

## МЕТОД ЗОБРАЖЕННЯ ЗВ'ЯЗКІВ МІЖ ВЕРШИНАМИ ГРАФА

*В статті проаналізовано проблеми, які виникають в процесі моделювання зв'язків між вершинами графа та показано необхідність створення нового методу відтворення. Відповідно до сформованих критеріїв візуалізації запропоновано новий метод проведення зв'язків між вершинами графа та наведений приклад використання в процесі візуалізації.*

### Вступ

Часто для опису деякої ситуації виникає необхідність у рисуванні точок, що зображають людей, населені пункти, хімічні речовини, і з'єднанні їх лініями або стрілками, що означають деякі відношення між ними. Такі схеми зустрічаються в різних сферах: психології (соціограми), топології (симплекси), фізиці (електричні ланцюги), економіці (організаційні діаграми) і т.д. У науці такі рисунки одержали назву графів [1-3].

Графи використовуються для опису різних об'єктів і процесів як з метою візуалізації, так і для створення аналітичних і алгоритмічних методів їх дослідження. Висока наочність дозволяє полегшити формулювання досить складних завдань і зробити доступними методи для їх вирішення. Графи й більш складні графоподібні об'єкти є зручним способом подання будь-якої структурованої інформації [1,4].

### Постановка задачі

Незважаючи на широке використання розглядуваної структури даних, на сьогодні не існує ефективного методу, який би вирішував проблему візуалізації графів, поданих у вигляді матриць [5,6]. Задача візуалізації графів полягає в такому їх відтворенні, при якому отримані зображення були б простими для споглядання та відповідали сформованим критеріям відтворення [6,7]:

- ✓ **Рівномірне розташування вершин на екрані** – забезпечується шляхом розміщення однакової кількості вершин по ярусах графа. Розміщення відбувається на однаковій відстані між вершинами, що сприятиме утворенню рівномірного рисунка на екрані монітора.
- ✓ **Мінімальна кількість перетинань дуг між собою** – забезпечується шляхом розміщення вершин, що містять спільні зв'язки на одному або сусідніх ярусах. Тобто шукається таке гомоморфне зображення, яке б забезпечувало даний критерій.
- ✓ **Неприпустимість перетину вершин графа** – забезпечується шляхом знаходження „точки обходу”, через яку здійснюється проведення променя від початкової до кінцевої вершини.
- ✓ **Мінімальна площа, яку займає рисунок** – при забезпеченні наведених трьох критеріїв створюється ситуація, при якій площа, на якій розміщується рисунок, є мінімальною, оскільки всі вершини, що мають спільні зв'язки, розміщуються поруч. Крім того, на цьому кроці роботи здійснюється операція „зменшення площі”, що полягає в переставленні місцями груп вершин, з метою зменшення довжин дуг, що їх з'єднують. Це дає змогу зменшити міжвершинні віддалі, а відтак і площу, яку займає рисунок.

Особливістю описаної задачі є те, що в комп'ютері граф подається у вигляді матриць, при цьому зберігається тільки структура зв'язків, а дані про взаємне розміщення елементів відсутні, що є недоліком. Адже, якщо провести зв'язки між вершинами графа абсолютно довільно, то можна домогтися ще меншої наочності, ніж у випадку формального подання.

Задача візуалізації графів складається з двох етапів: розміщення вершин на площині та проведення зв'язків між ними. Перший етап вдало реалізовано у роботі [8], щодо другого - то для зазначеного методу його не досліджено. З огляду на те, стаття присвячена вирішенню проблеми зображення зв'язків між вершинами графів, поданих з допомогою матриць суміжності.

**Основними завданнями дослідження є:** розробка методу зображення зв'язків між вершинами графа та формулювання висновків про доцільність його застосування в процесі візуалізації.

