ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: INFORMATYKA z. 37 Nr kol. 1421

Krzysztof DOBOSZ
Politechnika Śląska, Instytut Informatyki

WIZUALIZACJA GRAFOWYCH STRUKTUR DANYCH

Streszczenie. W publikacji opisano problem wizualizacji grafów. Przedyskutowano różne sposoby konstruowania rysunku grafu. Omówiono również metody wizualizacyjne wspomagające analizę struktury grafowej. Oprócz tego wskazane zostały główne kierunki rozwoju w dziedzinie wykreślania grafów.

VISUALIZATION OF GRAPH DATA STRUCTURES

Summary. In the paper, the problem of graph visualization is described. Different methods of constraining image of graph are discussed. Here are also discussed visualization methods supporting analysis of graph structure. Apart from that, main directions of progress on the field of graph drawing are pointed out.

1. Wstęp

W ostatnich latach wiele uwagi poświęca się sposobom wizualizacji struktur danych. Niniejszy artykuł ma na celu przedstawić w zwartej formie podstawowe informacje o wizualizacji grafów na podstawie zebranych materiałów i zgromadzonych do tej pory doświadczeń autora.

Zainteresowanie problemem wizualizacji grafów znacznie wzrosło w ostatnim czasie. Świadczy o tym choćby ostatnie oficjalne zestawienie publikacji naukowych na temat wykreślania grafów, obejmujące kilkaset pozycji [1]. Rozwijane są w nich zagadnienia dotyczące wielu możliwych sposobów wykreślania różnych typów grafów.

Wizualizacja struktur danych nie jest problemem trywialnym. O ile pokazanie prostej struktury listowej nie stanowi większego kłopotu, to umiejętne wykreślenie struktury drzewiastej może już sprawić pewną trudność. Komplikacje te związane są częściej z

optymalnym wykorzystaniem obszaru przeznaczonego do wykreślenia struktury niż ze sposobem prowadzenia krawędzi. Drzewo charakteryzuje się „naturalną" hierarchicznością, co znacznie ułatwia jego wykreślenie. Prawdziwe kłopoty zaczynają się wówczas, gdy należy wykreślić bardziej skomplikowaną strukturę, jaką jest graf [2,3,4], a w szczególności graf cykliczny. W grafie ponadto nie ma ograniczenia mówiącego, że ciąg krawędzi pomiędzy dowolnymi dwoma węzłami musi być jednoznaczny. Zatem poważny staje się problem wykreślenia grafu, czyli struktury, w której najczęściej nie można określić hierarchii i jednoznacznie wyznaczyć kolejności wyświetlania węzłów i krawędzi.

Wizualizacja nie kończy się jednak na samym wykreśleniu struktury. Każdy odbiorca musi mieć możliwość dogłębnego poznania i przeanalizowania zawartości obiektów połączonych w graf. Jest to możliwe dzięki zastosowaniu różnych technik wizualizacyjnych opartych na operacjach graficznych. Dokładniej zostały one opisane w rozdziale 6.

Wizualizacja grafów znajduje szerokie zastosowanie w wielu dziedzinach nauki, np.:

- inżynieria programowania (diagramy przepływu sterowania, schematy blokowe programów);
- projektowanie sieci komputerowych, instalacji elektrycznych, struktur baz danych, oprogramowanie typu CASE, połączeń w strukturach VLSI;
- tworzenie interfejsów graficznych.

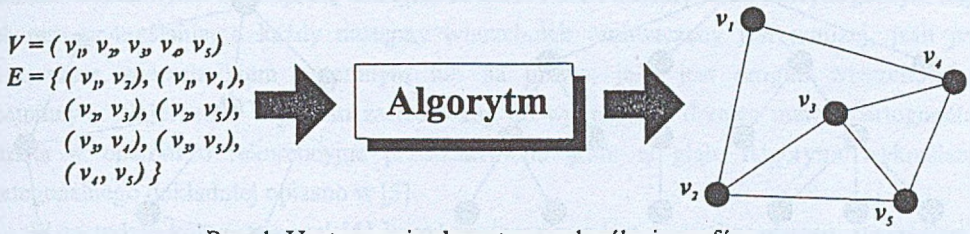
2. Na czym polega wykreślenie grafu?

Zazwyczaj, gdy mamy do czynienia z grafem, trzymając przed oczyma jego rysunek, analizujemy jego strukturę: śledzimy połączenia, znajdujemy cykle, rozpoznajemy charakterystyczne podgrafy itd. W przypadku problemu wykreślania mamy do czynienia z sytuacją zupełnie odmienną: znamy wszystkie właściwości grafu, a szukamy sposobu efektywnego pokazania jego struktury. Wykreślenie grafu polega na skonstruowaniu jego rysunku w zadanej przestrzeni za pomocą ściśle określonej metody.

