стрілки ребер.

Вікно попереднього перегляду також містить вбудовані параметри експорту *Gephi*, зокрема, експорт у формати *SVG*, *PDF* і *PNG*:

- *PNG* – найпростіший вибір. Ця опція забезпечує вивід рисунку з графом у графічному форматі *PNG*, але явно обмежена з точки зору редагування;
- *SVG* – експорт до *SVG* створює масштабовану векторну графіку, яку можна редагувати в інших програмах, таких як *Inkscape*;
- *PDF* – експорт до *PDF* дозволяє створювати файл, який потім можна редагувати як файл *PDF* з використанням чисельних програм-

редакторів, що в кінцевому підсумку дозволить налаштувати заголовки або інші позначення, що описують граф.

3.3. Вторинні вікна-вкладки

Gephi надає множину вторинних вкладок, які «оточують» робочий простір (вікно режиму), дозволяючи користувачу виконувати дії на графіку без необхідності перемикання між декількома вікнами. Це полегшує користувачу ітераційний підхід до управління та аналізу графіка.

Вкладка фільтрація

Вкладка фільтрація – це те місце, де застосовується ряд критеріїв для скорочення розміру досліджуваної мережі, щоб отримати краще уявлення про неї. Якщо мережа велика і щільна, то в ній важко орієнтуватися.

Фільтри надають необхідні інструменти для систематичного дослідження графіка, пошуку найбільш важливих параметрів або особливостей графа. *Gephi* надає можливість створення індивідуальних або складених фільтрів, в яких враховується декілька умов.

Вкладка статистика

Вкладка статистика надає великий обсяг інформації про параметри мережі і окремих її елементів, які можуть бути використані для кращого розуміння структури мережі.

Головними серед цих статистичних даних є різні міри центральності вузлів. Серед інших мір можна виділити: діаметр графа, коефіцієнти кластеризації, найкоротша відстань між вузлами та ін. Багато з цих параметрів включені в базову установку *Gephi*, а інші доступні через обрані плагіни.

Вкладка укладки

Gephi дозволяє досліджувати мережеві дані, використовуючи широкий спектр укладок графів, що дозволяє користувачам вибрати найкращий варіант для представлення графа. При цьому *Gephi* надає можливість протестувати чимало алгоритмів компоновання, перш ніж зупинитися на остаточному виборі. Багато алгоритмів компоновання використовують параметри, які дозволяють визначати ідеальне розміщення графа шляхом маніпулювання з тяжінням, відштовхуванням, гравітацією та іншими налаштуваннями.

Разом з цим можна вибрати заздалегідь визначену схему, яка представить мережу у вигляді кола або набору концентричних кіл, упорядкованих відповідно до якогось алгоритму ранжування.

Модулі (плагіни), що підключаються

Основна ідея використання плагінів в *Gephi*, як і плагінів для інших програм, полягає в додаванні функцій, які не завжди доступні в базовому варіанті програмного забезпечення. У деяких випадках плагіни представлені у вигляді функцій, які допомагають користувачам краще форматовувати графи, а в інших випадках – плагіни є реалізацією повноцінних алгоритмів компоновання або генерації графів, які дають додаткові можливості для створення і аналізу графів.

Велика кількість плагінів, що розширюють основні функції *Gephi*, знаходиться у вкладці *Service→Plugins* (рис. 26). Процес інсталяції плагінів дуже простий, а вимоги до пам'яті – мінімальні.

Після інсталяції *Gephi* програму можна запустити, в результаті чого на екрані користувача з'явиться зображення (рис. 27).

3.4. Створення графа в *Gephi*

Існує три основні режими створення нового графа в *Gephi*:

- через інтерфейс «Граф» в режимі «Обработка»;
- через «Лабораторію даних»;
- через экспорт даних графа із зовнішнього файлу (найпростіше з файлу в форматі CSV з роздільниками крапка з комою).

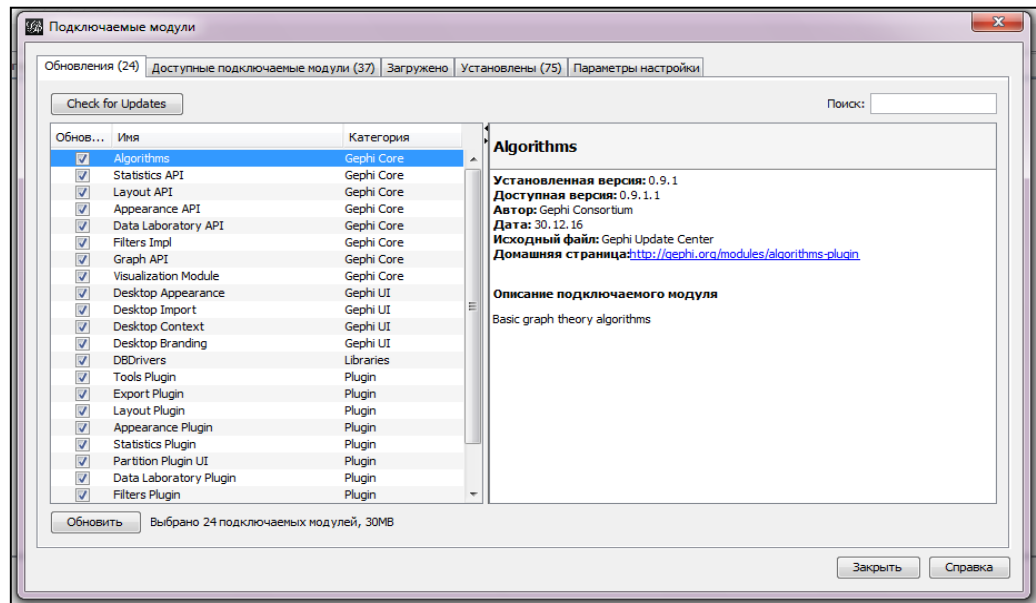


Рис. 26 – Вкладка «Модулі, що підключено»

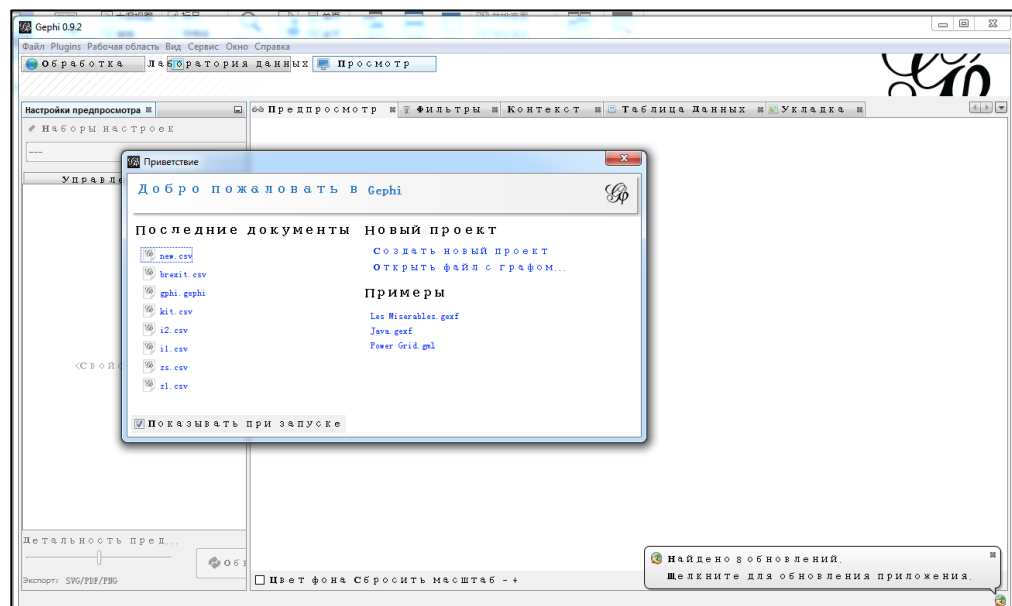


Рис. 27 – Стартовое вікно Gephi

Перейдемо до розгляду першого випадку.

3.5. Створення графа через інтерфейс «Граф»

Після закриття спливаючого при завантаженні екрану вікна, відразу ж активізується інтерфейс «Робоча область» (Обработка), в рамках якої можна створити новий граф. Для цього досить активізувати новий проект і скористатися інструментами, позначеними в правій частині вікна (рис. 28).

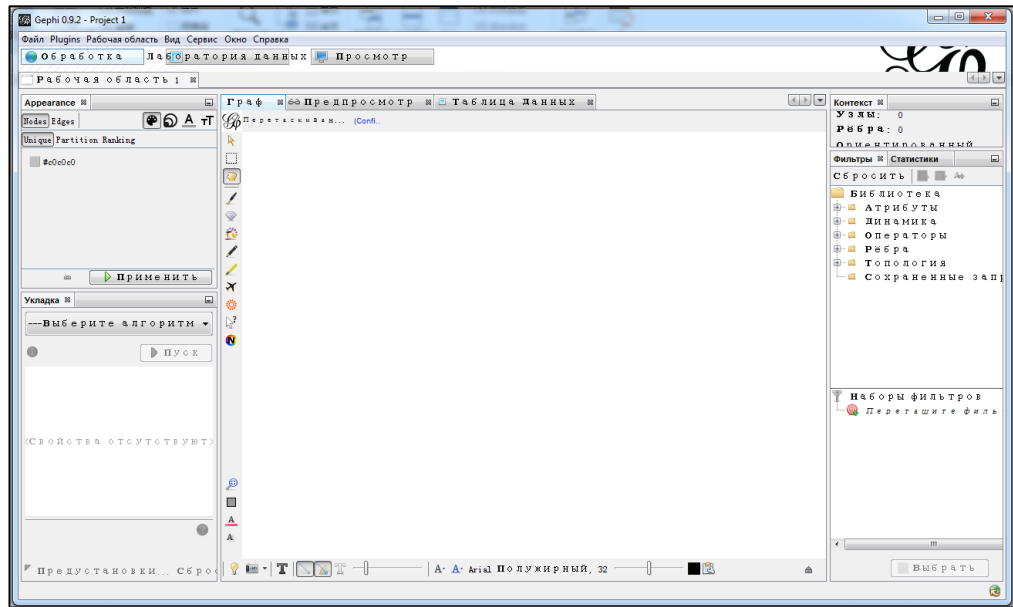


Рис. 28 – Робоча область, вкладка «Граф»

Для ручного нанесення вузлів за допомогою власних інструментів *Gephi* слід скористатися кнопкою «Олівець для малювання вузлів» (рис. 29).