**Метод проведення зв'язків між вершинами графа**

Проблема рисування зв'язків між вершинами графа розглядається у двох аспектах: рисування зв'язків між „сусідніми елементами” та елементами, між якими є ряд перепон, – „віддаленими елементами” (рис.1).

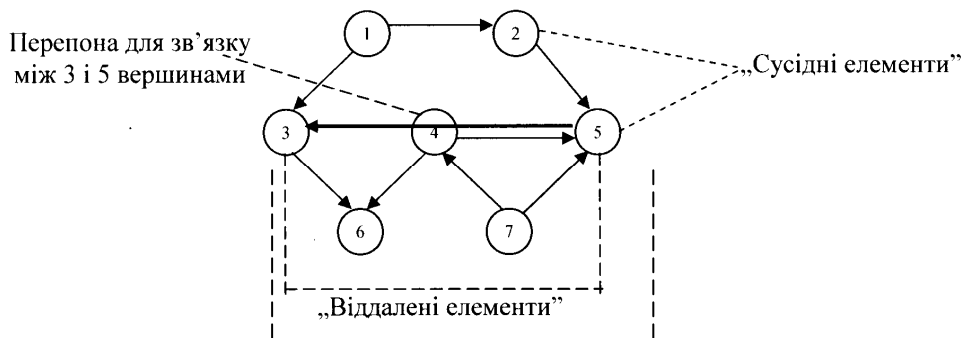


Рис.1. Приклад графа

Щодо **першого аспекту**, то складність його реалізації полягає у виборі виду дуги, з допомогою якої б забезпечувалась мінімальна довжина міжвершинних віддалей, та водночас зручне сприйняття та оцінювання відповідно до визначених критеріїв. Як відомо, найкоротший шлях між двома елементами є променем, що їх з'єднує. Відповідно до цієї особливості, пропонується використовувати в якості зв'язків між вершинами промені з координатами початку і кінця, що визначаються відповідно до формули (1) [8]:

$$\begin{aligned} X &= L_{\Sigma 1} + 3d * I_p \\ Y &= L_{\Sigma 2} + 3d * I_c \end{aligned} \tag{1}$$

де X, Y – шукані координати;

$L_{\Sigma 1}$ ,  $L_{\Sigma 2}$  – показники, які враховують розміри вікон прикладної програми та операційної системи;

$I_p$ ,  $I_c$  – коефіцієнти, які вказують місце розташування відповідної вершини (рядок та стовпець);

d – діаметр вершини.

Щодо зв'язаності вершин, то на них вказуватиме матриця суміжності, що вводиться на початку роботи і зберігається в пам'яті комп'ютера як масив однорозрядних цілих чисел [4,9].

Щодо **другого аспекту**, то проблема полягає в тому, що неможливо провести зв'язок у вигляді прямої між „віддаленими елементами”, оскільки він перетне площини інших вершин. Зокрема, зв'язок, що сполучає дві вершини (п'яту з третьою), проведений з допомогою прямої лінії, перетинає площину вершини за номером чотири. Така ситуація є неприйнятною, оскільки суперечить сформованим критеріям відображення [7]. У цьому випадку пропонується здійснювати „обхід” вершини графа.

На основі проведеного аналізу визначено, що оптимальним варіантом для вирішення поставленої задачі є алгоритм Лі (хвильовий алгоритм). Суть алгоритму полягає у проходженні з початкової вершини A всіх суміжних з нею клітинок. Якщо суміжної клітинки не виявлено, то проводиться повернення до тієї, з якої потрапили в

поточну, та робиться наступна спроба. В разі досягнення кінцевої вершини **B** здійснюється друга фаза роботи алгоритму - прокладання шляху між кінцевими вершинами в зворотному напрямку – від точки **B** до точки **A**. На цьому шляху кожен раз вибирається клітинка, яка розміщується найближче до точки **A**. Послідовність вибраних клітинок становить шуканий шлях. Початковими умовами, при яких розглядається функціонування алгоритму, є поле  $P(M \times N)$ , де  $M$  і  $N$ , відповідно, розміри по вертикалі та горизонталі, кожна з кліток якого може мати одну з двох властивостей - прохідність або непрохідність [10].

