

МЕТОД РОЗМІЩЕННЯ ВЕРШИН ГРАФА В ПРОЦЕСІ ВІЗУАЛІЗАЦІЇ

© Басюк Т.М., 2004

Запропоновано метод розміщення елементів графа на екрані монітора та наведено методику його використання в процесі візуалізації.

In the article the method of accommodation of elements the graphs on the screen of the monitor is offered and the technique of its use is resulted during visualization.

Постановка проблеми у загальному вигляді

Одним з найважливіших етапів дослідження будь-яких технологічних ліній і систем керування є візуалізація їх структури з метою адекватної оцінки, опрацювання та подальшого використання. Для цього їх зображають у вигляді рисунка, що складається з вершин, які становлять основні елементи ліній та зв'язків, що з'єднують визначені пари цих вершин [1]. Такі рисунки відомі під загальною назвою графів. Графи зустрічаються в багатьох галузях під різними назвами: "структури" у цивільному будівництві, "мережі" в електротехніці, "соціограми" у соціології й економіці, "молекулярні структури" у хімії [2, 3].

Завдяки своєму широкому застосуванню теорія графів в останні роки інтенсивно розвивається. Великою мірою цьому сприяє прогрес у галузі розвитку швидкодіючих обчислювальних машин. Оскільки безпосереднє і детальне подання реальних систем, таких, як розподільні мережі, системи зв'язку та технологічні лінії, приводить до утворення графів великих розмірів, успішний аналіз їх залежить однаковою мірою як від існування "добрих" алгоритмів, так і від можливості використання швидкодіючих обчислювальних машин [4].

Задача візуалізації графів полягає в такому їх відтворенні, при якому отримані зображення були б простими для сприйняття та відповідали заданим критеріям відтворення [5]:

Рівномірне розташування вершин на екрані забезпечується розміщенням однакової кількості вершин на ярусах графа на однаковій відстані між вершинами, що сприятиме утворенню рівномірного рисунка на екрані монітора.

Мінімальна кількість перетинань дуг між собою забезпечується розміщенням вершин, що мають спільні зв'язки на одному або сусідніх ярусах. Тобто шукається таке гомоморфне зображення, яке б відповідало цьому критерію.

Неприпустимість перетину вершин графа забезпечується знаходженням „точки обходу”, через яку проводять промінь від початкової до кінцевої вершини.

Мінімальна площа, яку займає рисунок. Якщо забезпечено умови наведених трьох критеріїв, створюється така ситуація, коли площа, яку займає 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.

Аналіз останніх досліджень

Сьогодні відома велика кількість методів візуалізації, однак більшість з них орієнтовані на відтворення графів окремих типів (ієрархічних, планарних, деревоподібних), при цьому передбачається, що наперед відома їх належність до того чи іншого типу [6,7].

Особливістю наведеної задачі є те, що в комп'ютері граф подається у вигляді матриць, при цьому зберігається тільки структура зв'язків, а даних про належність його до певного типу та про взаємне розміщення елементів немає, що є недоліком [4,8,9]. Адже, якщо розмістити елементи графа абсолютно довільно і провести зв'язки між ними, то це призведе до ще меншої наочності, ніж у випадку формульного подання. Неможливість застосування існуючих методів пояснюється і тим, що жоден з них не забезпечує визначені критерії відображення [5].

А саме, у алгоритмах формування планарних графів не забезпечується критерій рівномірності розміщення вершин графа, ієрархічних графів – не передбачається мінімізація міжреберних перетинань, а у випадку деревоподібних графів – відбувається недотримання критеріїв – мінімальної площі рисунку та його рівномірності. Різновиди наведених методів можна хіба що використати на етапі редагування як один з варіантів для покращання вигляду рисунку [10].

З огляду на це актуальною є задача розробки нового методу візуалізації графів, який би забезпечував сформовані критерії відтворення [1].

Цілі статті – розробка методу оптимального розміщення вершин графа на екрані комп'ютера та формування висновків про доцільність його застосування в процесі візуалізації.