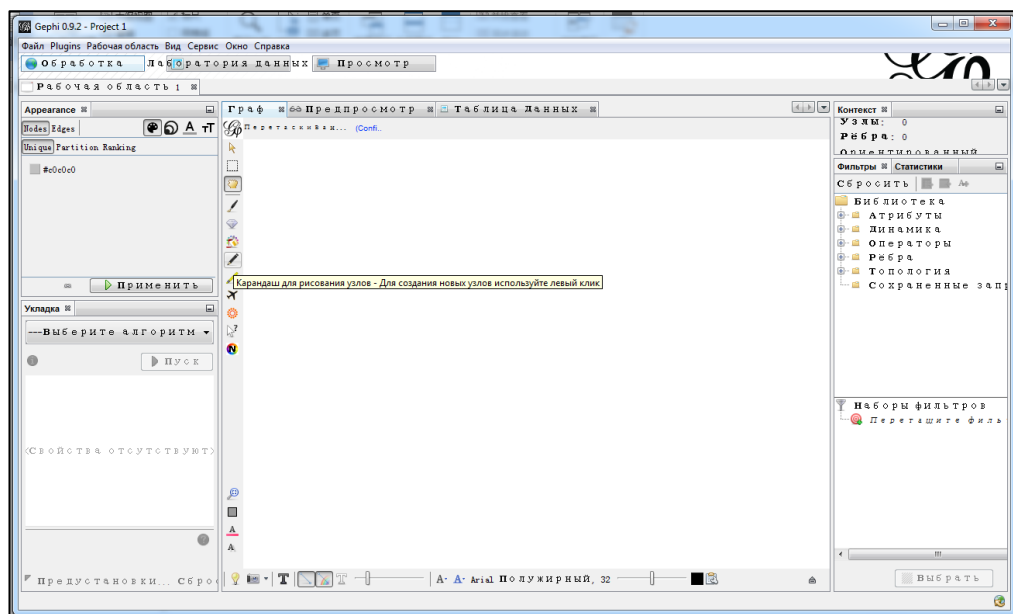


Рис. 29 – Інструмент «Олівець для малювання вузлів»

Вибравши місце на екрані «Олівцем для малювання вузлів» можна нанести нові вузли і за допомогою інструменту з меню «Розмір» (піктограма – розмір графа) збільшуємо їх (рис. 30).

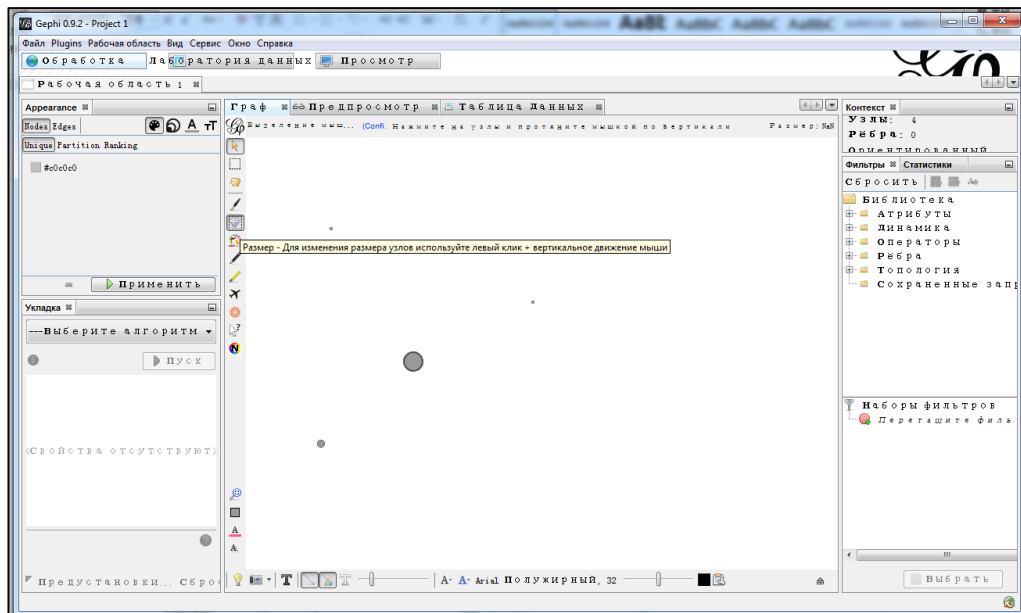


Рис. 30 – Інструмент «Зміна розміру вузлів»

За допомогою вибору інструменту «Олівець для малювання ребер» (рис. 31) розставляємо ребра графа. Спочатку робимо клік миші, перебуваючи на початковому вузлі, а потім, на вихідному. Отримуємо спрямоване ребро (у верхній частині інтерфейсу відображається тип графа – за замовчуванням «орієнтований»).

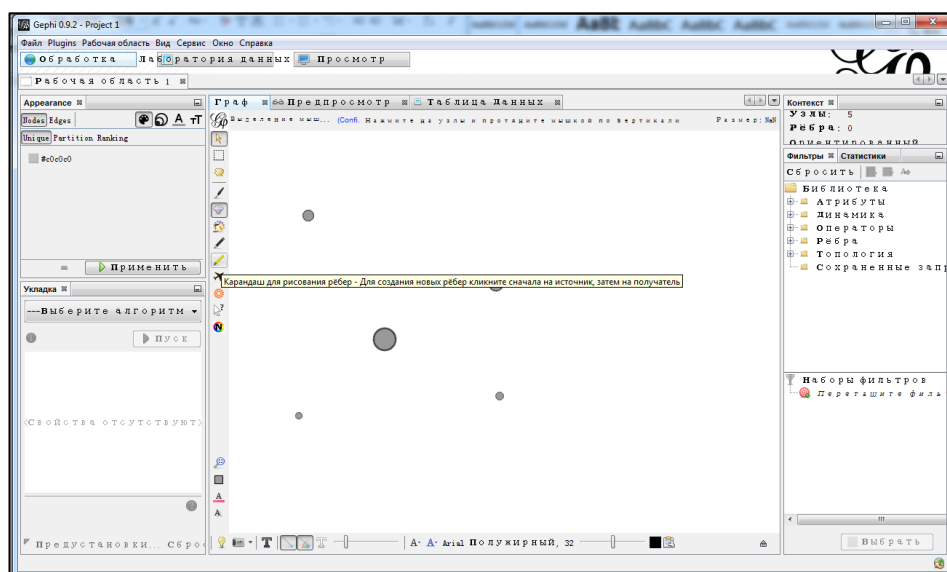


Рис. 31 – Інструмент «Олівець для малювання ребер»

Після цього налаштовуємо товщину ребра, для чого розкриваємо нижню область вкладки «Граф».

Після розстановки ребер можна перейти в режим розфарбовування (піктограма «фарба») (рис. 32).

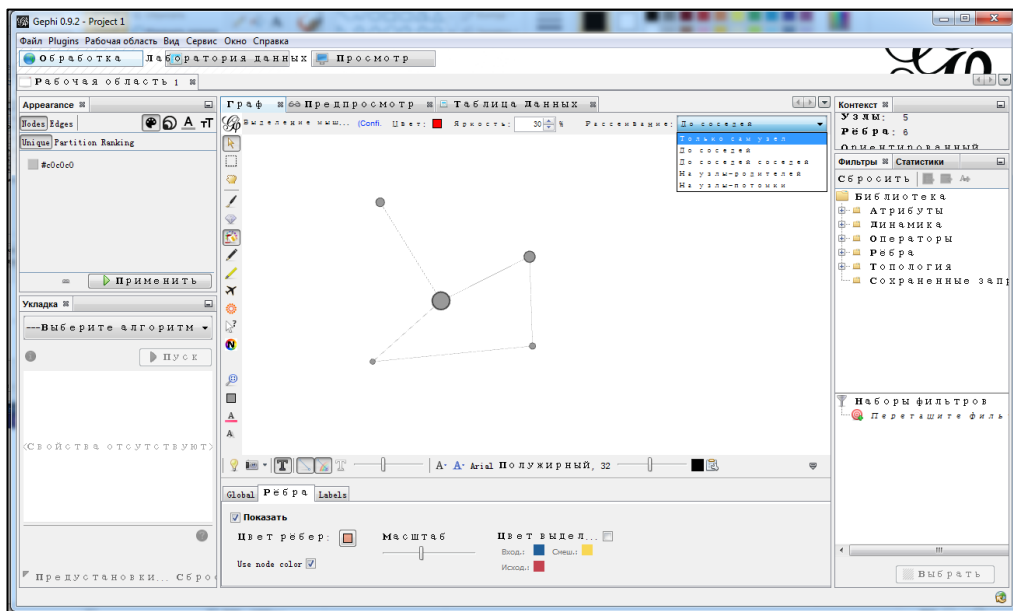


Рис. 32 – Вибір режимів розфарбовування графа

При цьому існує можливість або зафарбовувати всіх сусідів обраного вузла, або скористатися індивідуальним зафарбуванням (верхнє праве меню робочої області). У результаті чого отримуємо остаточно сформований граф (рис. 33).



Рис. 33 – Результат розфарбування

3.6. Створення графа через інтерфейс «Лабораторія даних»

Для створення графа також зручно перейти в режим «Лабораторія даних», де в табличній формі відображається вся інформація про поточний

стан графа (рис. 34). Причому інформація відображається у вигляді, придатному для зміни (редагування). Як видно з рисунку, можна додавати нові ребра (вузли), видаляти або змінювати існуючі.

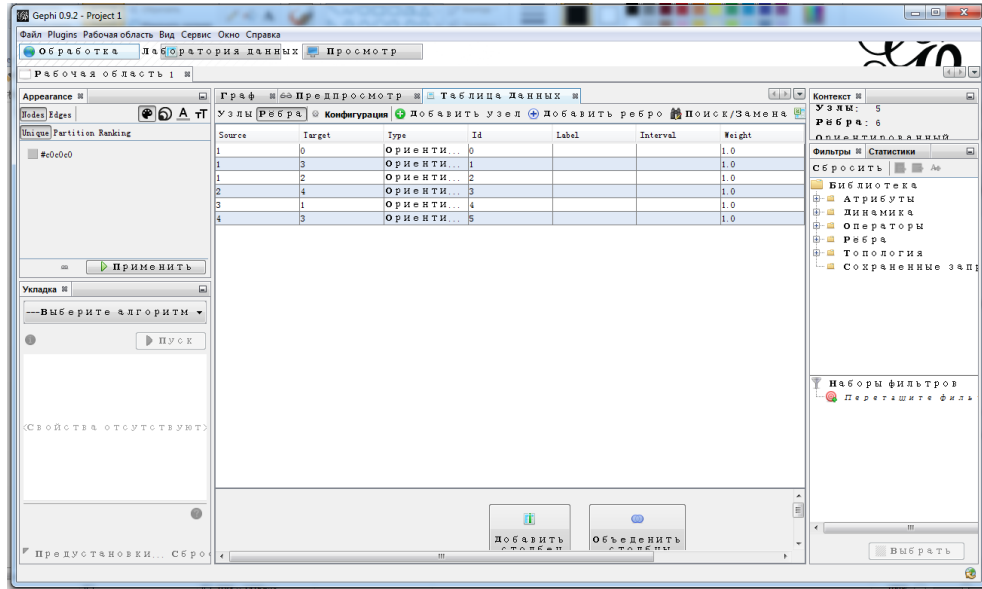


Рис. 34 – Интерфейс режима «Лаборатория данных»

«Лаборатория данных» зручна також і для нанесення текстових міток вузлів (рис. 35).

Граф		
Предпросмотр		
Таблица Данных		
Узлы Ребра Конфигурация + Добавить узел + Добавить ребро Поиск/Замена		
Id	Label	Interval
0	Узел 0	
1	Узел 1	
2		
3		
4		

Рис. 35 – Метки вузлів в інтерфейсі «Таблица данных»

У результаті на вкладці «Граф» відображаються мітки вузлів і ребер, шрифти, розмір та яскравість яких, можна змінювати за допомогою інструментів, представлених в нижній частині екрана (рис. 36).