Przykład:

Mamy zadany graf: $G = (V,E)$, gdzie $V = \{v1, v2, v3, v4, v5\}$ to zbiór węzłów, natomiast $E = \{(v1,v2), (v1,v3), (v2,v4), (v3,v5), (v2,v5), (v3,v4)\}$ to zbiór krawędzi. Następnie określamy cechy, jakie powinien spełniać rysunek grafu. Cechy te obejmują zarówno funkcjonalność rysunku (stopień szczegółowości, czytelność), jak i kryteria estetyczne, np. bezkolizyjne prowadzenie krawędzi, czy też stosowanie symetrii.

Dla danego grafu wykreślenie rysunku jest możliwe w wyniku wykonania określonego algorytmu.



Rys. 1. Usytuowanie algorytmu wykreślania grafów
Fig. 1. The place of a drawing algorithm

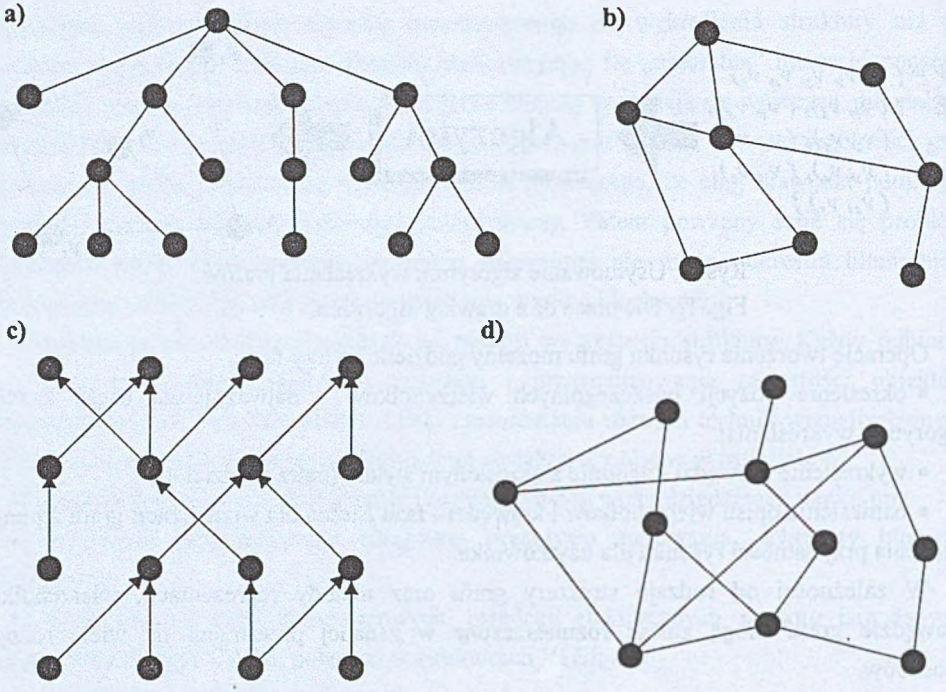
Operację tworzenia rysunku grafu możemy podzielić na trzy fazy:

- określenie pozycji poszczególnych wierzchołków - najważniejsza część każdego algorytmu wykreślania;
- wykreślenie krawędzi - zgodnie z określonym stylem (patrz rozdział 4);
- naniesienie opisu wierzchołków i krawędzi - faza istotna dla wizualizacji grafu z punktu widzenia przydatności rysunku dla użytkownika.

W zależności od rodzaju struktury grafu oraz metody reprezentacji, wierzchołki i krawędzie grafu mogą zostać rozmieszczone w zadanej przestrzeni na wiele różnych sposobów.

3. Klasyfikacja grafów

Ze względu na różne rodzaje grafów bardzo trudno opracować algorytm na tyle uniwersalny, aby jednakowo dobrze wykreślał dowolne grafy. Algorytmy wykreślania zazwyczaj lepiej działają dla pewnego typu grafów, a gorzej dla innych. Dlatego z każdym algorytmem ściśle związana jest klasa grafów, na której on operuje. Poniżej, na rys.2, przedstawiono cztery, podstawowe dla problemu wykreślania, typy grafów.



