

GRAFY DOSKONAŁE NA POTRZEBY ANALIZY KONFIGURACJI STANÓW LOKALNYCH STEROWNIKÓW LOGICZNYCH

Kamil Mielcarek

Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50
e-mail: K.Mielcarek@iie.uz.zgora.pl

STRESZCZENIE

Artykuł próbuje wskazać potencjalnie użyteczne algorytmy do rozpoznawania grafów doskonałych, możliwych do zastosowania w procesach automatycznej syntezy układów cyfrowych oraz przybliżyć temat grafów doskonałych. Prezentowane zagadnienia dają się rozwiązać w czasie wielomianowym, gdzie dla grafów w ogólności problem złożoności należy do klasy problemów NP-zupełnych. Badania w tej dziedzinie są prowadzone w wielu ośrodkach światowych mają na celu przyspieszenie procesów automatycznej analizy i syntezy układów dyskretnych. Analiza grafów doskonałych sugeruje ich wykorzystanie jako formalnego modelu w szeroko rozumianej syntezie systemowej. Użycie grafów doskonałych jako modelu formalnego jest o tyle użyteczne, że znacząca większość zależności, opisujących rzeczywiste układy sekwencyjne, daje w wyniku grafy należące do podklasy grafów doskonałych. Zagadnienie to jest powiązane z wykorzystaniem sieci Petriego do opisu układów cyfrowych.

1. WPROWADZENIE

Zastosowanie grafów skierowało różne dziedziny życia na drogę formalną, opisującą zadania z użyciem narzędzi matematycznych. Taka sytuacja miała też miejsce w procesach komputerowo wspomaganego projektowania układów, gdzie problem kolorowania grafów jest używany w procesach optymalizacji, harmonogramowania, przydziału i współdzieleniu zasobów czy kodowania stanów FSM oraz trasowania [2]. Do innych zastosowań związanych pośrednio z komputerowo wspomaganym projektowaniem należą np. kompilacja kodu czy optymalizacja sieci [1]. Z problemem kolorowania grafów jest również związanych cała gama zagadnień, które tak jak kolorowanie, należą do problemów NP-zupełnych i są realizowane przez algorytmy o nieoptymalnej wydajności w przypadku stosowania ich do grafów w ogólności. Ma to swój wydzźwięk w praktycznych zastosowaniach teorii grafów, gdzie zamiast właściwych wyników przyjmowane są wyniki zbliżone do optymalnych. Jest to wynikiem użycia algorytmów heurystycznych, których rezultaty są oddalone od optimum od 10% do 100%, jednak akceptowane z powodu niższej złożoności algorytmicznej. Czas

wykonania algorytmów operujących na grafach, związanych z zagadnieniami pokrycia klikami czy kolorowania, skutecznie odbija się na pracach projektowych i optymalizacyjnych.

Istnieje klasa grafów nazywanych Grafami Doskonałymi, dla których można uzyskać dokładne wyniki w czasie akceptowalnym, zbliżonym do wielomianowego, które można wykorzystać do różnych zadań występujących w czasie automatycznej syntezy. Potencjalna możliwość wykorzystania zależności powstających przy przekształcaniu grafów przepływu sterowania i danych w grafy, z racji sposobu ich tworzenia należących, do jednej z podklas grafów doskonałych, prowadzi do prac nad opracowaniem modelu formalnego opartego na grafach doskonałych.

Zależności powstałe przy zamianie specyfikacji wejściowej na model opisany za pomocą grafu przepływu sterowania i danych umożliwia stosowanie tej techniki do modelowania potencjalnie równoległych procesów w ramach większego układu opisanego za pomocą grafów doskonałych. Warto również wskazać, że istnieje możliwość stosowania grafów interwałowych (jednej z podklas grafów doskonałych) do zadań związanych z harmonogramowaniem oraz zadań opisujących problemy świata rzeczywistego, do czego grafy interwałowe jako aparat matematyczny nadają się znakomicie. Przyszły rozwój badań nad zastosowaniem grafów doskonałych może doprowadzić do powstania kolejnych ścieżek projektowych, które pozwolą sprowadzić większą część lub cały proces do modelu opartego wyłącznie na grafach doskonałych i wykorzystujących zalety tej klasy grafów widziane pod kątem złożoności algorytmicznej.

2. GRAFY DOSKONAŁE

Rozpoznawanie grafów doskonałych do konkretnego zastosowania należy poprzedzić wnikliwymi badaniami, które z potencjalnie możliwych algorytmów należy zastosować. Z uwagi na bogactwo podklas grafów doskonałych, różne podklasy można wykorzystać w różnych miejscach lub sytuacjach automatycznego procesu syntezy. Specyfika algorytmów oraz potencjalne użycie wcześniej otrzymanych wyników w dalszych częściach procesów może dodatkowo podnieść efektywność stosowanych algorytmów. Z uwagi na potencjalnie najszersze zastosowania praktyczne przedstawiony zostaną podklasy grafów doskonałych, takich jak grafy trójkątne, porównywalności oraz grafów interwałowe.