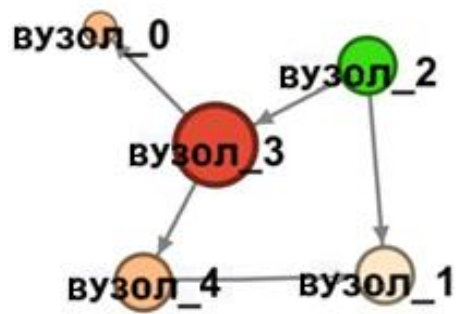


Рис. 36. Відображення міток вузлів

Розмір і колір вузлів можна встановити в залежності від їх загальної, вхідної або вихідної степеней (рис. 37, 38).

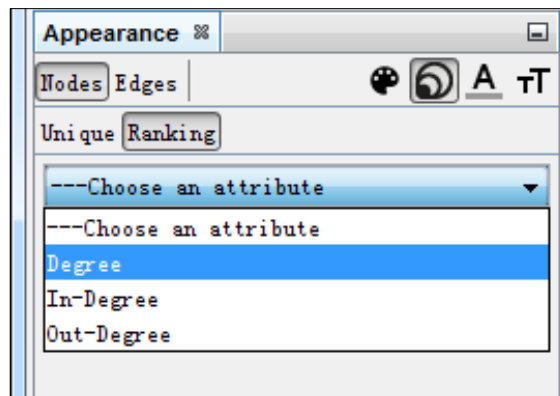


Рис. 37 – Вибір градації розмірів вузлів

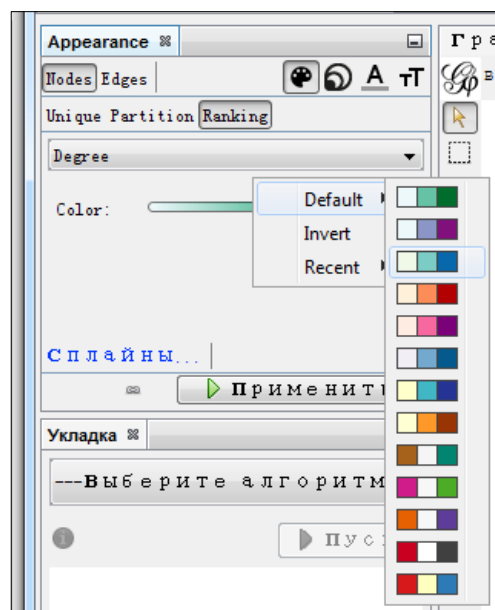


Рис. 38 – Налаштування кольору вузлів

3.7. Експорт даних графа з зовнішнього файлу

Дані графа можна завантажувати у *Gephi* з текстового формату, в якому елементи-мітки вузлів розділені знаком "крапка з комою". У цьому випадку до вузла, що відповідає першій мітці в рядку "приєднуються" усі інші вузли, мітки яких приведені в цьому рядку. Наприклад, нехай зовнішній файл містить такі записи:

```
Node1;Node2;Node3;Node4;Node5  
Node5;Node3
```

У цьому випадку після завантаження в систему *Gephi* та їх обробки (підготовки до візуалізації вже описаним способом), отримуємо відображення (рис. 39):

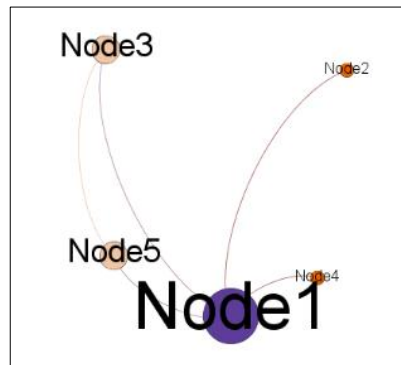


Рис. 39. Відображення графа після завантаження і обробки

Однак основний варіант експорту даних графа із зовнішнього файлу – це завантаження початкових мережевих даних в форматі *CSV*, в якому елементи розділені знаком «крапка з комою». У цьому випадку у *CSV*-файлі фактично повинна знаходитися розширена мітками матриця інцидентності мережі. Нижче наводиться приклад для мережі з п'яти вузлів:

```
;Node1;Node2;Node3;Node4;Node5  
Node1;0;1;0;1;0  
Node2;1;0;0;1;0  
Node3;0;1;0;0;1  
Node4;1;1;1;0;0  
Node5;0;1;0;1;0
```

Після завантаження в систему *Gephi* і обробки вже описаним способом, отримуємо відображення (рис. 40):

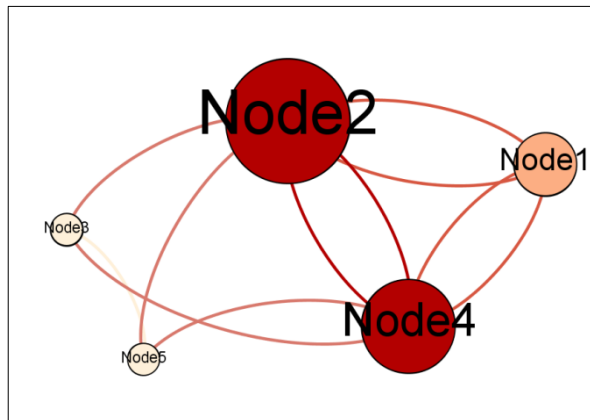


Рис. 40 – Відображення графа, завантаженого з файлу в форматі CSV

Наведений вище файл можна підготувати в програмі *Excel*, після чого зберегти його у форматі *CSV*. Слід звернути увагу, що перед завантаженням в *Gephi* необхідно в *CSV*-файлі поміняти усі коми («,») на крапки з комами («;») – така особливість *Gephi*.

На рис. 41 наведено приклад підготовки файлу в *Excel* для його подальшого збереження в форматі *CSV*.

	A	B	C	D	E	F	G
1	Node1	Node1	Node2	Node3	Node4	Node5	
2	Node1	0	1	0	1	0	
3	Node2	1	0	0	1	0	
4	Node3	0	1	0	0	1	
5	Node4	1	1	1	0	0	
6	Node5	0	1	0	1	0	
7							
8							

Рис. 41 – Приклад підготовки файлу в *Excel*

Питання для самоперевірки

1. Чому *Gephi* одна з найпопулярніших, у теперішній час, програма візуалізації та аналізу мереж?
2. На яких платформах працює *Gephi*?
3. Назвіть основні формати файлів, які *Gephi* дозволяє завантажувати для аналізу.
4. Назвіть основні режими створення графа у *Gephi*. Охарактеризуйте їх.

4. КОМПОНОВКА ТА РАНЖУВАННЯ

При аналізі великих і щільних мереж, швидке компонування (впорядкування вузлів графів) є вузьким місцем, оскільки більшість складних алгоритмів компонування є вимогливими до параметрів процесора, пам'яті і часу виконання. У той же час, *Gephi* постачається з ефективними алгоритмами компонування, такими як *Yifan-Hu*, *Force-directed*. Зокрема, алгоритм *Yifan-Hu* є ідеальним варіантом для застосування після інших, більш швидких і грубих алгоритмів. У той час, як більшість із запропонованих у *Gephi* методів можуть виконуватися протягом допустимого часу, поєднання, наприклад, *OpenOrd* і *Yifan-Hu*, дає найбільш якісні візуальні уявлення. Звичайно, правильна параметризація будь-якого алгоритму компоновки може впливати як на час роботи, так і на результат візуалізації.

4.1. Компоновка тестового графу

Якщо для побудованого в попередньому розділі графу перейти в режим «Укладка графа» і обрати режим *Fruchterman Reingold*, то буде одержаний результат, представлений на рис. 42.

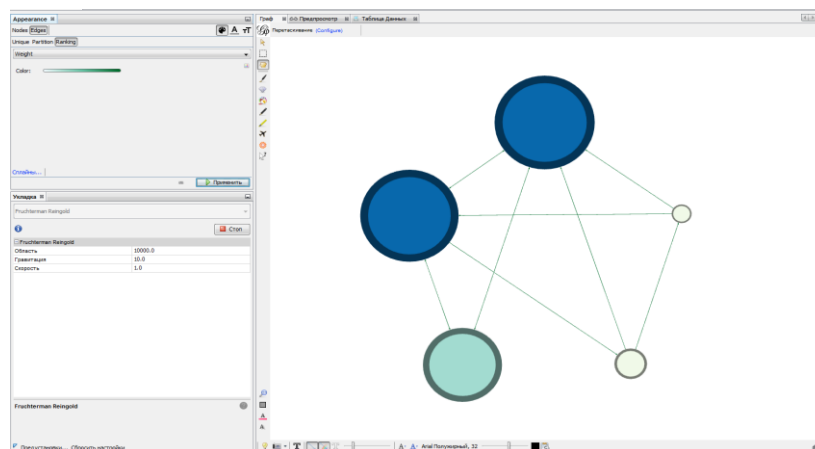


Рис. 42 – Відображення тестового графа в режимі *Fruchterman Reingold*

Після цього можна перейти в режим «Перегляд», задати в лівій панелі параметри виведення найменування вузлів (пункт меню «Мітки вузлів»: «Відображати мітки», «Пропорційний розмір», розмір шрифту 3 пт та ін.), а потім натиснути клавішу «Оновити», що знаходиться внизу панелі. Отримаємо відображення (рис. 43), яке потім можна зберегти у форматах *PDF*, *PNG* і *SVG*.

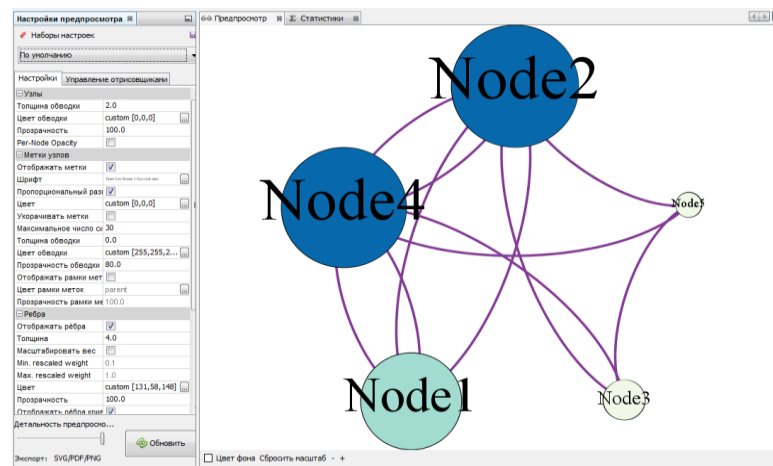


Рис. 43 – Відображення навчального графа в режимі «Перегляд»

4.2. Укладка графу *Les Miserables*

При завантаженні набору даних *Les Miserables*, представленого в форматі *GEXF*, програма *Gephi* відображає спливаюче вікно (рис. 44), в якому йдеться про те, що завантажений неорієнтований граф має 77 вузлів і 254 зв'язки. Вузлам, звичайно, відповідають герої роману «Знедолені».