Rys. 2. Rodzaje grafów: a) drzewo, b) graf planarny, c) graf skierowany, d) graf nieskierowany
Fig. 2. Kinds of graphs: a) tree, b) planar graph, c) directed graph, d) undirected graph

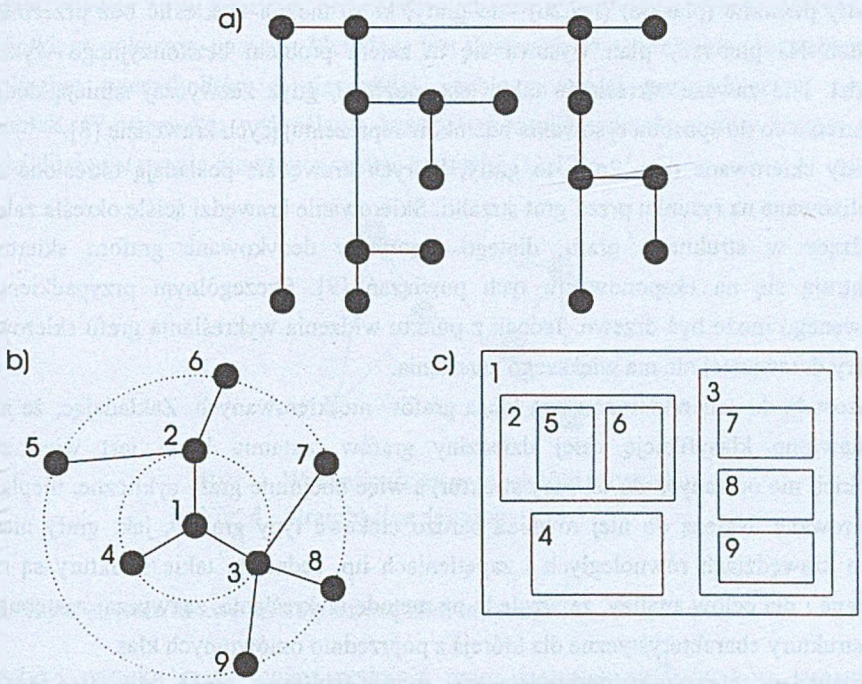
3.1. Drzewa

Drzewa to grafy acykliczne, czyli takie, w których dowolne dwa wierzchołki łączy dokładnie jedna ścieżka zbudowana z krawędzi grafu. Najczęściej spotykany rysunek drzewa (metoda piramidowa) charakteryzuje się tym, że wierzchołki o tej samej długości ścieżki od korzenia umieszczane są wzdłuż tej samej poziomej linii (rys.2a). Tym samym rysunek, w miarę wykreślania wierzchołków i krawędzi bardziej odległych od korzenia, staje się coraz bardziej zagęszczony. Dlatego też podstawową cechą algorytmów do wykreślania tej klasy grafów jest zachowywanie minimalnej odległości pomiędzy wierzchołkami znajdującymi się na tym samym poziomie. Ma to na celu zachowanie czytelności rysunku. Niestety, widoczną wadą takiego rozwiązania jest nierównomierne wykorzystanie przestrzeni, przeznaczonej do wykreślenia drzewa.

Alternatywnymi rozwiązaniami problemu wykreślania drzew są metody: ortogonalna, koncentryczna i blokowa (rys 3). Metoda ortogonalna przeznaczona jest w praktyce wyłącznie

dla drzew binarnych. Cechuje się ona tym, że korzeń umieszczony jest w lewym górnym rogu obszaru wykreślania, a każdy następny wierzchołek umieszczany jest poniżej, jeśli jest pierwszym wierzchołkiem potomnym lub na prawo, jeśli jest drugim wierzchołkiem potomnym. Najprostszy algorytm zastosowany do wykreślania drzewa metodą ortogonalną działa w oparciu o rekurencyjne przeszukiwanie grafu w głąb. Algorytm wykreślania ortogonalnego dokładniej opisano w [5].

W metodzie koncentrycznej [6] węzeł umieszczany jest w środku obszaru wykreślania. Zbiory węzłów potomnych rozmieszczane są mniej lub bardziej równomiernie na sferach otaczających korzeń. Metodę tę cechuje dość dobre wykorzystanie przestrzeni, jednakże rysunek drzewa może sprawiać patrzącemu pewne kłopoty podczas identyfikacji, który z węzłów jest rodzicem, a który potomkiem. Szczególnie trudne jest to w przypadku drzew niebinarnych.



Rys. 3. Metody wykreślania drzew: a) ortogonalna, b) koncentryczna, c) blokowa
Fig. 3. Methods of drawing trees: a) orthogonal, b) concentric, c) block

W metodzie blokowej [7] wierzchołki grafu przedstawiane są w postaci płaskich figur geometrycznych, np. prostokątów o różnej wielkości. Natomiast krawędzie pomiędzy

wierzchołkami ilustrowane są poprzez odpowiednie umieszczenie owych prostokątów względem siebie. Prostokąt wierzchołka potomnego położony jest w obszarze prostokąta reprezentującego wierzchołek nadrzędny. Zaletą metody jest dobre wykorzystanie obszaru przeznaczonego na wykreślenie drzewa. Metoda nie sprawdza się dla drzew o zbyt wielkich odległościach liści od korzenia, ponieważ powoduje brak czytelności obiektów graficznych symbolizujących odległe liście.