Można wykazać, że dla dowolnego grafu zachodzą następujące zależności [2], [3]:

$$\omega(G) \leq \chi(G) \quad (1)$$

$$\alpha(G) \leq \kappa(G) \quad (2)$$

czyli najmniejsza liczba kolorów potrzebna do właściwego pokolorowania wierzchołków grafu G jest równa lub większa od największego pełnego podgrafu w grafie G . Jednocześnie

najmniejsza liczba pełnych podgrafów potrzebnych do pokrycia wierzchołków grafu G jest równa lub większa niż rozmiar największego zbioru stabilnego w G . [3]

Nierówności (1) i (2) powodują, że problemy kolorowania właściwego oraz minimalnego pokrycia klikami stają się problemami o wysokim stopniu złożoności obliczeniowej. Jeżeli zamiast nierówności (1) i (2) dla rozpatrywanych grafów wystąpią równości, to wyżej wymienione problemy mogą być rozwiązane w czasie wielomianowym (dla grafów doskonałych).

Nieskierowany graf G jest grafem doskonałym (ang. perfect graph), jeśli spełnia trzy ekwiwalentne własności (Lovász 1972):

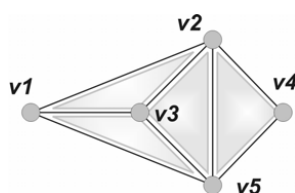
$$\omega(G_A) = \chi(G_A) \quad (\text{dla } \forall A \subseteq V) \quad (3)$$

$$\alpha(G_A) = \kappa(G_A) \quad (\text{dla } \forall A \subseteq V) \quad (4)$$

$$\omega(G_A) \alpha(G_A) \geq |A| \quad (\text{dla } \forall A \subseteq V) \quad (5)$$

dla nieskierowanego grafu $G = (V, E)$. Jednocześnie graf G jest grafem doskonałym, wtedy i tylko wtedy, gdy jego dopełnienie $\bar{G}$ jest grafem doskonałym.

Grafem trójkątnym jest każdy graf, w którym nie występują cykle proste o długości większej niż 3



Rys. 1. Graf trójkątny (przykład)

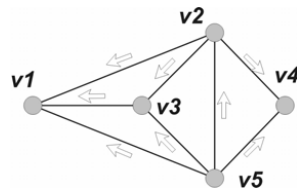
Grafem porównywalności jest nieskierowany graf $G = (V, E)$, dla którego istnieje orientacja (V, F) spełniająca

$$F \cap F^{-1} = \emptyset \quad (6)$$

$$F + F^{-1} = E \quad (7)$$

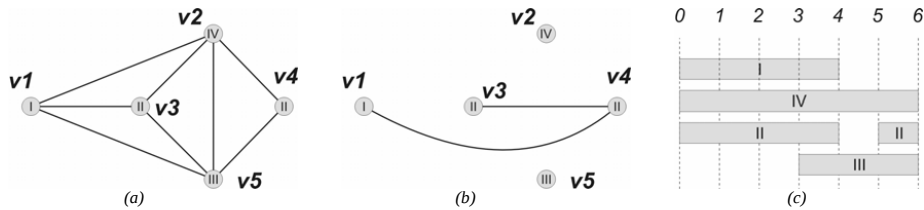
$$F^2 \subseteq F, \text{ gdzie } F^2 = \{(a, c) \mid (a, b), (b, c) \in F\} \text{ dla pewnego wierzchołka } b \quad (8)$$

Relacja F jest częściowym porządkiem zbioru wierzchołków V , gdzie relacją porównywalności jest E , natomiast F jest nazywane orientacją tranzytywną grafu G (lub zbioru krawędzi E). Innymi słowy graf musi posiadać orientację tranzytywną. Grafy porównywalności można spotkać pod nazwą grafów orientowalnych tranzytywnie (ang. transitively orientable).



Rys. 2. Graf porównywalności

Graf interwałowy jest grafem nieskierowanym, w którym wierzchołki można umieścić w przedziałach tak, aby występująca między nimi krawędź odpowiadała części wspólnej przedziałów. Jest to równoważne stwierdzeniu, że graf G jest grafem interwałowym wtedy i tylko wtedy, gdy G jest grafem trójkątnym, a jego dopełnienie $\bar{G}$ jest grafem porównywalności.



Rys. 3. Graf interwałowy (a), jego dopełnienie (b) oraz reprezentacja przedziałowa (c)

3. ROZPOZNAWANIE GRAFÓW DOSKONAŁYCH

Grafy trójkątne

Rozpoznawanie grafów trójkątnych można przeprowadzić dwuetapowo, poprzez obliczenie doskonałego schematu eliminacji wierzchołków (*ang. Perfect Vertex Elimination Scheme - PVES*) lub schematem doskonałym (*ang. perfect scheme*) metodą przeszukiwania leksykograficznego grafu w szerz, a następnie sprawdzić czy wygenerowany schemat jest schematem doskonałym.