Основний матеріал

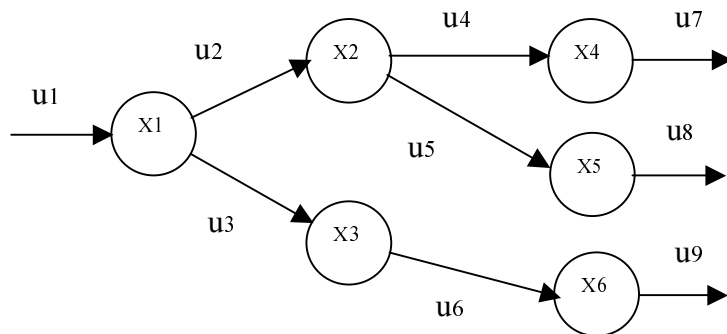
Задача візуалізації графів складається з двох етапів: розміщення елементів на площині та проведення зв'язків між ними. Кожен з цих етапів є важливим, оскільки безпосередньо впливає на вигляд майбутнього рисунка [6,11,12].

У статті запропоновано ярусний метод розміщення вершин графа за принципом: донизу → ліворуч (праворуч). З його допомогою створюється граф деревоподібної форми з відповідними перевагами цього подання [13]:

- кількість рівнів (ярусів) графа не обмежена. Ця ознака відповідає потребам задач, пов'язаних зі складними системами, оскільки дозволяє сформувати велику кількість ярусів;
- будь-які операції, що стосуються відображення графа, можуть бути застосовані до усіх вузлів дерева. Наприклад, геометричні перетворення (зсув, поворот і т.п.) можуть належати до всіх піддерев, що виходять з вузла, до якого вони належать;
- можливість інтерактивної взаємодії. Простою вказівкою на екрані можна повідомляти прикладній програмі не тільки ознаку ідентифікації елемента, що вказується (мітку, вагу), але і підструктуру, у яку він входить;
- можливість оптимізації зв'язків шляхом „розумного” використання деревоподібної структури. Мається на увазі вдале розташування вершин за ярусами.

Наведений метод розроблений для помічених орієнтованих графів, що подаються за допомогою матриць суміжності за виходами. Тобто матриць, що мають розмірність $N \times N$, де N – кількість вершин у графі. Елементи матриці суміжності можуть набувати значення 0 або 1. Значення 1 відповідає наявності зв'язку, а значення 0 – відсутності. На рис. 1 наведений орієнтований граф, що має структуру “дерево” та відповідна матриця суміжності $S(6,6)$ [2,4,14].

Враховуючи сучасний стан теорії графів [6] та максимально можливу кількість вершин, що можуть розміститись на екрані монітора без використання операції масштабування [15], пропонується обмежити порядок матриці до 80 вершин. Така кількість є достатньою для візуалізації великих графових структур, що описують складні системи [6].



$$S = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1)$$

Рис. 1. Орієнтований граф та відповідна матриця суміжності за виходами

Запропонований метод розміщення елементів графа на екрані монітора передбачає такі основні етапи: введення в комп'ютер матриці суміжності, що подає граф; визначення номера та координат „опорної вершини”; знаходження кількості вершин, що розміщуються на ярусах; знаходження номерів та координат решти вершин графа.

Перший етап є відносно простим, оскільки він є підготовчим в процесі розміщення елементів на площині і пов'язаний лише з операцією введення матриці суміжності. Матриця суміжності є тією структурою, завдяки якій зберігається внутрішнє подання графа і яка зберігається як двовірний масив однорозрядних двійкових чисел (1).

Що стосується **другого етапу** – визначення номера та координат „опорної вершини” – то він є надзвичайно важливим, оскільки безпосередньо впливає на формування подальшої структури графа. На цьому етапі здійснюється пошук вершини, що має найбільшу кількість зв'язків. Цей процес здійснюється операцією множення введеної матриці суміжності $A[N*N]$ на одиничний вектор-стовпець $B[N]$. Результатом операції є матриця-рядок $C[N]$, елементами якої є цілі числа з діапазону $[0, 1, 2, 3 \dots N]$. Величина чисел матриці $C[N]$ визначає кількість зв'язків відповідної вершини. Наприклад:

$$C=A*B = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \textcircled{1} & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 1 & 0 & 3 & 2 & 1 & 0 \end{pmatrix} \quad (2)$$