Проілюструємо роботу алгоритму Лі для графа, що наведений на рис.1, скориставшись для цього програмою **GWave 4.5** (рис.2,3) [11].

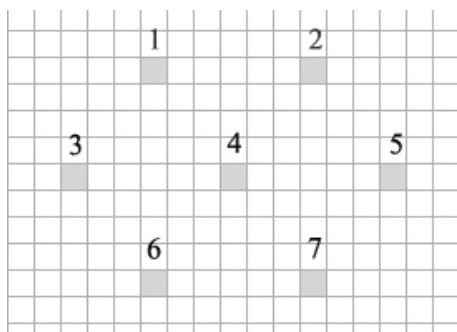


Рис. 2. Рисунок перед застосуванням алгоритму Лі

Після закінчення роботи програми на екрані монітора утвориться рисунок:

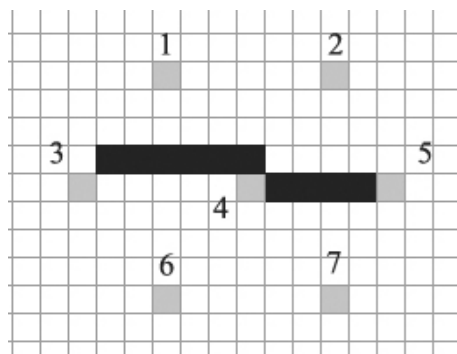


Рис. 3. Рисунок після застосування алгоритму Лі

Як можна зауважити, програма, побудована на основі алгоритму Лі, відносно легко може знайти шлях між двома вершинами. Однак отриманий шлях є ламаною лінією незрозумілої геометрії і його використання значно погіршує сприйняття рисунка. У випадку існування декількох шляхів, оцінювання сформованого зображення значно ускладниться. З огляду на те, пропонується здійснити модифікацію поданого алгоритму, а саме внести зміни у фазу проведення зв'язків між вершинами. Тобто під час проходження другої фази, коли в пам'яті комп'ютера зберігається масив чисел для проведення зв'язку, шукається точка (С), що найбільше відхиляється від траєкторії шляху. Далі відбувається запам'ятовування її координат і проведення через неї променя від початкової вершини в кінцеву. Таким чином створюється обхід перепон. Отриманий малюнок поданий на рис.4.

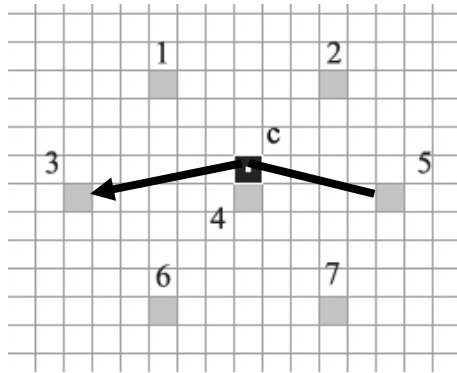


Рис. 4. Рисунок після модифікації алгоритму Лі

Для отримання „шляху обходу” необхідно внести зміни в другу фазу роботи алгоритму, оскільки на даному етапі відбувається рисування ламаного шляху [10]. Враховуючи специфіку проведення шляху в алгоритмі Лі (рух здійснюється в горизонтальному та вертикальному напрямках) та методику розміщення вершин [8], знайдений шлях буде формувати лінію з одним зламом. Пошук точки обходу здійснюється шляхом відслідкування зміни значення координати абсцис (ординат) при сталому значенні координати ординат (абсцис). Як тільки розпочинається зміна значення сталої координати, це і буде шукана „точка обходу”.