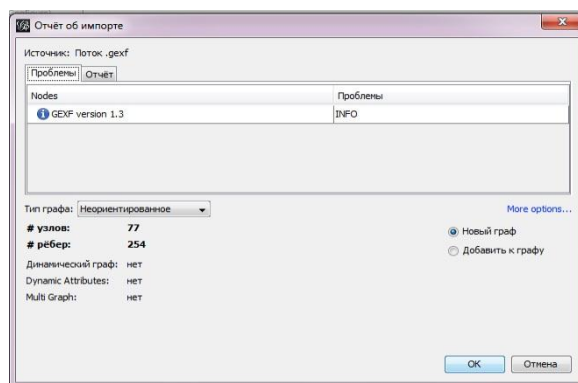


Рис. 44 – Інформація про завантажені мережеві дані

Після завантаження набору даних в режимі «Обробка» у вікні «Граф» з'явиться вже оброблений граф (рис. 45) з детально відпрацьованим компонуванням. Навмисне змінимо цю компоновку (режим випадкова укладка, рис. 46), скасуємо ранжування вузлів мережі, щоб продемонструвати різні види укладки.

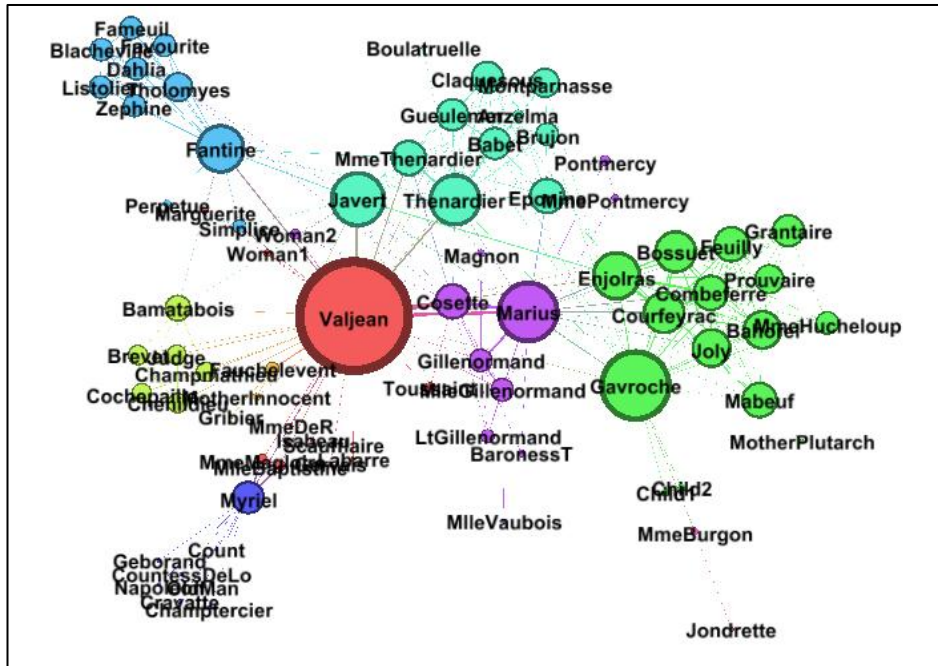


Рис. 45 – Завантажений граф *Les Miserables*

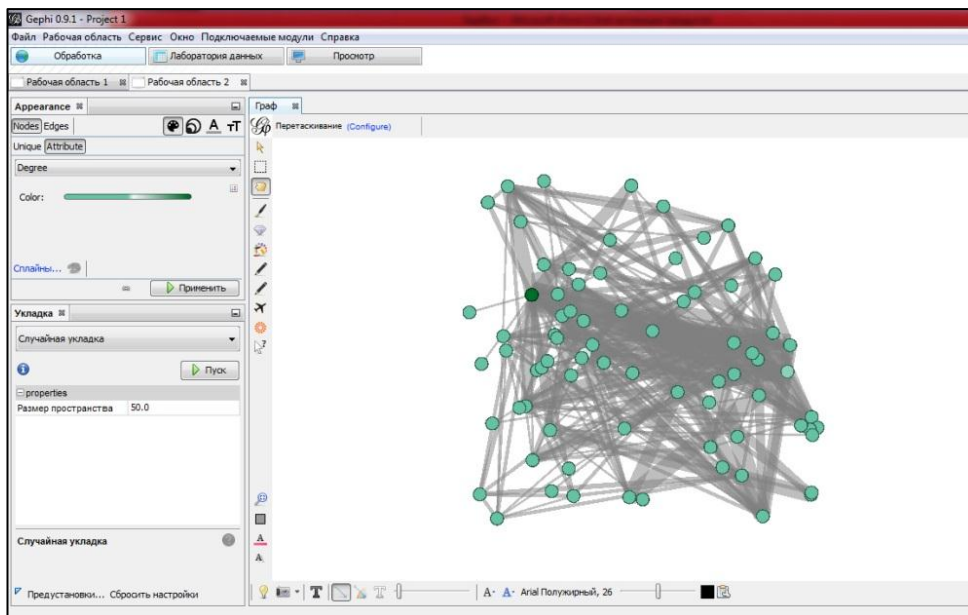


Рис. 46 – Випадкова укладка графа

На рис. 47 наведені результати роботи різних алгоритмів укладки. Як можна бачити, кращі результати дають у даному випадку алгоритми *Force Atlas* і *Yifan Hu*.

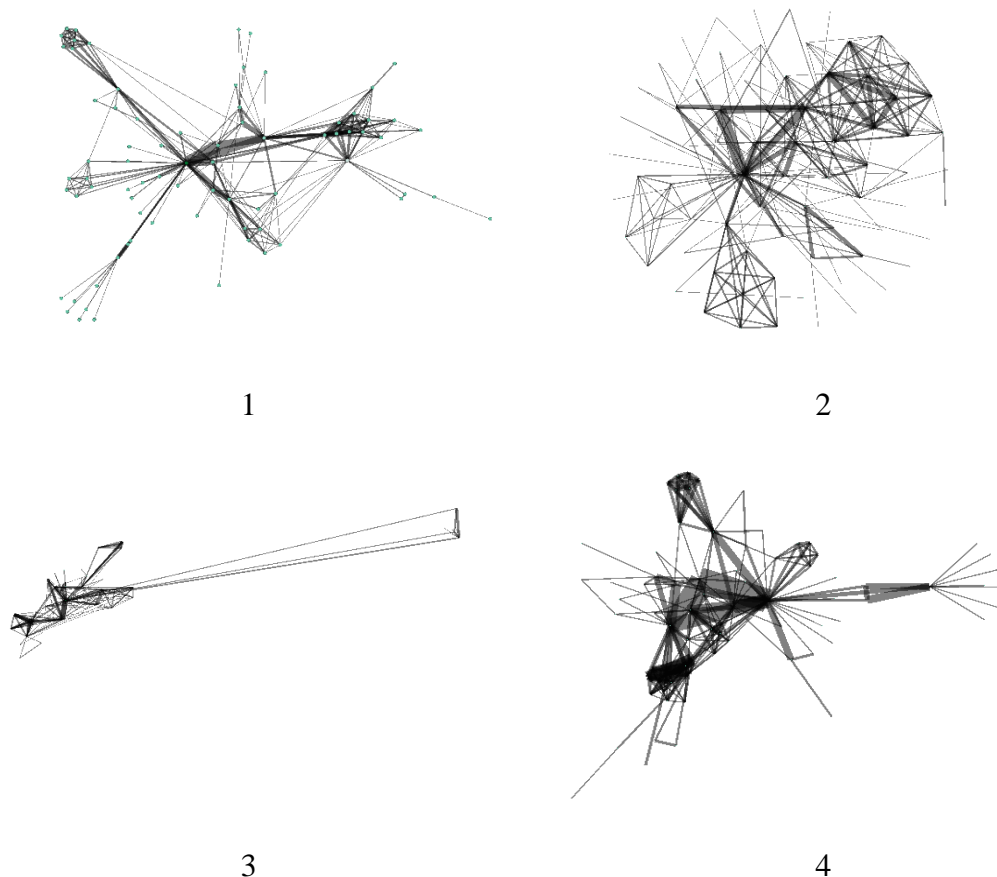


Рис. 47 – Результати застосування різних алгоритмів укладки:
1 – *Force Atlas* ; 2 – *Fruchterman Reingold*; 3 – *OpenOrd*; 4 – *Yifan Hu*

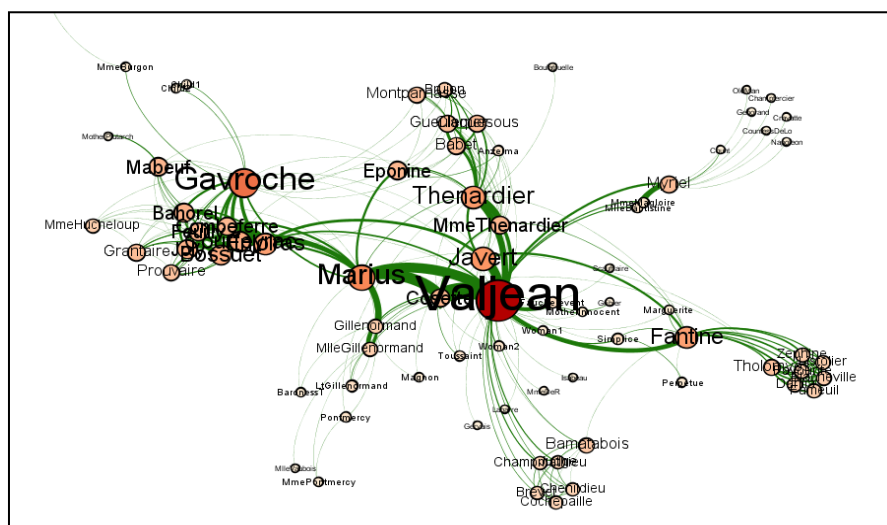


Рис. 48 – Результати обробки алгоритмом *Force Atlas* в режимі «Перегляд»

Ранжуючи вузли і ребра за ступенем і вагою відповідно, і перейшовши в режим перегляд за допомогою вже розглянутих інструментів, отримуємо підсумкове відображення (рис. 48).

4.3. Ранжування і статистика

Ранжування дозволяє встановлювати розміри вузлів і фарбувати граfi на основі зазначеного користувачем атрибута, такого як вага, ступень (за замовчуванням) і багато іншого, у залежності від доступних полів в нашому наборі даних. Багато з таких полів, наприклад, значення *PageRank*, модульність, можна отримати в режимі «Лабораторія даних» у вкладці «Статистика» (рис. 49), яка знаходиться у правій частині інтерфейсу. Цей розділ дозволяє отримувати різні спеціальні розрізи статистики для подальшого аналізу різних параметрів графів.

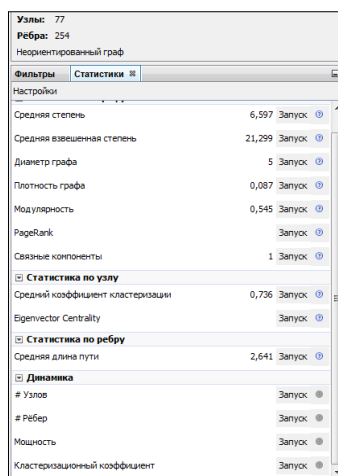


Рис. 49 – Аналіз мережі – режим «Лабораторія даних». Папка «Статистика»

При послідовному натисканні всіх кнопок «Запуск» поспіль, можна отримати значення характеристик графа (рис. 50).