„Петля”

Елементи матриці-рядка формуються:

$$1 \text{ елемент: } 1*0+1*0+1*1+1*0+1*1+1*1+1*0+1*0=3$$

$$2 \text{ елемент: } 1*0+1*0+1*0+1*0+1*0+1*0+1*0+1*1=1$$

$$3 \text{ елемент: } 1*0+1*0+1*0+1*1+1*0+1*1+1*0+1*0=1$$

$$4 \text{ елемент: } 1*0+1*0+1*0+1*0+1*0+1*0+1*0+1*0=0$$

$$5 \text{ елемент: } 1*0+1*0+1*0+1*1+1*1+1*0+1*1+1*0=3$$

$$6 \text{ елемент: } 1*0+1*1+1*0+1*0+1*0+1*0+1*0+1*1=2$$

$$7 \text{ елемент: } 1*0+1*0+1*0+1*1+1*0+1*0+1*0+1*0=1$$

$$8 \text{ елемент: } 1*0+1*0+1*0+1*0+1*0+1*0+1*0+1*0=0$$

Далі у сформованому масиві чисел здійснюється процедура швидкого сортування за спаданням [16]. Особливістю даного сортування є те, що в утвореному двовимірному масиві даних ($M [N,N]$) зберігаються номери та відповідні кількості зв'язків кожної з вершин, що дає змогу наочно спостерігати їх взаємовідношення.

При проведенні сортування можливі випадки, коли знаходять декілька вершин з однаковою кількістю зв'язків. У такому випадку здійснюється процедура відкидання одиничних петель (зв'язків самі на себе), суть якої полягає у знаходженні вершин матриці суміжності, елементи головної діагоналі яких дорівнюють одиниці (цикли чи складні петлі на порядок визначення „опорної вершини” не впливають) [2,4]. Зокрема, у матриці $A[8,8]$ п'ята вершина містить петлю, оскільки елемент її головної діагоналі $A[5,5]=1$. З врахуванням описаної процедури або коли не знайдено петель „опорною вершиною” є та, що містить максимум зв'язків та найменший порядковий номер. Необхідність вибору вершини з найменшим порядковим номером пояснюється тільки зручністю сприйняття, тобто, щоби в „корені графа” стояла вершина з мінімально можливим номером. Наприклад, у варіантах, наведених у виразі (3):

$$\begin{array}{cccccc}
 2 & 3 & 5 & \textcircled{8} & 4 & 1 \\
 8 & \textcircled{8} & 8 & 8 & 8 & 8 \\
 2 & 3 & 6 & 8 & \textcircled{8} & 1
 \end{array} \quad (3)$$

будуть вибрані вершини, що обведені колом. Відповідно до деревоподібної структури формування графа знайдена так вершина розміститься у „верхній середині” екрана [12, 18]. Площина екрана з відповідними позначеннями наведена на рис. 2.

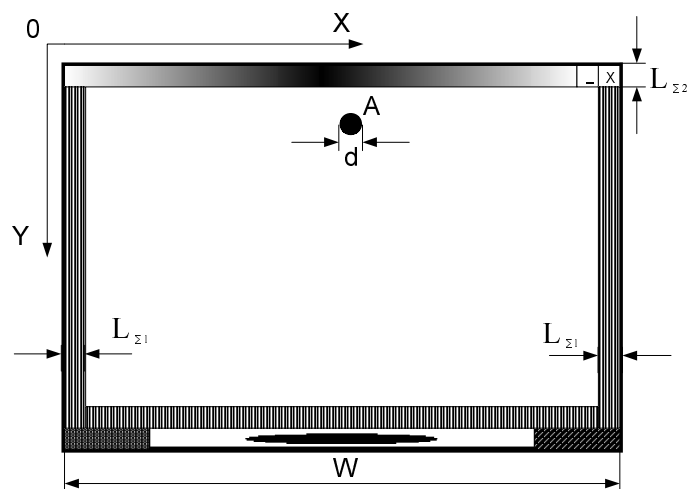


Рис. 2. Основні позначення при знаходженні координат „опорної вершини”