Виходячи з описаної процедури, для знаходження „точки обходу” потрібно безперервно перевіряти умову на змінність координат абсцис та ординат. Наведемо методику знаходження „точок обходу” для різних розміщень вершин (рис.5,6).

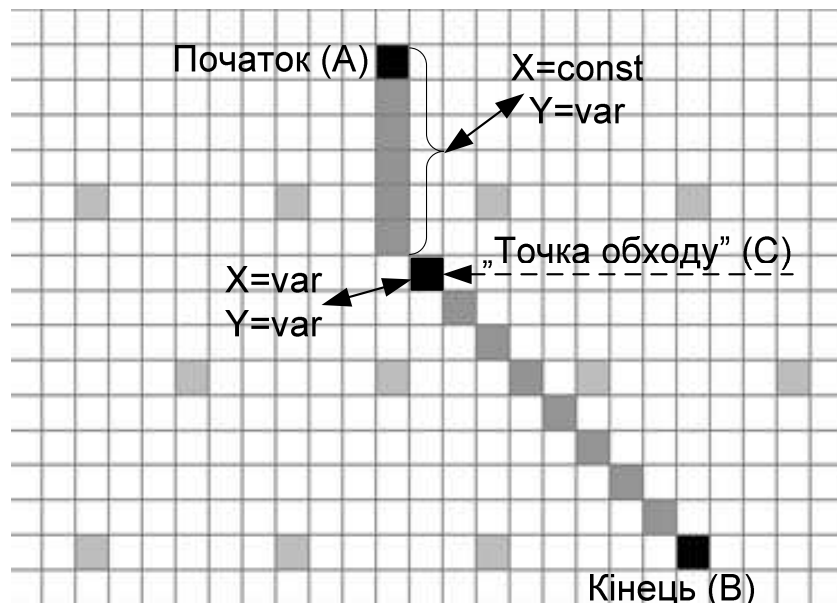


Рис. 5. Знаходження „точки обходу” при зміні координати у

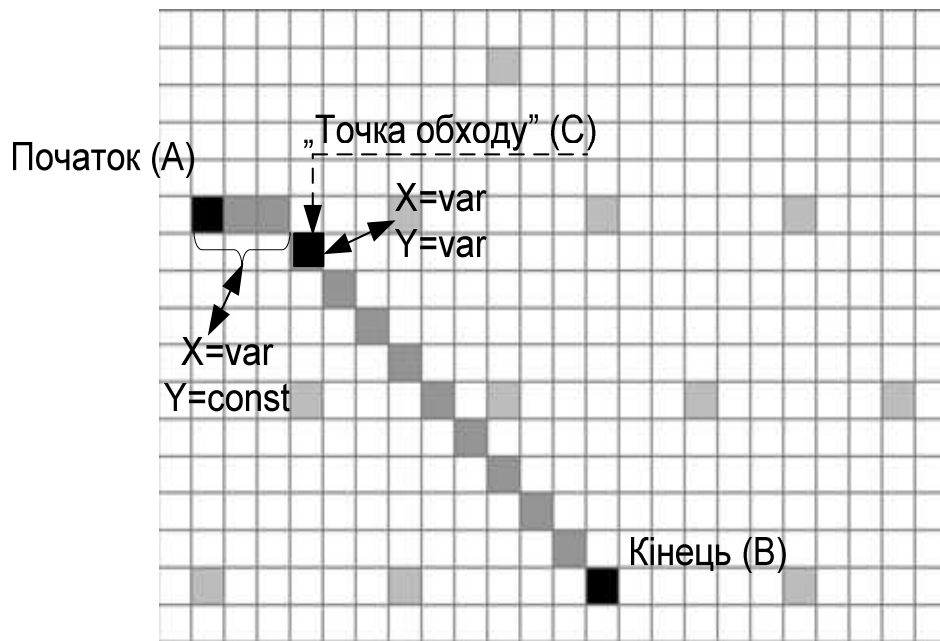


Рис. 6. Знаходження „точки обходу” при зміні координати x

Після знайдення „точки обходу” відбувається формування остаточного зв’язку. На рисунку 7 наведена блок-схема методу рисування зв’язків між вершинами графа.