Id	Label	Modularity C...	PageRank	Clustering Coeff...	Number of tra...	Component...	Сумарна пот...	Вісцевана пот...	Eccentricity	Closeness Centrality	Harmonic Closeness C...	Betweenness Centr...	Eigenvector Centra...
11	Valjean	0	0.075436	0.120635	76	0	36	158.0	3.0	0.644068	0.724566	1624.4688	1.0
48	Gervoise	1	0.035762	0.354978	82	0	22	56.0	3.0	0.513514	0.605263	470.570632	0.995942
55	Marius	0	0.030861	0.333333	57	0	19	104.0	3.0	0.531469	0.60307	376.292393	0.828965
27	Javert	0	0.030302	0.323529	44	0	17	47.0	3.0	0.517007	0.585526	194.844945	0.676536
25	Thouard	0	0.027925	0.400333	49	0	16	61.0	3.0	0.517007	0.581114	213.468481	0.676465
23	Fantine	3	0.027022	0.314286	33	0	15	47.0	4.0	0.460606	0.539474	369.486942	0.404673
58	Enjolras	1	0.023878	0.609524	64	0	15	91.0	3.0	0.481013	0.552632	121.277067	0.816259
62	Courfeyrac	1	0.018574	0.75641	59	0	13	84.0	4.0	0.463553	0.510335	15.011035	0.680372
64	Bossuet	1	0.018958	0.760231	60	0	13	86.0	3.0	0.475	0.539474	87.847903	0.72414
63	Bahaud	1	0.017196	0.863636	57	0	12	39.0	4.0	0.393782	0.472588	6.228642	0.643606
65	Joly	1	0.017196	0.863636	57	0	12	43.0	4.0	0.393782	0.472588	6.228642	0.643606
24	MmeThena...	0	0.019501	0.490909	27	0	11	34.0	4.0	0.460606	0.520833	82.656893	0.463376
26	Cosette	0	0.020111	0.381818	21	0	11	68.0	4.0	0.477987	0.533991	67.819322	0.420949
41	Eponine	0	0.017192	0.454545	25	0	11	19.0	4.0	0.395833	0.470395	32.739519	0.477549
57	Mabeuf	1	0.017475	0.690909	38	0	11	16.0	4.0	0.395833	0.470395	78.834524	0.576356

Рис. 50 – Дані у вкладці «Таблиця даних»

Розраховані характеристики доступні у вкладці «Таблица даних».

Використаємо результати роботи в режимі «Лабораторія даних» для візуалізації. Після розрахунку відповідних параметрів з'являється можливість змінювати розміри і кольори вузлів не тільки за степенем, але і за іншими характеристиками, таким як модульність, кластерність і т.п.

Якщо повернутися до ранжирування в режимі «обробка», то побачимо додаткові можливості (рис. 51).

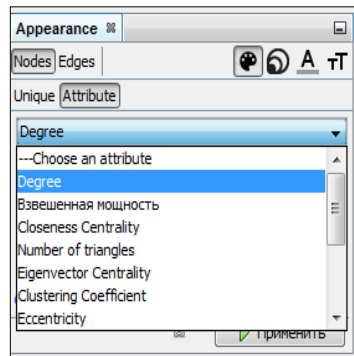


Рис. 51 – Критерії ранжування для кольорів вузлів

Застосувавши ранжування за класами модульності, можна виділити кластери мережі (рис. 52).

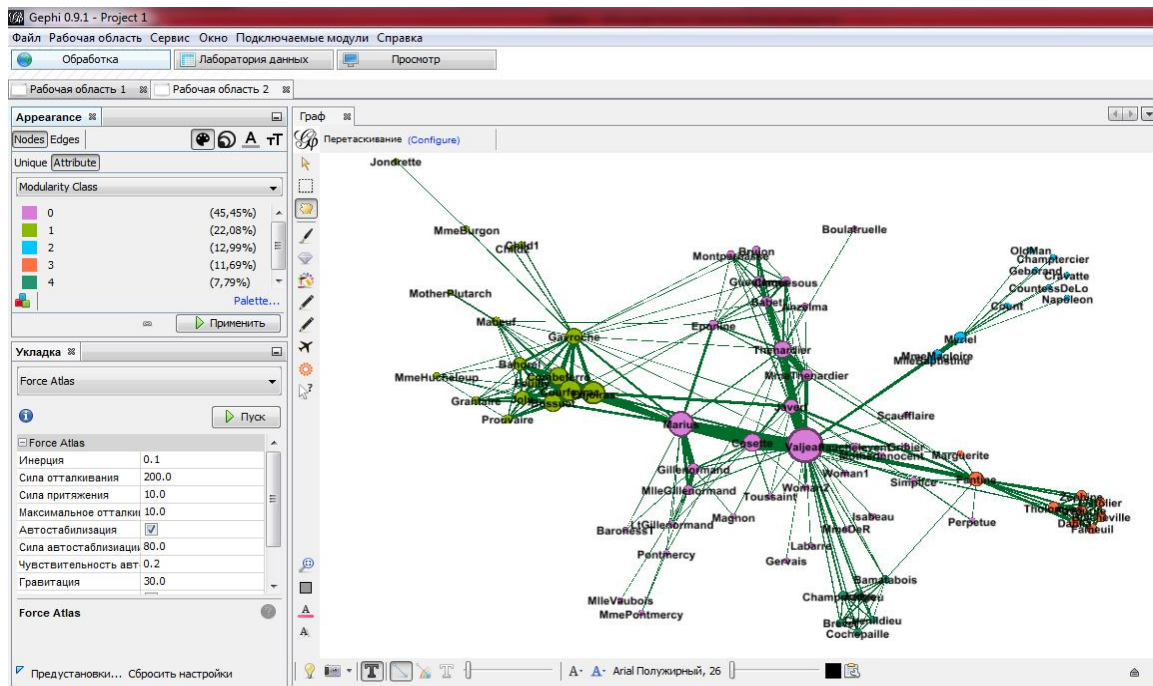


Рис. 52 – Підсумкове розфарбування вузлів

4.4. Фільтрація мережі

Причина, за якою необхідна фільтрація мережевих даних зрозуміла: часом неможливо візуально аналізувати великі мережі, ребра яких через їх кількість утворюють на екрані суцільне покриття. При цьому стає неможливим свідомо переглядати щільні графи, хоча вони і відображаються за допомогою *Gephi*.

Розумна фільтрація повинна раціонально скоротити кількість вузлів (відповідно і зв'язків) мережі та забезпечити зменшення її візуальної складності. При цьому мається на увазі, що скорочення кількості вузлів мережі не здійснить істотного впливу на базову структуру даних. Таким чином, раціональна фільтрація скорочує кількість несуттєвих зв'язків, дозволяє користувачеві зосередитися на тих фрагментах, які мають найбільше значення для цільового завдання.

Деякі елементи мережі, безумовно, можна розрізняти, зокрема, врегулюємо *Gephi* їх розмір і колір, вибираючи варіанти компоновки. Фільтрація забезпечує наочність, яку важко досягти за допомогою будь-яких інших засобів цієї системи. При видаленні з поля зору несуттєвих вузлів і зв'язків забезпечується зниження складності, скорочується кількість проблем при інтерпретації графу. Різні види фільтрації можуть забезпечити різнопланову візуалізацію мережі, привести до кількох ідей при інтерпретації цільового завдання.

Основні функції фільтрації в Gephi

Фільтри *Gephi* поділяються на кілька груп, які відображаються у вигляді окремих папок у вікні (рис. 53). У кожній з цих папок представлено кілька варіантів фільтрації, які можуть використовуватися самостійно як прості фільтри, або комбінуватися для створення складних фільтрів. Первинні категорії фільтрів включають:

- Атрибути: ця папка містить множину параметрів, які дозволяють фільтрувати вузли, ребра, кластери і різні параметри графа, такі як ексцентриситет і різні рівні централізації. Крім того, в цій папці можуть бути знайдені і оброблені визначені користувачем атрибути (наприклад, новий стовпець);
- Ребра: цей фільтр застосовується суворо до зв'язків всередині мережі;
- Оператор: цей фільтр дозволяє виконувати кілька функцій на графіку;
- Топологія: цей фільтр пропонує ряд опцій, які дозволяють використовувати графічні заходи, такі як діапазони степеней вузлів для фільтрації мережі.

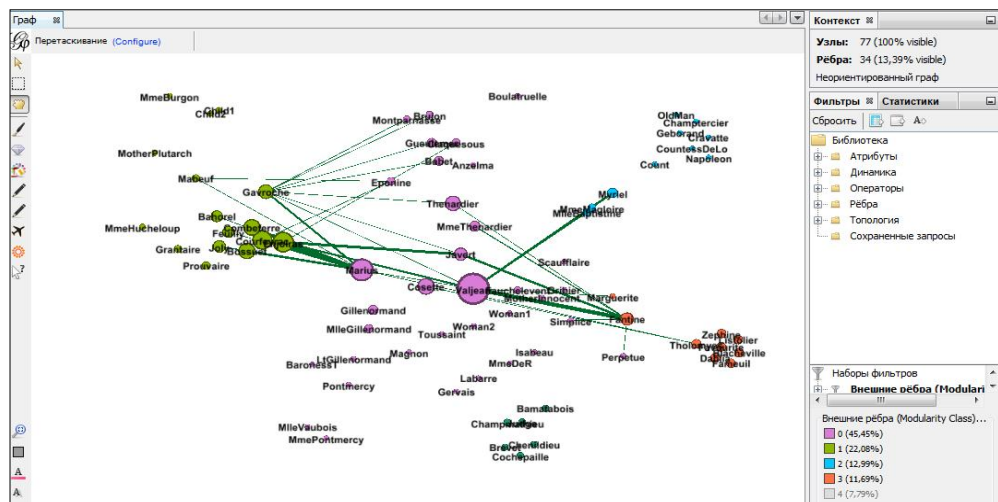


Рис. 53 – Застосування функції фільтрації за класами модулярності

Питання для самоперевірки

1. Для чого необхідна фільтрація мережі?
2. Розкрийте основні функції фільтрації мережі.
3. На основі яких параметрів може здійснюватися ранжування вузлів мережі?
4. Порівняйте алгоритми, які застосовуються в *Gephi* для укладки графів.

5. РЕАЛЬНІ МЕРЕЖІ. ВІЗУАЛІЗАЦІЯ ТА АНАЛІЗ

5.1. Тестові мережеві дані програми *Gephi*

Нижче наведені приклади наборів даних (*Datasets*) для *Gephi*, які доступні в різних форматах (*GEXF*, *GDF*, *GML*, *NET*, *GraphML*, *DL*, *DOT*). Одна з цих мереж, побудована за твором Гюго «Знедолені» (*Les Miserables*), вже використовувалася у 4 розділі.

Web i Internet

GEXF-файл (<https://gephi.org/datasets/eurosis.gexf.zip>). *EuroSiS web mapping study*: Відображення взаємовпливу між наукою і суспільством в сегменті веб-простору 12-ти європейських країн. Оригінальний звіт і дані знаходяться за адресою <http://www.webatlas.fr/exhibition/eurosis/>.

GML-файл (https://gephi.org/datasets/internet_routers-22july06.gml.zip).

Internet: симетричний знімок структури інтернету на рівні автономних систем, реконструйований з таблиць *BGP*, розміщених університетом *Oregon Route Views Project* (<http://routeviews.org/>). Цей знімок був створений Марком Ньюманом 22 липня 2006 року.