Jednak zdecydowanie najpopularniejszym sposobem wykreślania drzew jest rysowanie ich tak, aby zachować na rysunku istniejącą w strukturze hierarchię. Wierzchołki potomne umieszczane są wtedy zawsze w tym samym kierunku względem wierzchołków nadrzędnych, np. z góry na dół jak pokazano na rys 2a.

3.2. Grafy planarne, skierowane i nieskierowane

Grafy planarne (płaskie) (rys.2b) - to grafy, które można wykreślić bez przecinania się krawędzi. Na pierwszy plan wysuwa się tu zatem problem bezkolizyjnego wykreślania krawędzi. Nie zawsze wkreślanie takie jest możliwe, gdyż zazwyczaj istnieją dodatkowe ograniczenia co do sposobu rysowania odcinków reprezentujących krawędzie [8].

Grafy skierowane (rys. 2c) - to grafy, których krawędzie posiadają określone zwroty, symbolizowane na rysunku przez grot strzałki. Skierowanie krawędzi ściśle określa zależności zachodzące w strukturze grafu, dlatego algorytmy dedykowane grafom skierowanym koncentrują się na eksponowaniu tych powiązań [9]. Szczególnym przypadkiem grafu skierowanego może być drzewo. Jednak z punktu widzenia wykreślania grafu skierowalność struktury drzewiastej nie ma większego znaczenia.

Pozostała do scharakteryzowania klasa grafów nieskierowanych. Zakładając, że na rys.2 przedstawiono klasyfikację całej dziedziny grafów, ostatnia klasa jest więc zbiorem wszystkich nie opisanych do tej pory struktur, a więc obejmuje grafy cykliczne, nieplanarne i nieskierowane. Należą do niej również bardzo ciekawe typy grafów, jak: grafy niespójne, grafy o krawędziach równoległych i zapętleniach itp. Jednakże takie struktury są rzadziej spotykane i dla celów analizy, ze względu na metodę wykreślania, zazwyczaj zastępuje się je przez struktury charakterystyczne dla którejś z poprzednio omówionych klas.

Przykładem algorytmu wykreślającego grafy cykliczne może być algorytm, który koncentruje się na pokazywaniu jak największej ilości cykli w postaci geometrycznych figur foremnych. W tym celu na podstawie długości cyklu oraz liczby przyległych do niego krawędzi obliczana jest jego waga. Wszystkie cykle są sortowane wg obliczanych wag, z zachowaniem zasady polegającej na tym, że w posortowanym ciągu każdy z cykli musi mieć co najmniej jeden wierzchołek wspólny z jednym z cykli, które w ciągu go poprzedzają.

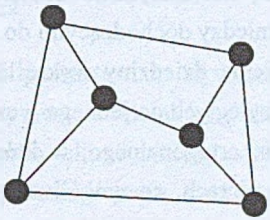
Według tak ustalonego porządku cykle zostają kolejno wykreślone. Algorytm ten dokładnie został opisany w [10].

4. Style wykreślania

W poprzednim rozdziale powiedziano, że metody wykreślania są zazwyczaj ściśle związane z daną klasą grafów. Obecnie zajmiemy się omówieniem uniwersalnych stylów wykreślania, które można zastosować do tworzenia rysunku dowolnego grafu.

4.1. Wykreślanie krawędzi za pomocą linii prostych

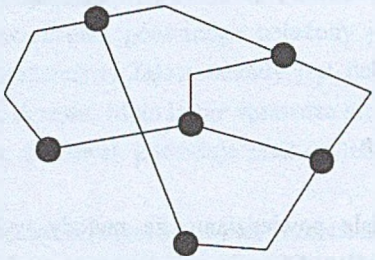
Bez wątpienia metoda ta jest najłatwiejsza do implementacji. Odcinki reprezentujące krawędzie zawarte są w prostych przechodzących przez symbole graficzne reprezentujące wierzchołki, co pokazano na rys.4. Metoda ta jest skuteczna do wykreślenia prostych grafów o małej liczbie wierzchołków i niewielkiej liczbie krawędzi przypadających na jeden wierzchołek. W przypadku wykreślania bardziej skomplikowanych grafów bardzo szybko dochodzi do powstawania punktów przecięć krawędzi [11,12].



Rys. 4. Wykreślanie za pomocą linii prostych
Fig. 4. Straight-line drawing

4.2. Wykreślanie krawędzi za pomocą linii łamanych

Estetykę rysunku grafu można podnieść przez zastosowanie odcinków łamanych do zilustrowania krawędzi. Dzięki temu podczas prowadzenia krawędzi można ominąć miejsca, w których krawędź w postaci odcinka prostego doprowadziłaby do zmniejszenia czytelności rysunku [13], jak pokazano na rys.5.