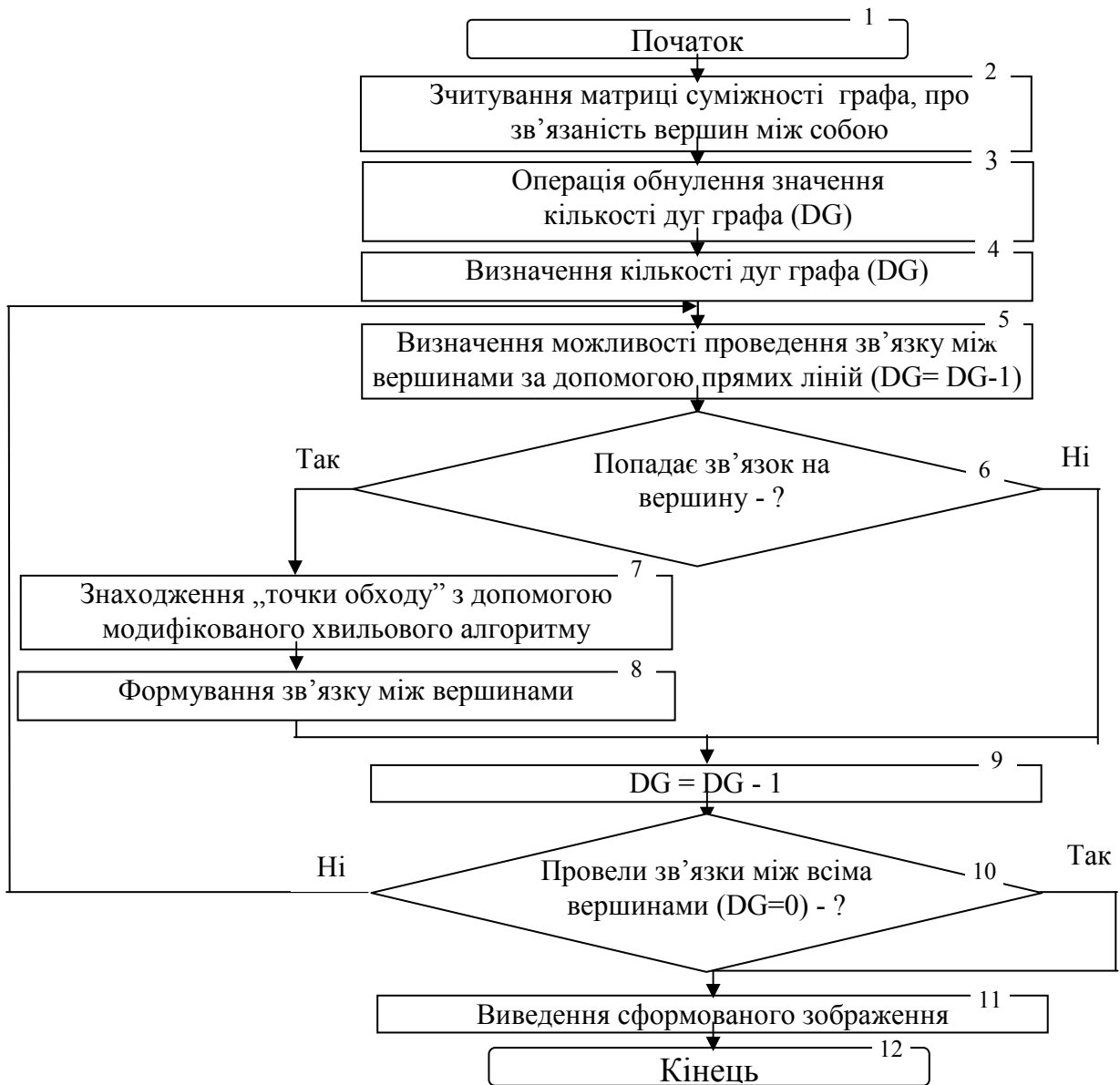


Рис.7. Блок-схема методу рисування зв'язків

Дотримуючись наведеної блок-схеми, граф, зображений на рис. 1, набуде вигляду, поданого на рис.8.