Соціальні мережі

- *GEXF*-файл (<https://github.com/seinecle/gephi-tutorials/blob/master/docs/resources/miserables-with-dates.gexf>). *Les Miserables*: дані зваженої мережі спільної появи персонажів в романі Гюго «Знедолені». D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, Addison-Wesley, Reading, MA (1993);
- *GEXF*-файл (https://gephi.org/datasets/ht2009_15min.gexf.gz). *Hypertext 2009* – динамічна мережа контактів учасників конференції *Hypertext 2009*. Source: *Sociopatterns.org*;
- *GEXF*-файл (http://www.org-soz.uni-uppertal.de/fileadmin/soziologie/org-soz/files/Heidler_et_al_2013_Relationship_patterns_in_the_19th_century.gephi). *CLASS OF 1880/81*: мережу дружби в класі німецької школи для

- хлопчиків у 1880-1881 роки. Це, ймовірно, перший набір даних із соціальних мереж, який зібрав вчитель початкової школи Йоханес Делич. Дані були проаналізовані і скопільовані для статті: *Heidler, R., Gamper, M., Herz, A., Eßer, F. (2014): Relationship patterns in the 19th century: The friendship network in a German boys' school class from 1880 to 1881 revisited. Social Networks 13: p. 1–13;*
- *GML-файл* (<https://gephi.org/datasets/karate.gml.zip>). Клуб карате Захарія: соціальна мережа дружби між 34 членами клубу карате в американському університеті у 1970-х роках. *W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977);*
 - *GML-файл* (<https://gephi.org/datasets/netscience.gml.zip>). Мережа співавторів у науці – в теорії та експериментах, зібрана М. Нейманом у травні 2006 року. *M. E. J. Newman, Phys. Rev. E 74, 036104 (2006);*
 - *GEXF-файл* (<https://gephi.org/datasets/cpan-authors.gexf.zip>). Автори *CPAN: CPAN Explorer* –це проект візуалізації, спрямований на аналіз відносин між розробниками і пакетами мови *Perl*, відомий як спільнота *CPAN*. Ця мережа була створена *Linkfluence* у липні 2009 року. Цей файл відповідає мережі розробників, які посилаються у програмах на однакові модулі мовою *Perl*;
 - *NET-файл* (<https://gephi.org/datasets/jazz.net.zip>). Мережа джазових музикантів: Мережа взаємних зв'язків джазових музикантів. *P. Gleiser and L. Danon, Adv. Complex Syst.6, 565 (2003);*
 - *TGZ-файл* (<http://franck.lumberjaph.net/graphs.tgz>). Розробники відкритого коду з *Github*: <https://github.com/franckcuny/blog/blob/master/posts/2010-03-25-github-explorer.md/>;
 - *DL-файл* (https://gephi.org/datasets/OClinks_w.dl.zip). Соціальна мережа: 1899 вузлів – *Opsahl, T., Panzarasa, P., 2009. Clustering in weighted networks. Social Networks 31 (2), 155-163;*

- *GEPHI*-файл (<https://gephi.org/datasets/hero-social-network.gephi>). Соціальна мережа супергероїв *Marvel*, створена авторами *Cesc Rosselló*, *Ricardo Alberich*, і *Joe Miro* з університету на Балеарських островах. Зібрано в *Infochimps*, перетворена *Kai Chang*;
- *GDF*-файл (<https://gephi.org/datasets/comic-hero-network.gdf.zip>). Комікси і герої мережі: ті ж дані, що і в попередній мережі, але включають комікси і героїв які входять до них;
- *DOT*-файл (<http://rankinfo.pkqs.net/twittercrawl.dot.gz>). Згадки і ретвіти деякої частини мережі *Twitter*. Файл час від часу змінюється;
- *GEXF*-файл (<http://www.sociopatterns.org/datasets/primary-school-cumulative-networks/>). Мережа контактів в початковій школі, команда *SocioPatterns*, 2011.

Біологічні мережі

- *GEXF* (<http://gephi.org/datasets/diseasome.gexf.zip>). Мережа генів розладів і хвороб, пов'язаних відомими асоціаціями генів розладів, що вказують на загальне генетичне походження багатьох захворювань. Оригінальна база даних може бути знайдена за посиланням: *The Human Disease Network*, Goh K-I, Cusick ME, Valle D, Childs B, Vidal M, Barabási A-L (2007), *Proc Natl Acad Sci USA* 104:8685-8690;
- *GEXF* (<http://gephi.org/datasets/celegans.gexf.zip>). Нейронна мережа Еледженса: спрямована зважена мережа, що представляє нейронну мережу Ц. Еледженса. Дані узагальнені Д. Ваттсом і С. Строгатцем. D. J. Watts and S. H. Strogatz, *Nature* 393, 440-442 (1998). *Original experimental data taken from J. G. White, E. Southgate, J. N. Thompson, and S. Brenner, Phil. Trans. R. Soc. London* 314, 1-340 (1986);
- *GEXF* (<https://gephi.org/datasets/yeast.gexf.zip>). *Yeast*: мережа білкової взаємодії в дріжджах;

Інфраструктурні мережі

- *GML* (<https://gephi.org/datasets/power.gml.zip>). Енергетична мережа: ненаправлена, невиважена мережа відображає топологію енергомережі західних штатів США. Дані узагальнені Д. Уоттсом і С. Строгатцем. D. J. Watts and S. H. Strogatz, *Nature* 393, 440-442 (1998);
- *NET* (<https://gephi.org/datasets/us-air97.net.zip>). *US Air97: North American Transportation Atlas Data (NORTAD)*.

Терористичні мережі

Набори даних для опису прихованих терористичних мереж були складені Центром аналізу соціальних мереж Мітчелла (*Mitchell Centre for Social Network Analysis*), Манчестерський університет (*University of Manchester*). Дані доступні для вільного завантаження за адресою:

<http://www.socialsciences.manchester.ac.uk/mitchell-centre/research/covert-networks>.

Прихована мережа за їхнім визначенням – це соціальна мережа, в якій є один або кілька елементів секретності. Члени мережі можуть спробувати зберегти свою особистість в таємниці (як у випадку з кримінальними або терористичними організаціями) або мережа може формуватися навколо дій, які повинні зберігатися в секреті з різних причин.

Перерахуємо деякі з таких мереж, представлених на наведеному ресурсі.

- Терористична мережа Малі (*Mali Terrorist Network*, <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/maliterroristnetwork>). Дані відносяться до терористичної мережі, яка діє в регіоні Сахель-Сахель-Сахара, і описують відносини між ісламістами та повстанцями-туарегами під час малійського конфлікту. Дані екстраговані з ряду газетних статей, опублікованих в період з 2010 по 2012 рік;
- Внутрішні терористичні веб-посилання (*Domestic Terrorist Web Links*, доступно <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1511999>

&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1511999). Мережа гіперпосилань між веб-сайтами місцевих терористичних груп в Сполучених Штатах. За матеріалами статті *Zhou et al. (2005), 'US domestic extremist groups on the web: link and content analysis'*;

- Лондонська банда (*London Gang*, <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/londongang>). Дані про спільне правопорушення в лондонській міській вуличній банді, що діяла у 2005-2009 роках. Дані отримані з анонімних джерел поліції про арешт і засудження "усіх підтверджених" членів банди;
- Чеська корупція (<https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/czechcorruption>). Дані надійшли з чеської бази даних ЗМІ під назвою *Newton Media Search*, що охоплює всі основні чеські газети за період з 4 червня 2013 року по 4 червня 2014 року;
- *Al Qaeda* 1993-2003 (<http://doitapps.jjay.cuny.edu/jjatt/data.php>). Серія операцій «Аль-Каїди» – це дані, які об'єднують відносини окремих осіб, пов'язаних з більш ніж 10 атаками. Це дані, які охоплюють сукупний набір атак, проведених «Аль-Каїдою» протягом десятиліття;
- Бомбардування посольства Австралії, Індонезія 2004 (http://en.wikipedia.org/wiki/2004_Australian_Embassy_bombing_in_Jakarta).
Часовий ряд, в якому розглядаються окремі атаки, еволюція відносин між людьми, які побічно і безпосередньо пов'язані з бомбардуванням посольства Австралії;
- Тимчасова Ірландська республіканська армія (*Provisional IRA, PIRA*, <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/provisionalirishrepublicanarmy>). Дані по активним тимчасовим членам *IRA* (далі – *PIRA*) в період з 1970 по 1998 рік. Дані зібрані в Міжнародному центрі по вивченню тероризму, Університет штату Пенсільванія. З цих даних формуються мережеві структури і характер

залежності всередині них. Мережа *PIRA* включає в себе наступні чотири типи відносин: (1) – спільна участь у заході *PIRA*, (2) – друзі до приєднання до руху *PIRA*, (3) - родичі і (4) – відносини у шлюбі.

5.2. Терористична мережа *Mali*

У математичному сенсі соціальна мережа – не що інше як граф з вершинами і ребрами, які, відповідно, представляють акторів (суб'єктів спілкування) і зв'язки між ними.

Розглянемо одну з мереж, представлених у попередньому пункті, а саме, *Mali Terrorist Network*.

При завантаженні цієї мережі в систему *Gephi* (з формату *CSV*), відображається інформація про кількість вузлів (акторів) і зв'язків між ними (рис. 54).

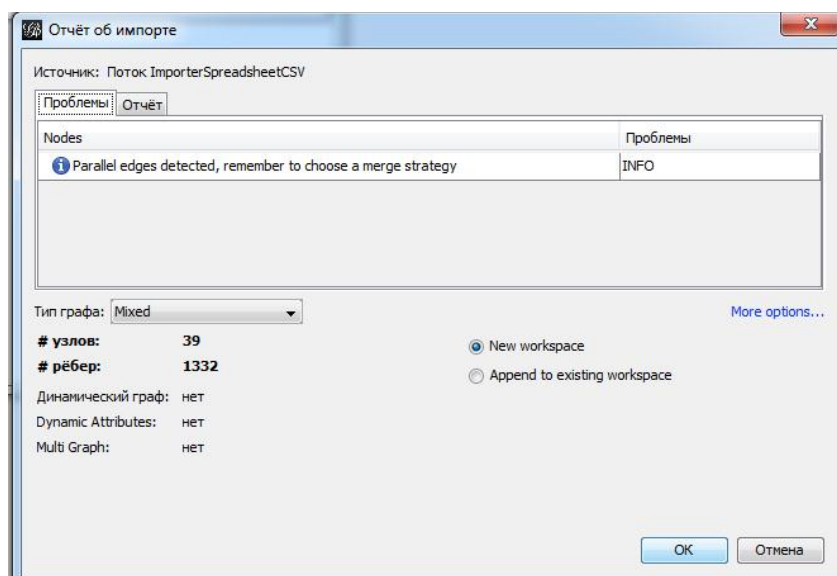


Рис. 54 – Інтерфейс завантаження мережевих даних

Після цього відкривається файл, який в режимі «Обробка» відображається як мережа в неврегульованому вигляді (рис. 55).

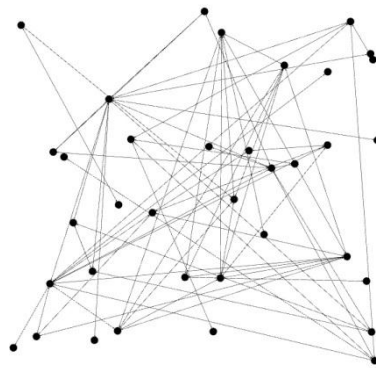


Рис. 55 – Первинне відображення мережі

Далі мережа обробляється описаними вище способами – укладання за алгоритмом *OpenOrd*, розмір вузлів – за ступенем, розфарбування – по класу модульності. У режимі «Обробка» виходить наступна візуалізація (рис. 56).