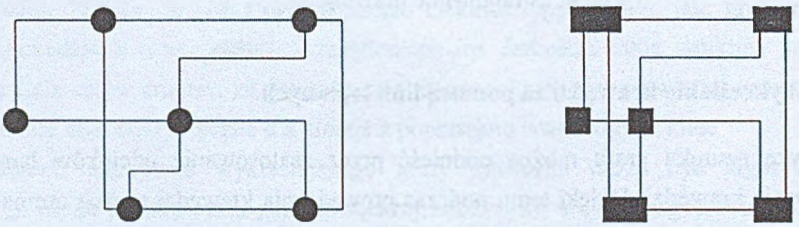


Rys. 5. Wykreślanie za pomocą linii łamanych
Fig. 5. Polyline drawing

4.3. Wykreślanie ortogonalne

W metodzie tej każda krawędź jest reprezentowana jako linia (łamana bądź prosta) przebiegająca wyłącznie poziomo lub pionowo [14], co przedstawiono na rys.6. Metoda ta jest wykorzystywana głównie podczas projektowania obwodów drukowanych lub układów VLSI. Wiąże się z tym konieczność generowania minimalnej ilości *crossingów,* czyli punktów przecinania się krawędzi, co dla powyższego procesu jest rzeczą kluczową. Aplikacja używająca tego stylu zazwyczaj automatycznie dobiera rozmiar każdego węzła, tak aby zachować jednakowy odstęp pomiędzy dochodzącymi do niego krawędziami.

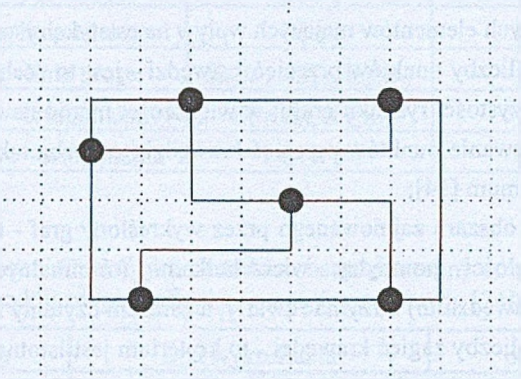
Wadą metody jest ograniczenie dziedziny wykreślanych grafów do takich, w których maksymalna liczba krawędzi przyległych do jednego węzła wynosi 4. Dlatego też spotyka się często inny wariant wykreślania ortogonalnego, w którym wierzchołki reprezentowane są przez prostokąty o różnych wymiarach, co umożliwia dołączenie do nich dowolnej liczby krawędzi.



Rys. 6. Wykreślanie ortogonalne: a) standardowe, b) zmodyfikowane
Fig. 6. Orthogonal drawing: a) standard, b) modified

4.4. Wykreślanie na siatce

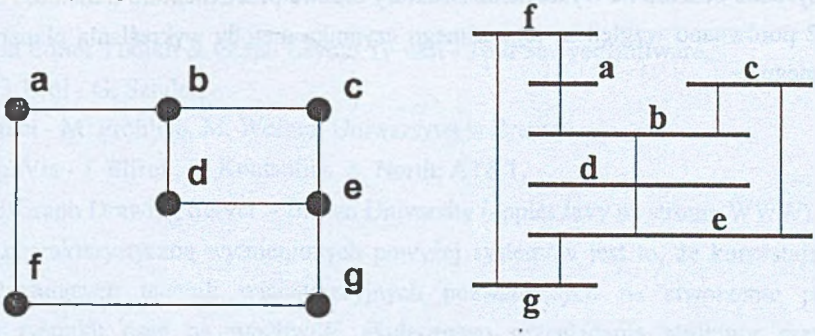
Jest to odmiana wykreślania ortogonalnego. Również tutaj każda krawędź reprezentowana jest jako odcinek prosty lub łamany, równoległy do osi układu współrzędnych. Różnica polega na tym, że każdy wierzchołek i punkt zagięcia krawędzi umieszczony jest w węźle siatki [15].



Rys. 7. Wykreślanie ortogonalne na siatce
Fig. 7. Orthogonal drawing on grid

4.5. Reprezentacja alternatywna

W metodzie tej wierzchołki grafu przedstawiane są w postaci odcinków równoległych do osi poziomej, natomiast krawędzie w postaci odcinków pionowych [16]. Sąsiadujące z sobą wierzchołki grafu są połączone z sobą takimi właśnie odcinkami. Jak łatwo zauważyć, za pomocą opisanej, alternatywnej reprezentacji grafu, można przedstawić pewną hierarchię w wykreślanej strukturze. Jednak poza specyficznymi zastosowaniami, styl ten jest rzadko wykorzystywany, ze względu na stosunkowo dużą zajętość obszaru i małą czytelność przy reprezentacji dużych grafów.