Оскільки прив'язка в розробленому методі є до абсолютних одиниць довжини (міліметрів), а не до пікселів, то координати розміщення вершин та їх розміри не залежатимуть від діагоналі монітора [19,20]. Отже, „верхня середина екрана” знаходитиметься у точці А (рис.2). Для знаходження її координат розміщення необхідно скористатись формулами:

$$\begin{aligned} X &= [W - L_{\Sigma 1}] / 2 \\ Y &= L_{\Sigma 2} + 3 d, \end{aligned} \quad (4)$$

де X, Y – координати „опорної вершини”; W – ширина області екрана; $L_{\Sigma 1}$ – показник, який визначає ширину вікон операційної системи та прикладної програми; $L_{\Sigma 2}$ – показник, який визначає висоту верхнього краю прикладної програми; d – діаметр вершини.

Третій етап роботи передбачає процедуру знаходження кількості вершин графа, що розміщуються на ярусах. А саме, на основі проведеного дослідження стосовно кількості вершин, що розміщуються на екранах різних діагоналей [15] та сформованих критеріїв відображення [5] на даному етапі роботи перевіряється умова (5).

$$\frac{N_1}{N} \geq 2, \quad (5)$$

де N_1 – загальна кількість вершин, що розміщуються на моніторі відповідної діагоналі без використання операції масштабування [15]; N – порядок введеної матриці.

Якщо ця умова виконується, то кількість вершин, що розміщуються на ярусах ($N_{вя}$), зменшується вдвічі [15]. Цей етап необхідний для забезпечення критерію рівномірності [5] при невеликому ($N_1/N \geq 2$) порядку введеної матриці суміжності.

На **четвертому етапі** розміщуються решта вершин. Цей процес здійснюється додаванням до структури графа вершин, що безпосередньо з'єднані з опорною – вершини за номерами 3, 5 та 6 (2). Далі відбувається пошук вершини, що має зв'язки з вже розподіленими вершинами і, якщо таких вершин є декілька, то вибирається та, яка має найменший порядковий номер – вершина за номером 2 (2). Процес продовжується доти, поки не будуть знайдені всі вершини. Їх розміщення розпочинається з лівого краю екрана у ярусі, що розташовується під першим і формується зі сталим кроком, що дорівнює $3d$ між центрами вершин, або $2d$ між їх краями.

Величина кроку вибрана такою для забезпечення неможливості розміщення вершин суміжних ярусів одна під іншою. Координати центрів вершин визначаються за формулами:

$$\begin{aligned} X_v &= L_{\Sigma 1} + 3d * I_p \\ Y_v &= L_{\Sigma 2} + 3d * I, \end{aligned} \quad (6)$$

де X_v, Y_v – координати вершин; I_p – коефіцієнт парності, який визначається залежно від парності стовпця. Якщо стовець, в якому розміщується вершина, парний (0, 2, 4, 6... $N_{вя}$), то коефіцієнт набуває значення з діапазону $I_p = [0, 1, 2, 3...N_{вя}]$, якщо непарний (0, 1, 2, 3... $N_{вя}$) – $I_p = [1, 2, 3...N_{вя}]$. Змінна $N_{вя}$ визначає кількість вершин, що розміщуються в ярусі [5]; I – коефіцієнт рядності, який визначає номер ярусу, в якому здійснюється розміщення, і набуває значення з діапазону $I = [2, 3, 4...N_{я}]$. Значення коефіцієнта збільшується на одиницю з кожним новим ярусом. Змінна $N_{я}$ визначає кількість ярусів графа [5].

З врахуванням наведених формул та описаного розміщення вершини графа розташуються, як показано на рис. 3.

Таке розміщення вершин передбачено для випадку, коли закінчується ярус, в якому здійснюється розташування, а ще існують вершини, які мають спільні зв'язки. Тоді у проміжках між ними можна прокласти дуги без додаткових згинів, тобто мінімальної довжини [5].

Блок-схема цього наведена на рис. 4.