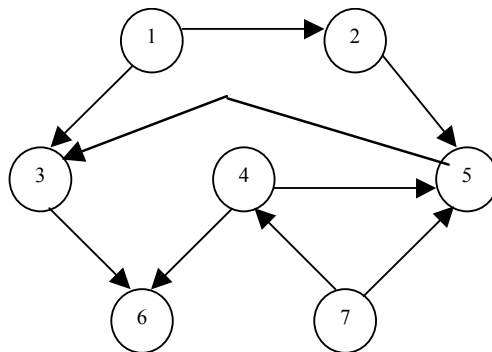


Рис.8. Перерисований граф відповідно до описаного методу

Можна зауважити, що одержаний рисунок відповідає сформованим критеріям відображення, а отже, є оптимальним для сприйняття та оцінювання [4,7].

### **Висновки**

У статті запропоновано метод зображення зв'язків між вершинами графа, в якому передбачено варіанти проведення як звичайних зв'язків у вигляді променя, так і ламаних, які формуються у відповідності до знайдення „точки обходу”. Наведено приклад використання описаного методу та показана ефективність його застосування відповідно до визначених критеріїв відображення. З допомогою описаного методу можна провести зв'язки між вершинами графа, що дозволяє адекватно оцінювати подану ним структуру та здійснювати відповідні перетворення над нею.

*In article the analyzed problems which arise during modeling connections between tops the graphs and the shown necessity of new method creation of reproduction. According to the generated criteria of visualization the new method of carrying out connections between tops the graphs is offered and the example of use is resulted during visualization.*

### **Література**

1. Di Battista G., Eades P., Tamassia R., Tollis I.G. Graph Drawing: Algorithms for Visualization of Graphs. – Prentice Hall, 1999. – P. 898
2. Дистель Р. Теория графов. - Новосибирск: Изд-во Института математики, 2002. - 336с.
3. Басюк Т.М. Аналіз та класифікація методів візуалізації // Поліграфія і видавнича справа. –2003. – Вип. 40. – С. 109-114.
4. Дунець Р.Б., Басюк Т.М. Основні задачі візуалізації графів, що описують топології поліграфічних систем // Наукові записки УАД. 2002. Вип. 5. С.93 – 96.
5. Дунець Р.Б. Аналіз та синтез топологій комп'ютерних видавничо-поліграфічних систем: НВФ „Українські технології”. Львів – 2003. -192с.
6. Касьянов В.Н., Евстегнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ, 2003. – 1104 с.
7. Басюк Т.М. Критерії відображення графів в процесі візуалізації. // Наукові записки УАД. – 2004.-Вип. 7. С.60 – 63.
8. Басюк Т.М. Метод розміщення вершин графа в процесі візуалізації. // Вісник НУ „Львівська політехніка”. – 2004.- Вип. 145. С.124 – 132.
9. Хорн Р., Джонсон Ч. Матричный анализ. — М.: Мир, 1989. — 655 с.
10. Нечепуренко М.И., Попков В.К., Майнагашев С.М. и др. Алгоритмы и программы решения задач на графах и сетях. - Новосибирск, Наука (сибирское отделение), 1990. - 515 с.
11. Battista G., Tamassia R. GWave – Instrument of Waves Drawing: An Annotated Bibliography// Computational Geometry: Theory and Applications. - 2002. - №8. P. 285-293.

*Одержано 02.09.2004 р.*