Rys. 8. a) Przykładowy graf, b) reprezentacja alternatywna
Fig. 8. a) Sample graph, b) alternative representation

5. Kryteria estetyczne rysunku

Jak czytelnik zdołał zapewne zauważyć, algorytmy wykreślania grafów oceniane są pod kątem jakości generowanego rysunku, przy czym określenie „jakość" oznacza tutaj jego przejrzystość i czytelność.

Do najważniejszych elementów mających wpływ na estetykę rysunku zaliczamy:

- minimalizację liczby punktów przecięć krawędzi - jest to cecha, która ma podstawowe znaczenie dla przejrzystości rysunku grafu; w większości metod nawet jeśli nie jest możliwe całkowite wyeliminowanie punktów przecięć krawędzi, to liczba takich przecięć jest zawsze ograniczana do minimum [14];
- minimalizację obszaru zajmowanego przez wykreślony graf - tak aby przy zachowaniu minimalnych odległości pomiędzy wierzchołkami (minimalnych długości odcinków odpowiadających krawędziom) otrzymać zwarty, a zarazem czytelny rysunek grafu [17];
- minimalizację liczby zagięć krawędzi - to kryterium jest istotne w metodach rysujących krawędzie w postaci odcinków łamanych (wykreślanie ortogonalne); zbyt wielka liczba zagięć utrudnia prześledzenie połączeń pomiędzy wierzchołkami grafu [14];
- minimalizację odległości pomiędzy wierzchołkami i nieincydentnymi krawędziami - cecha ta powoduje powstawanie zwartego rysunku grafu;
- maksymalizację wartości najmniejszych kątów pomiędzy przyległymi krawędziami - w przypadku stylów wykreślania nieortogonalnych kąt między krawędziami przyległymi do tego samego wierzchołka powinien być jak największy po to, aby nie dochodziło do nakładania się przyległych krawędzi w pobliżu obiektu graficznego reprezentującego wierzchołek grafu [18];
- stosowanie symetrii - znacznie podnosi czytelność rysunku i ułatwia wzrokowe znajdowanie charakterystycznych podgrafów w całej wykreślonej strukturze [19].

Większość metod służących do wykreślania grafów można porównywać z sobą, mając na względzie każdy z powyżej opisanych czynników. Dla przykładu, w tabeli 1 porównano wykorzystanie obszaru na wykreślenie struktury drzewa przez niektóre z metod. Natomiast w tabeli 2 porównano względem tego samego czynnika metody wykreślania planarnego grafu cyklicznego.

Tabela 1
Zapotrzebowanie niektórych metod na obszar potrzebny do planarnego wykreślenia drzewa

metoda wykreślania	obszar	źródło oszacowań
kierunkowa - piramidowa	$\Theta(n^2)$	[20]
kierunkowa - linie łamane	$\Theta(n)$	[21]
kierunkowa - linie proste	$\Omega(n)$ $O(n \log n)$	[22]
kierunkowa - ortogonalna	$\Theta(n \log \log n)$	[21]
niekierunkowa - ortogonalna	$\Theta(n)$	[23]

Tabela 2
Zapotrzebowanie niektórych metod na obszar potrzebny do wykreślenia cyklicznego grafu planarnego (na podstawie [24])

klasa grafu	Styl wykreślania	obszar
planarny	planarny (linie łamane i wierzchołki na siatce)	$\Theta(n^2)$
planarny 3-stopnia	planarny, na siatce	$\Theta(n^2)$
planarny	planarny, ortogonalny na siatce	$\Theta(n^2)$
planarny 4-stopnia	ortogonalny na siatce	$\Omega(n \log n)$ $O(n \log^2 n)$

6. Aplikacje

Rozwojowi badań nad wykreślaniem grafów towarzyszy powstawanie różnorodnych systemów wizualizacji struktur danych. Poniżej wymieniono te dedykowane wizualizacji grafów:

a) Graph Editor Toolkit & Graph Layout Toolkit - Tom Sawyer Software,

b) VCG Tool - G. Sander,

c) daVinci - M. Frohlich, M. Werner; Uniwersytet w Bremen,

d) GraphVis - J. Ellson, E. Koutsofios, S. North; AT&T,

e) GDS:Graph Drawing Server - Brown University (applet Javy na stronie WWW).