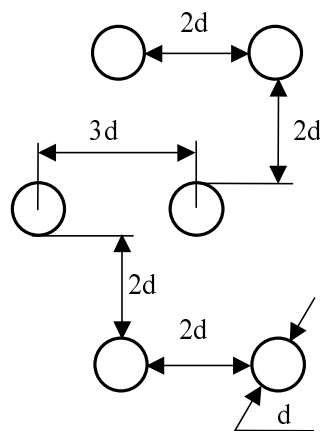


Рис.3. Визначення міжвершинних та між'ярусних віддалей

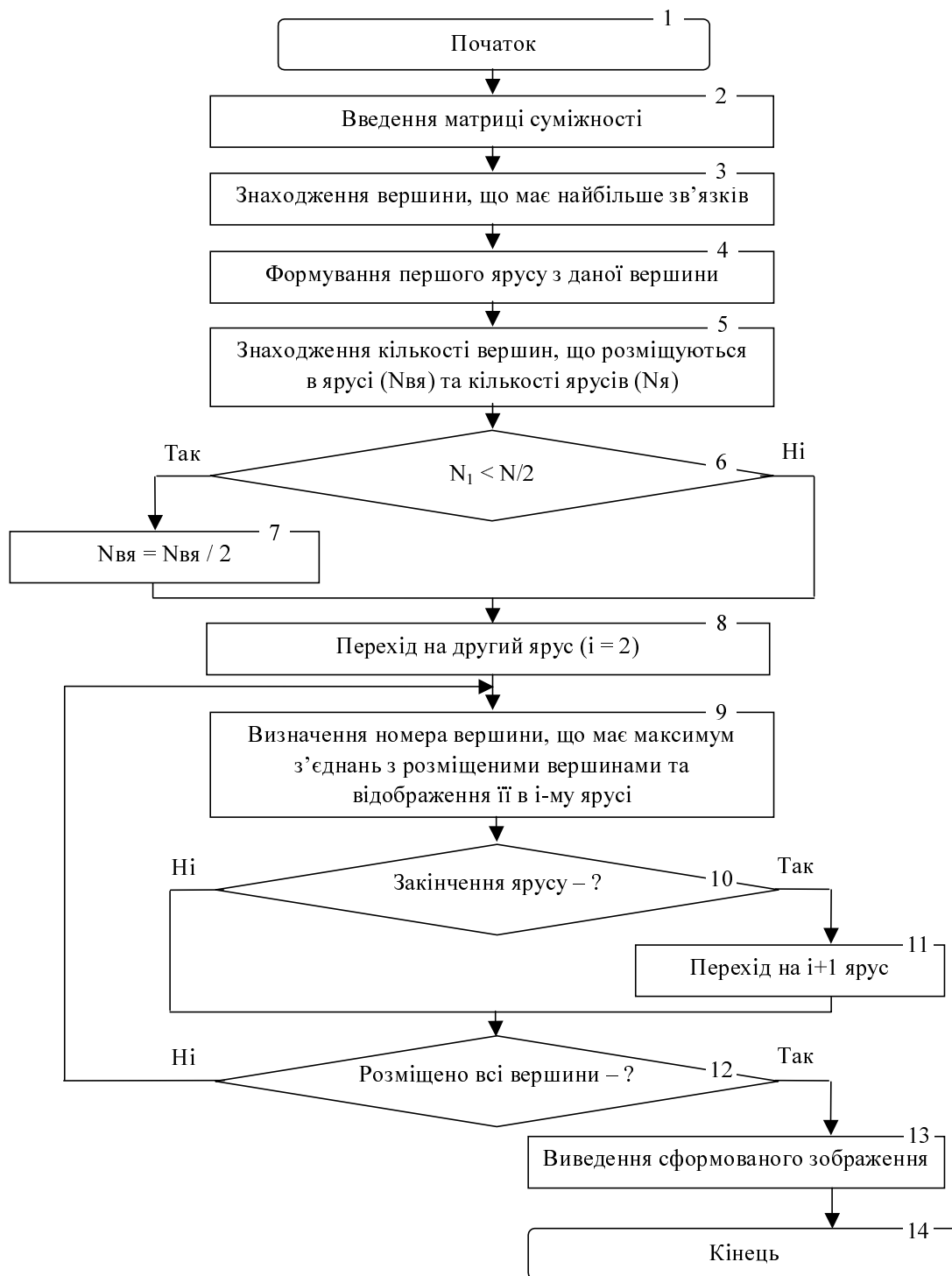


Рис.4. Блок-схема методу розміщення вершин графа

Алгоритм передбачає такі етапи:

1. Початок – на цьому етапі відбувається ініціалізація підпрограми розташування вершин графа та запуск відповідних процедур, які організують взаємодію з користувачем.