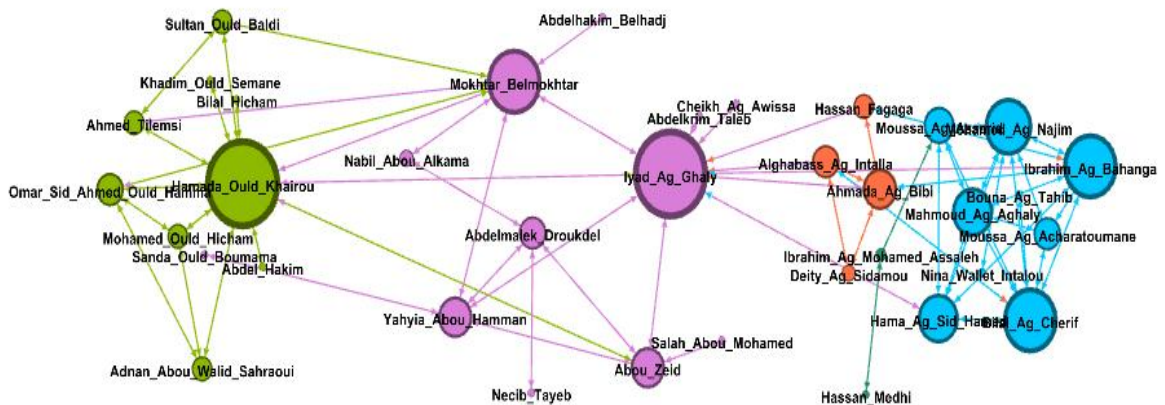


Рис. 56 – Візуалізація терористичної мережі в режимі «Обробка»

Для усунення зайвої деталізації в режимі «Лабораторія даних» створюється копія стовпчика «Мітки», потім стовець «Мітки» очищається і туди копіюються мітки вузлів – центри кластерів, причому велика частина з них з найвищими значеннями *PageRank* (рис. 57). Потім мережа відображається в режимі «Перегляд», формується остаточна візуалізація (рис. 58).

Таблица Данных					
Узлы Ребра					
Конфигурация					
Добавить узел					
Добавить ребро					
Поиск/Замена					
Импортировать из CSV					
Экспортировать таблицу					
Другие действия					
Id	Label	Interval	PageRank	Modularity Class	Label_copy
Hamada_Ould_Khairou	Hamada_Ould_Khairou		0.084053	0	Hamada_Ould_Khairou
Iyad_Ag_Ghaly	Iyad_Ag_Ghaly		0.076327	1	Iyad_Ag_Ghaly
Mokhtar_Belmokhtar	Mokhtar_Belmokhtar		0.060029	1	Mokhtar_Belmokhtar
Bilal_Ag_Cherif	Bilal_Ag_Cherif		0.052431	3	Bilal_Ag_Cherif
Ibrahim_Ag_Bahanga	Ibrahim_Ag_Bahanga		0.049476	3	Ibrahim_Ag_Bahanga
Mohamed_Ag_Najim			0.046429	3	Mohamed_Ag_Najim
Abou_Zeid			0.040313	1	Abou_Zeid
Yahya_Abou_Hamman			0.040201	1	Yahya_Abou_Hamman
Hama_Ag_Sid_Hamed			0.038626	3	Hama_Ag_Sid_Hamed
Mahmoud_Ag_Aghaly			0.038308	3	Mahmoud_Ag_Aghaly
Abdelmalek_Droukdel			0.035489	1	Abdelmalek_Droukdel
Ahmada_Ag_Bibi	Ahmada_Ag_Bibi		0.03407	2	Ahmada_Ag_Bibi
Moussa_Ag_Assarid			0.031405	3	Moussa_Ag_Assarid
Omar_Sid_Ahmed_Ould_Hamma			0.030624	0	Omar_Sid_Ahmed_Ould_Hamma
Alghabass_Ag_Intalla			0.028203	2	Alghabass_Ag_Intalla
Moussa_Ag_Acharatoumane			0.026056	3	Moussa_Ag_Acharatoumane
Adnan_Abou_Walid_Sahraoui			0.023961	0	Adnan_Abou_Walid_Sahraoui
Mohamed_Ould_Hicham			0.023961	0	Mohamed_Ould_Hicham
Ahmed_Tilemsi			0.023783	0	Ahmed_Tilemsi
Sultan_Ould_Baldi			0.023783	0	Sultan_Ould_Baldi
Hassan_Fagaga			0.021111	2	Hassan_Fagaga
Nabil_Abou_Alkama			0.018097	1	Nabil_Abou_Alkama
Deity_Ag_Sidamou			0.015949	2	Deity_Ag_Sidamou
Ibrahim_Ag_Mohamed_Assaleh			0.012072	4	Ibrahim_Ag_Mohamed_Assaleh
Necib_Tayeb			0.011716	1	Necib_Tayeb
Salah_Abou_Mohamed			0.011019	1	Salah_Abou_Mohamed
Sanda_Ould_Boumama			0.010999	1	Sanda_Ould_Boumama
Abdel_Hakim			0.010663	0	Abdel_Hakim
Bilal_Hicham			0.010663	0	Bilal_Hicham
Khadim_Ould_Semane			0.010663	0	Khadim_Ould_Semane
Abdelhakim_Belhadj			0.010548	1	Abdelhakim_Belhadj
Abdelkrim_Taleb			0.010065	1	Abdelkrim_Taleb
Cheikh_Ag_Awissa			0.010065	1	Cheikh_Ag_Awissa
Nina_Wallet_Intalou			0.009804	3	Nina_Wallet_Intalou
Bouna_Ag_Tahib			0.009737	3	Bouna_Ag_Tahib
Hassan_Medhi			0.009301	4	Hassan_Medhi

Рис. 57 – Режим «Лабораторія даних»

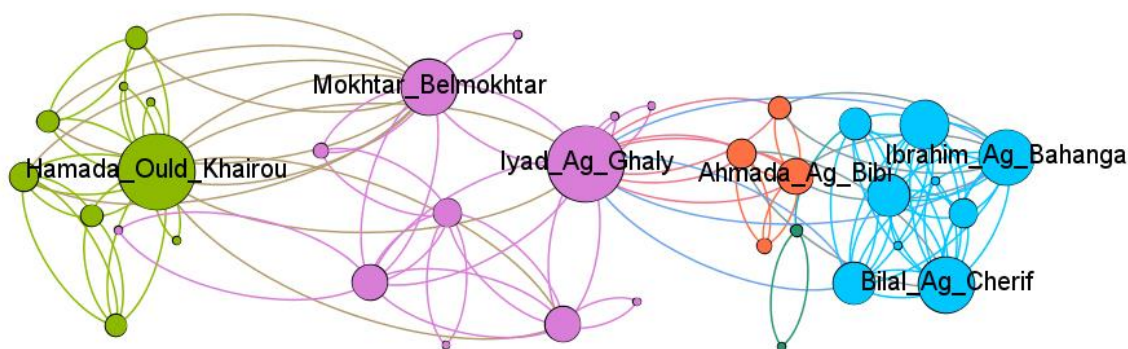


Рис. 58 – Візуалізація терористичної мережі

5.3. Мережа понять, пов'язаних з початковим, по *Wikipedia*

Мережева енциклопедія *Wikipedia* складається зі статей, пов'язаних один з одним гіперпосиланнями. Система *Wikipedia* доступна в мережі Інтернет і не передбачає підписки, крім того, доступна для завантаження в повному обсязі. Досліджуючи фрагмент цієї мережі, пов'язаний з певним поняттям можна побудувати мережу, що охоплює більшу частину предметної

області. В якості такого поняття можна вказати ім'я вченого і побудувати мережу пов'язаних з ним понять.

Для первинного доступу до системи був застосований термін, за яким існує відповідна стаття в *Wikipedia*. Створювана стаття і редагована експертами-авторами. Вочевидь, що побудована шляхом пересування по гіперпосиланням у рамках *Wikipedia* мережа понять може мати досить великі розміри, якщо її не обмежувати певною тематикою відповідної предметної області – вільний перехід за гіперпосиланнями веде до ефекту так званого "зсуву тематик" (*Topic Drift*). Для подолання цього ефекту застосовується елементарна тематична фільтрація – для аналізу використовуються тільки ті статті з *Wikipedia*, які містять базовий термін, що визначається експертом-аналітиком. Розпізнавання кластерів у таких мережах може розглядатися як основа для виявлення окремих наукових напрямків.

Розглядався наступний алгоритм побудови моделей предметних областей за даними сервісу *Wikipedia*, який припускає подолання ефекту *Topic Drift*:

1. Обирається перший термін-поняття, з якого починається зондування *Wikipedia*.
2. Відкривається сторінка веб-сервісу (стаття *Wikipedia*), яка відповідає обраному терміну-поняттю. До створюваної мережі додаються всі терміни-поняття, які відповідають гіперпосиланнями на обраній сторінці. Формуються ребра-зв'язки до цих вузлів з початкового вузла.
3. Статті, що відповідають гіперпосиланням на попередній сторінці, визначаються як базові, якщо на них міститься гіперпосилання на статтю, відповідну першому терміну-поняттю, з якого починалося зондування.
4. Зі списку вузлів формованої мережі визначається той, за яким ще не здійснювалося переходу та на сторінку якого планується перейти для

подальшого аналізу. Цей вузол повинен відповідати вимозі, наведеній у попередньому пункті, і входити до складу тих вузлів, до сторінок яких вже був здійснений перехід.

5. Якщо такий базовий вузол обраний, то здійснюється перехід до пункту 2.

6. Якщо такого вузла не існує, то вважається, що мережа, яка відповідає моделі предметної області, побудована.

Відповідно до наведеного алгоритму, процес збору інформації з *Wikipedia*, починаючи з певного вузла-поняття, припиняється, коли вже неможливий перехід до нового вузлу (базових вузлів для переходів вже не залишилося), тобто "зациклювання" неможливо.

Для побудови мережі понять був використаний спеціальний інтернет-робот, результат роботи якого – набір даних у форматі *CSV*. Потім у програмі *Gephi* відкривається цей *CSV*-файл, який обробляється описаними вище способами (рис. 59). Особливості – укладка за алгоритмом *Yifan Hu*, розмір вузлів – по ступеню, розфарбування – по класу модулярності.