Cechą charakterystyczną wymienionych powyżej systemów jest to, że korzystają one z wielu interesujących technik wizualizacyjnych pozwalających na stworzenie pełnego, czytelnego rysunku oraz na możliwość skutecznego przeglądania struktury grafu. Do najczęściej spotykanych cech tych systemów można zaliczyć:

- wykreślenie grafu w jednym z dostępnych styli (ortogonalny, symetryczny, hierarchiczny i inne) - pozwala narzucić określony sposób rozkładu elementów grafu w obszarze rysunku (a, b, d, e);
- możliwość definiowania i wyświetlania opisów wierzchołków i krawędzi - ma znaczenie w przypadku, gdy rysunek grafu powinien zawierać również opisy najistotniejszych jego części (b, c, d);
- definiowanie minimalnych odległości pomiędzy wierzchołkami grafu - nie pozwala na zmniejszenie czytelności rysunku przez zbyt bliskie ułożenie wierzchołków (b, c, d);
- automatyczne usuwanie punktów przecięć krawędzi (a, c, d);
- orientowanie grafów - polega na narzuceniu kierunku, w którym wykreślane są wierzchołki podrzędne (dotyczy grafów skierowanych) (b, c);
- abstrakcja - polega na ukryciu tej części rysunku grafu, która dla użytkownika nie jest istotna; ukrywaniu takiemu najczęściej poddaje się wybrane wierzchołki wraz ze wszystkimi przyległymi do nich krawędziami (a, c);
- nawigacja po strukturze grafu skierowanego wg zwrotu krawędzi - umożliwia śledzenie istniejących w strukturze powiązań (a, b, c);
- inspekcja zawartości poszczególnych wierzchołków grafu oraz krawędzi - pozwala użytkownikowi na przeglądanie i modyfikowanie zawartości obiektu reprezentowanego przez wierzchołek lub krawędź grafu (b, d);
- powiększanie części obrazu grafu - umożliwia dokładniejszą analizę wybranych części grafu; zastosowanie ma tu klasyczna metoda skalowania obrazu bądź też metoda "rybie oko" [25] (a, b, c, d);
- stosowanie perspektywy - powoduje uzyskanie na płaskim rysunku efektu trójwymiarowości (d);
- generowanie rysunków grafu w różnych formatach graficznych - użyteczne do dalszej obróbki rysunku grafu w postaci wektorowej lub w postaci bitmapy (e);
- generowanie plików tekstowych z opisem struktury grafu (zgodnie z przyjętą gramatyką) - przydatne do wykorzystania w innych aplikacjach potrafiących odczytywać tak zapisaną strukturę (a, b, c, e).

7. Uwagi końcowe

Dziedzina prezentacji struktur grafowych jest dziedziną stosunkową młodą i wciąż dynamicznie rozwijającą się. Niejednokrotnie chcąc przedstawić za pomocą rysunku strukturę jakiegoś problemu lub zjawiska, posługujemy się rysunkiem grafu. Aby jednak rysunek taki stworzyć jak najlepiej, należy posłużyć się konkretną metodą wykreślającą graf w określonym

stylu. Następnie, aby móc skuteczniej pracować nad obrazem grafu, stosujemy różne techniki wizualizacyjne. W rezultacie otrzymujemy najczęściej skuteczne narzędzie do konstruowania i przeglądania struktury grafu.

W dziedzinie wizualizacji grafów wciąż pojawiają się nowe zadania do rozwiązania. Aktualne, najistotniejsze problemy w wizualizacji grafów zostały wymienione poniżej:

- Szybkie metody badania planarności grafu.
  Znane algorytmy badania planarności grafów charakteryzują się dużym stopniem skomplikowania i są trudne do zrozumienia oraz do implementacji. Jest to poważne ograniczenie, które praktycznie wyklucza je z zastosowań w tworzonych systemach wizualizacji grafów. Dlatego też wciąż trwają badania nad znalezieniem prostego i efektywnego algorytmu testującego planarność grafu.

- Wykreślanie w przestrzeni 3D.
  Pomimo to, że istnieją już systemy potrafiące wykreślać grafy w przestrzeni trójwymiarowej, teoretyczne badania na metodami prezentacji grafów w tej przestrzeni są jeszcze mało zaawansowane. Najbardziej rozwijającym się kierunkiem są badania nad ortogonalnym wykreślaniem grafów na trójwymiarowej siatce.

- Dynamiczne wykreślanie grafów.
  Wiele systemów wizualizacji dopuszcza możliwość interakcji użytkownika polegającej: na modyfikacji struktury wykreślonego grafu, na zmianie ilości węzłów, krawędzi lub zmianie połączeń w grafie. Po takiej modyfikacji zachodzi konieczność szybkiego, ponownego wykreślenia całej struktury, bądź tylko zmodyfikowanej jej części. Ściśle związany jest z tym problem szybkiego badania planarności grafu.