2. Введення матриці суміжності – підготовчий етап в процесі відтворення матриці суміжності, який полягає у запам'ятовуванні даної структури даних у відповідному компоненті програмного засобу.

3. Знаходження вершини, що має найбільше зв'язків – на цьому етапі здійснюється пошук вершини, що має найбільше зв'язків. Для цього здійснюються операції: множення введеної матриці на одиничний вектор-стовпець, відкидання петель та швидкого сортування за спаданням.

4. **Формування першого ярусу з даної вершини** – оскільки в результаті рисування утворюється граф деревоподібної форми, то вершина, що має найбільше зв'язків, сформує перший ярус.

5. **Знаходження кількості вершин, що розміщуються в ярусі ($N_{в\text{я}}$) та кількості ярусів ($N_{\text{я}}$)** – відповідно до отриманих формул (4) [15] здійснюється пошук кількості вершин, що розміщується в ярусі та загальної кількості ярусів.

6. $N_1 < N/2$ – за цією умовою порівнюється введений порядок матриці суміжності з $1/2$ кількості вершин, що може розміститись на моніторі з відповідною діагоналлю.

7. $N_{\text{в\text{я}}} = N_{\text{в\text{я}}} / 2$ – при введеному порядку матриці, меншому за $N/2$, у два рази зменшується кількість вершин, що розміщуються на ярусах ($N_{\text{в\text{я}}}$). Проведення такої операції дає змогу ефективніше використовувати робоче поле екрана та сприяє зручності спостереження та оцінювання утвореного рисунка.

8. **Перехід на другий ярус ($i = 2$)** – змінна i позначає яруси; на даному етапі введена для початку формування з другого ярусу, оскільки перший ярус сформований з вершини, що має найбільше зв'язків.

9. **Визначення номера вершини, що має максимум з'єднань з розміщеними вершинами та відображення її в i -му ярусі** – на цьому етапі здійснюється пошук та відображення вершини, що має найбільше з'єднань із вже розміщеними вершинами.

10. **Закінчення ярусу** – відповідно до цієї умови здійснюється перевірка „кінця ярусу”.

11. **Перехід на $i+1$ ярус** – у випадку ствердної відповіді на питання 10 етапу роботи алгоритму здійснюється перехід на новий ярус.

12. **Розміщено всі вершини** – на цьому етапі перевіряється кількість вершин, які були введені в програму та розміщені. На основі різниці кількостей формується відповідна дія. А саме, у випадку негативної відповіді (розміщено не всі вершини) здійснюється перехід до 9 кроку роботи алгоритму.

13. **Виведення сформованого зображення** – у разі успішного виконання всіх кроків алгоритму на екрані формується граф, який відповідає визначеним критеріям відтворення.

22. **Кінець** – закінчення роботи алгоритму та збереження відповідних результатів роботи.

З врахуванням описаного методу елементи матриці суміжності (2) розмістяться, як на рис. 5.

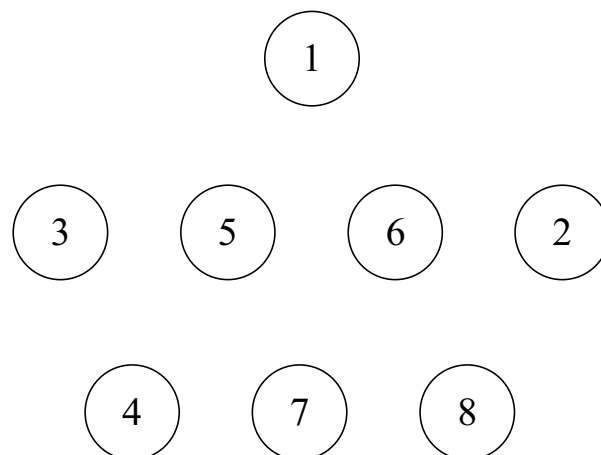


Рис. 5. Розміщення вершин відповідно до описаного методу

Можна зауважити, що отриманий рисунок відповідає сформованим критеріям відображення, а отже, є оптимальним для сприйняття та оцінювання [4,15].