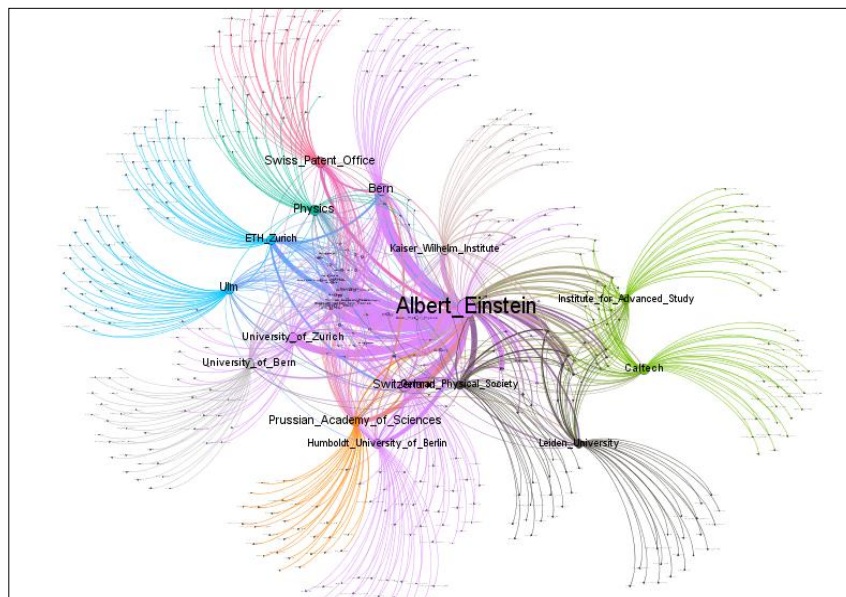


Рис. 59 – Мережа понять, що відповідає поняттю «*Albert Einstein*»

5.4. Мережа співавторства вчених

Для цілого ряду завдань, пов'язаних з управлінням наукою, необхідно використати мережу співавторства учених. До таких завдань відносяться формування наукових колективів, експертних груп, розрахунків рейтингів окремих вчених та ін. Для побудови такої мережі був використаний спеціальний інтернет-робот, що сканує мережеву наукометричну базу даних *Google Scholar*, працюючий за наступним алгоритмом:

1. Обирається перший автор (вузол), зі сторінки якого в *Google Scholar* починається зондування мережі.
2. Експертним шляхом визначається невеликий перелік базових тегів, що відповідають найважливішим поняттям, які відповідають автору, наприклад, теги *computer*, *cyber security*, *information security*.
3. Відкривається сторінка веб-сервісу *Google Scholar*, який відповідає обраному автору. До створюваної мережі додаються всі співавтори, що містяться на сторінці обраного учасника. Формуються ребра-зв'язки до цих вузлів (співавторів) з початкового вузла (автора).
4. З сформованого списку вузлів мережі випадковим чином вибирається той, на сторінку якого планується перейти для подальшого аналізу. Цей вузол також повинен задовільняти тематиці обраної предметної області – його теги входять до складу дескрипторів, визначених на 2 кроці, і не входять до складу тих вузлів, до сторінок яких вже був здійснений перехід.
5. Якщо такий вузол-автор обраний, то здійснюється перехід до кроку 3.
6. Якщо такого вузла не існує, то вважається, що мережа співавторів, побудована.

Відповідно до цього алгоритму процес зондування мережі, починаючи з певного вузла, припиняється при "зацикленні", тобто коли згідно з

алгоритмом повинен відбуватися перехід до вже пройденого вузла, а також при відхиленні сусідніх вузлів від основної тематики (що визначається з урахуванням лексичного складу тегів).

Результат роботи програми сканування *Google Scholar* – набір даних у вже поданому форматі *CSV*. Потім в програмі *Gephi* відкривається цей *CSV*-файл, який обробляється описаними вище способами (рис. 60). Особливості – укладка за алгоритмом *Yifan Hu*, розмір вузлів – по *PageRank*, розфарбування – по класу модулярності.

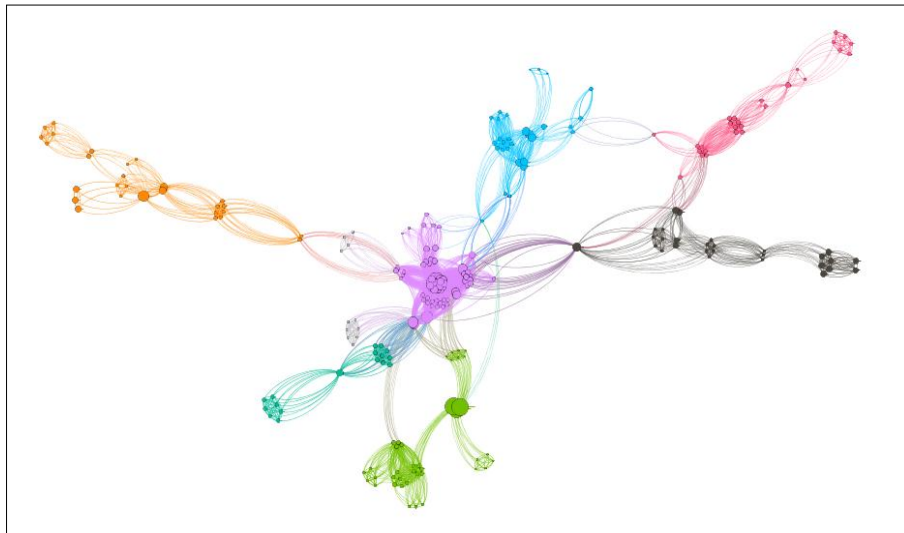


Рис. 60 – Мережа співавторства вчених, розділена на кластери

Ще одна мережа співавторства (в науці про мережі, 1589 співавторів, +2742 зв'язки), зібрана М. Нейманом у травні 2006 року (<https://gephi.org/datasets/netscience.gml.zip>) після укладки і кластеризації приймає вид, представлений на рис. 61.

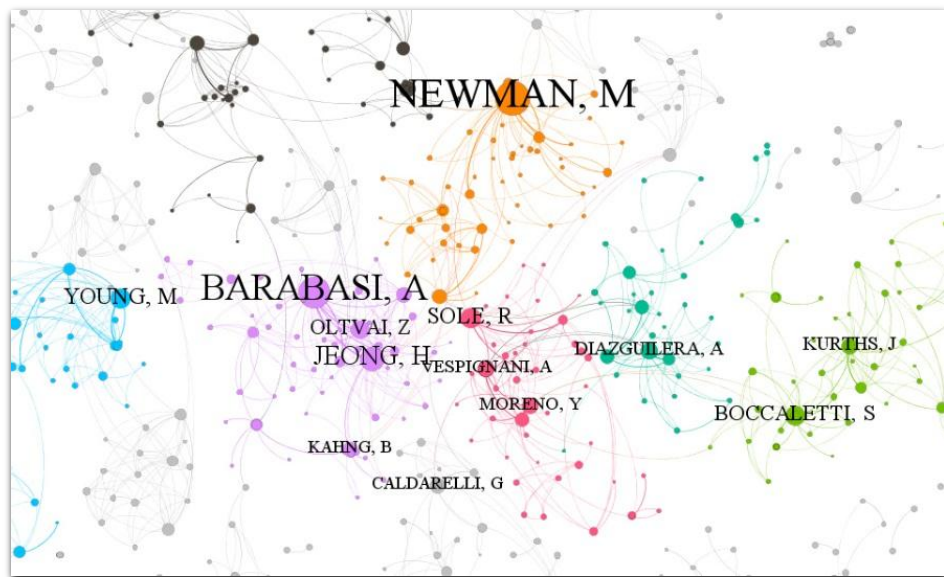


Рис. 61. Мережа співавторства в науці про мережі (укладка *Yifan Hu*)

Питання для самоперевірки

1. Наведіть класифікацію відомих наборів даних (*Datasets*) для дослідження їх у програмі *Gephi*. Охарактеризуйте їх.
2. Поясніть методику обробки терористичної мережі *Mali* в середовищі *Gephi*.
3. Наведіть алгоритм побудови моделей предметних областей за даними сервісу *Wikipedia*.
4. Наведіть алгоритм побудови мережі співавторства вчених.

Література

1. *Albert-László Barabási*. Network Science. – Cambridge University Press, 2016.
2. *Amy N. Langville, Carl D. Meyer*. Google's PageRank and beyond: the science of search engine rankings. – Princeton University Press, 2006. ISBN: 0691122024, 9780691122021.
3. *Mark Newman, Albert-László Barabási, Duncan J. Watts*. The Structure and Dynamics of Networks (Princeton Studies in Complexity). – Princeton University Press, 2006. ISBN: 0691113572, 9780691113579.
4. *Diogo R. Ferreira*. «A Primer on Process Mining: Practical Skills with Python and Graphviz». – Springer International Publishing, 2017. – 101 P. ISBN: 978-3-319-56426-5, 978-3-319-56427-2.
5. Emden R. Gansner and Eleftherios Koutsoos and Stephen North. «Drawing Graphs with dot». Dot User's Manual. – AT&T Bell Labs, January 5, 2015.
6. *Social Networks Visualizer*. Software Requirements Specification Version 1.0. – Faculty of Natural Sciences Aristotle University of Thessaloniki, 2012.
7. *Devangana Khokhar*. Gephi Cookbook. – Packt Publishing, 2015. ISBN 78-1-78398-740-5.
8. *Ken Cherven*. Mastering Gephi Network Visualization. – Packt Publishing, 2015. ISBN 78-1-78398-734-4.
9. *Ken Cherven*. Network Graph Analysis and Visualization with Gephi. – Packt Publishing, 2013. ISBN 78-1-78328-013-1.
10. *John W. Foreman*. *Data Smart*. Using Data Science to Transform Information into Insight. – Wiley, 2013. ISBN 111-8-66146-X, 978-1-11866-146-8.
11. *Ulrik Brandes, Markus Eiglsperger, Jürgen Lerner*. GraphML Primer. (<http://graphml.graphdrawing.org/primer/graphml-primer.html>). – 2015.
12. *Ландэ Д.В., Снарский А.А., Безсуднов И.В.* Интернетика: Навигация в сложных сетях: модели и алгоритмы. – М.: Либроком (Editorial URSS), 2009. – 264 с. ISBN 978-5-397- 00497-8.

13. *Снарский А.А., Ландэ Д.В.* Моделирование сложных сетей: учебное пособие. – К.: Инжиниринг, 2015. – 212 с. ISBN 978-966-2344-44-8.
14. *Ланде Д.В., Субач І.Ю., Бояринова Ю.Є.* Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навчальний посібник. – К.: ІСЗЗІ КПІ ім. Ігоря Сікорського, 2018. – 300 с. ISBN 978-966-2577-12-9.

ПРЕДМЕТНИЙ ПОКАЖЧИК

Алгоритм	17
PageRank	18
HITS	17
Дейкстри	23
Флойда-Уоршелла	23
Eades Spring Embedder	31
Fruchterman-Reingold	31
Yifan-Hu	55
Force-directed	55
OpenOrd	55
Force Atlas	58
Коефіцієнт	9
Глобальної ефективності	9
Кластерності	10
Компановка	55
Модель	15
Ердеша-Реньї	15
Барабаші-Альберт	16
Уоттса-Строгатца	14
Мережа	68
Mali	70
Wikipedia	63
Web i Internet	63
Соціальна	64
Біологічна	64
Інфраструктурна	66
Терористична	66
Ранжування	59
Укладка	56
Фільтрація	61

Навчальне видання

Ланде Дмитро Володимирович
Субач Ігор Юрійович

**ВІЗУАЛІЗАЦІЯ ТА АНАЛІЗ
МЕРЕЖЕВИХ СТРУКТУР**

Навчальний посібник

В авторській редакції

Інститут спеціального зв'язку та захисту інформації
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»

Підписано до друку 29.04.2020. Формат 60×84¹/₁₆. Папір офіс. Гарнітура Times.

ІСЗЗІ КПІ ім. Ігоря Сікорського,
м. Київ вул. Верхньоключова, 4, тел. 204-91-51