LITERATURA

1. Di Battista G., Eades P., Tamassia R., Tollis I.G.: Algorithms for Drawing Graphs: an annotated bibliography. Comput. Geom. Theory Appl. 1994, t. 4, s.235-182.
2. Drozdek A., Simon D.L.: Struktury danych w języku C. WNT, Warszawa 1996.
3. Lipski W.: Kombinatoryka dla programistów. WNT, Warszawa 1982.
4. Reingold E. M., Nievergelt J., Deo N.: Algorytmy kombinatoryczne. PWN, Warszawa 1985.
5. Eades P., Lin T., Lin X.: Minimum size h-v drawings. Advances Visual Interfaces 1992, World Scientific Series in Computer Science, vol. 36, s. 386-394.

6. Eades P.: Drawing Free Trees. Bulletin of the Institute for Combinatorics and its Applications, 1992, vol. 5, s. 1-36.
7. Sindre G., Gulla B., Jokstad H.G.: Onion graphs: aesthetics and layout. s. 287-291.
8. Chiba N., Onaguchi K., Nishizeki T.: Drawing planar graphs nicely. Acta Informatica 1985, vol. 22, s.187-201.
9. Eades P., Lin X.: How to draw a directed graph. Proc IEEE Workshop on Visual Languages 1989, p. 13-17.
10. Dobosz K.: Wykreślanie grafów cyklicznych. ZN Pol. Śl. s. Informatyka, Gliwice 1998, z. 35, s. 35-51.
11. Lipton R.J., North S.C., Sandberg J.S.: A Method for Drawing Graphs. Proc. ACM Symp. on Computational Geometry 1985, s. 153-166.
12. Eades P.: A Heuristic for Graph Drawing. Congressus Numerantium 1984 t. 42, s. 149-160.
13. Trickey H.: Drag: A Graph Drawing System. Proc. Internat. Conf. on Electronic Publishing, Cambridge University Press 1988, s. 171-181.
14. Tamassia R.: Planar Orthogonal Drawings of Graphs. Proc. IEEE Int. Symp. on Circuits and Systems, 1990.
15. Tamassia R., Tollis I.G.: Planar grid embending in linear time. IEEE Trans. on Circuits and Systems 1989, vol. CAS-36, nr 9, s. 1230-1234.
16. Wismath S.K.: Characterizing Bar Line-of-Sight Graphs. Proc ACM Symp. on Computational Geometry, 1985, s. 147-152.
17. Di Battista G., Tamassia R., Tollis I.G.: Area requirement and symmetry display of planar upward drawings. Discrete Computational Geometry 1992, t. 7, s. 381-401.
18. Di Battista G., Vismara L.: Angles of planar triangular graphs. Proc. ACM Symp. on Theory of Computing 1993, s. 431-437.
19. Manning J.B: Geometric Symmetry in Graphs. PhD Thesis, Department of Computer Sciences, Purdue University, December 1990.
20. Reingold E., Tilford J.: Tider drawing of trees. IEEE Trans. on Software Engineering 1981, vol. SE-7, nr 2, s 223-228.
21. Garg A., Goodrich M.T., Tamassia R.: Area-efficient upward tree drawings. Proc. ACM Symp. on Computational Geometry 1993, s. 359-368.
22. Crescenzi P., Di Battista G., PiperinoA.: A note on optimal area algorithms for upward drawings of binary trees. Computational Geometry: Theory and Applications, 1992, vol. 2, s. 187-200.
23. Vaucher J. Pretty printing of trees. Software Practice and Expirience 1980, vol. 10, nr 7, s. 553-561.

24. Di Battista G., Eades P., Tamassia R., Tollis I.G.: Graph drawing, algorithms for the visualization of graphs. Prentice Hall Inc., New Jersey 1999.
25. Sarkar M., Brown H.M.: Graphical Fisheye Views of Graphs. Technical report CS-91-61, Dept. of Computer Science, Brown University.

Recenzent: Dr inż. Mieczysław Kłopotek

Wpłynęło do Redakcji 19 listopada 1998 r.

Abstract

The paper presents fundamental informations about visualization of graphs, basing on author's experience and different articles. At the start of this paper, drawing graph problem is characterized. Then, in chapter 3, the classification of graphs with the criteria of different method of drawing is made. Sample graphs are shown in figure 2. In figure 3 different methods of drawing tree graphs are ilustrated. Informations contained in chapter 4 are concentrated on discussing to styles of presentation of different graph structures. Graphs drawed according such styles are shown in figures 4-8. Graph drawing algorithms often are compared in consideration of keeping aesthetic criteria of image of graph. Such example criteria are pointed out in chapter 5. There, in the tables 1 and 2, the issue of the area needed to draw a graph is considered. In chapter 6 applications dedicated to visualization of graphs are pointed out. There is also description of its possibilities. There in the last part of the paper actual problems in visualization of graphs are characterized: simple and fast methods of planarity testing, three dimensional drawing, dynamic drawing algorithms.