Висновки

Запропонований новий метод розміщення елементів графів, заданих матрицями суміжності дозволяє ефективно розташувати вершини графа на екрані монітора із забезпеченням сформованих

критеріїв відтворення. З його допомогою оптимально розташовуються вершини графа на площині, що дозволяє адекватно оцінювати наведену ним структуру та здійснювати відповідні перетворення над нею.

1. Дунець Р.Б., Басюк Т.М. Основні задачі візуалізації графів, що описують топології поліграфічних систем // Наукові записки УАД. – 2002. – Вип. 5. С.93–96. 2. Дистель Р. Теорія графов. – Новосибірськ: Изд-во Інститута математики, 2002. – 336 с. 3. Зыков А.А. Основы теории графов. – М.: Наука, 1987. – 384 с. 4. Харари Ф. Теория графов- 2 изд. – М.: УРСС, 2003. – 296 с. 5. Басюк Т.М. Критерії відображення графів в процесі візуалізації // Наукові записки УАД. – 2004. – Вип. 7. 6. Касьянов В.Н., Евстегнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ, 2003. – 1104 с. 7. Бабури Д.Е. Иерархический подход для автоматического размещения ациклических графов // Современные проблемы конструирования программ. – Новосибірськ: ИСИ СО РАН, 2002. – С.7–37. 8. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с. 9. Хорн Р., Джонсон Ч. Матричный анализ. – М.: Мир, 1989. – 655 с. 10. Басюк Т.М. Аналіз та класифікація методів візуалізації // Поліграфія і видавнича справа. – 2003. – Вип. 40. – С. 109–114. 11. Gasner E.R., Koutsofios E., North S.C., Vo K.P. A technique for drawing directed graphs // IEEE Transactions on Software Engineering. – 1993. Vol. 19, №3. – P.214–230. 12. Lin X. Analysis of algorithms for drawing graphs: PhD thesis – Dep. of Comput. Sci. Univ. of Queensland. – 1992. 13. Мартинес Ф. Синтез изображений. Принципы, аппаратное и программное обеспечение: Пер. с франц. – М.: Радио и связь, 1990. – 192 с. 14. Дунець Р.Б. Аналіз та синтез топологій комп'ютерних видавничо-поліграфічних систем: НВФ „Українські технології”, Львів – 2003. – 192 с. 15. Басюк Т.М. Фактори вибору графічних примітивів для візуалізації топологій. // Комп'ютерні технології друкарства. – 2004. – №11. – С.124–129. 16. Кнут Д. Искусство программирования: Сортировка и поиск. – М.: Вильямс, 2000. – 822 с. 17. Попов С.Н. Аппаратные средства мультимедиа. Видеосистема РС. – СПб.: БНВ, 2002. – 400 с. 18. Петров М.Н., Молочков В. П. Компьютерная графика (+CD). – СПб.: Питер, 2003. – 736 с. 19. Muller-Hannemann M. Drawing trees, series-parallel digraphs, and Lattices // Lect. Notes Comput. Sci. – 2001. Vol. 2025. – P.46–70. 20. Lenhart W., Liotta G. Proximity drawings of outer planar graphs // Lect. Notes Comput. Sci. – 1997. Vol. 1190. – P.286–302.

УДК 681.518:681.327.8

А.Ю. Берко, В.А. Висоцька, Л.В. Чирун
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

АЛГОРИТМИ ОПРАЦЮВАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ В СИСТЕМАХ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

© Берко А.Ю., Висоцька В.А., Л.В. Чирун, 2004

Проаналізовано основні проблеми електронної комерції та запропоновано методи вирішення цих проблем.

In the given article main problems of electronically commercial are analyzed. New methods for solution of discussed problems are proposed.

Постановка проблеми у загальному вигляді

Електронний бізнес – підвищення ефективності бізнесу, засноване на використанні інформаційних технологій для того, щоб забезпечити взаємодію ділових партнерів і створити інтегрований ланцюжок доданої вартості. Поняття "електронний бізнес" ширше за поняття "електронна