Rozpatrując wejściowy graf $G = (V, E)$ będący grafem nieskierowanym oraz $\sigma = \{v_1, v_2, \dots, v_n\}$ będącym pewnym uporządkowaniem wierzchołków możemy mówić o doskonałym schemacie eliminacji wierzchołków, jeśli sąsiedztwo węzła v_i jest podstawą do wyznaczenia podgrafu pełnego. Inaczej mówiąc każdy zbiór $X_i = \{v_j \in \text{Adj}(v_i) \mid j > i\}$ jest grafem pełnym.

Grafy trójkątne charakteryzują się występowaniem takiego węzła. Jeżeli graf nie jest kliką to ma on dwa takie węzły. Drugim etapem (powiązany z poprzednim) jest przeprowadzenie weryfikacji, czy wygenerowane przyporządkowanie σ , jest PVES. Do tego celu używana jest lista $A(u)$ w której gromadzone są wierzchołki, które należy sprawdzić czy są incydentne z wierzchołkiem u . Samo sprawdzenie jest opóźnione aż do momentu gdy $u = \sigma(i)$. Tej techniki używa się dlatego, że w $\sigma^{-1}(v)$ iteracji nie następuje przeszukiwanie węzłów

sąsiadujących z u . Powoduje to, że większa część algorytmu może być implementowane w jednym przebiegu przeszukującym węzły sąsiadujące z v . Instrukcja skoku zostanie wykonana $j-1$ razy, gdzie j oznacza liczbę komponentów grafu G .

```

Etap 1
przypisz etykietę  $\emptyset$  do każdego wierzchołka;
dla  $i \leftarrow n$  do 1 krok (-1) wykonaj
{
    wybór: wziąć nieponumerowany wierzch.  $v$  o największej etykiecie;
     $\sigma(i) \leftarrow v$ ; komentarz: przypisuje do  $v$  numer  $i$ 
    odśwież: dla każdego nieponumerowanego wierzchołka
             $w \in \text{Adj}(v)$  wykonaj dodaj  $i$  do etykieta( $w$ );
}

Etap 2
dla wszystkich wierzchołków  $v$  zrób:  $A(v) \leftarrow \emptyset$ ;
dla  $i \leftarrow 1$  do  $n$  wykonaj
{
     $v \leftarrow \sigma(i)$ ;
     $X \leftarrow \{x \in \text{Adj}(v) \mid \sigma^{-1}(v) < \sigma^{-1}(x)\}$ ;
    jeśli  $X = \emptyset$  to wtedy skocz do linii 8;
     $u \leftarrow \sigma(\min\{\sigma^{-1}(x) \mid x \in X\})$ ;
    połącz  $X - \{u\}$  z  $A(u)$ ;
    jeśli  $A(v) - \text{Adj}(v) \neq \emptyset$  to wtedy zwróć „fałsz”;
}

```

Rys. 4. Algorytm rozpoznawania grafów trójkątnych; [3]

Grafy porównywalności

Rozpoznanie przynależności grafu do podklasy grafów porównywalności przeprowadza się przez próbę zorientowania tranzytywnego wejściowego grafu. Dokonuje się tego przez przypisanie wymuszeń orientacji grafu nieskierowanego, według pewnej relacji. Dokładna definicja wymuszeń, oznaczana jako relacja Γ , jest zapisywana jako

$$ab \Gamma a'b' \Leftrightarrow \{a = a' \text{ oraz } bb' \notin E \text{ lub } b = b' \text{ oraz } aa' \notin E \quad (9)$$

Istnieje zależność powstała przez połączenie w ciąg krawędzi w relacji Γ między sobą, nazywana *klasą implikacji*. Mówimy więc, że krawędzie ab oraz cd należą do tej samej klasy implikacji tylko gdy istnieje sekwencja

$$ab = a_0b_0 \Gamma a_1b_1 \Gamma \dots \Gamma a_kb_k = cd \quad \text{dla } k \geq 0 \quad (10)$$

Tego rodzaju sekwencja nazywana jest łańcuchem Γ z ab do cd . Jeżeli podejmie się próbę zorientowania tranzytywnego grafu zaczynając od krawędzi np. ab a w wyniku stosowania relacji Γ otrzyma się wymuszenie orientacji krawędzi ba można stwierdzić, że graf nie należy do podklasy grafów porównywalności.

```

Inicjuj:  $i \leftarrow 1$ ;  $E_i \leftarrow E$ ;  $F \leftarrow \emptyset$ 
Arbitralnie wybierz krawędź  $x_i y_i \in E_i$ 
Oblicz klasę implikacji  $B_i$  zbioru  $E_i$  zawierającą  $x_i y_i$ 
jeśli  $B_i \cap B_i^{-1} = \emptyset$  to wtedy
    dodaj  $B_i$  do  $F$ 
w przeciwnym razie
    zwróć: „G nie jest grafem porównywalności”
    STOP
Zdefiniuj:  $E_{i+1} \leftarrow E_i - \{B_i \cup B_i^{-1}\}$ 
jeśli  $E_{i+1} = \emptyset$  to wtedy
     $k \leftarrow i$ 
    zwróć  $F$ 
    STOP
w przeciwnym razie
     $i \leftarrow i + 1$ 
    idź do arbitralnego wyboru krawędzi

```

Rys. 5. Algorytm rozpoznawania grafów porównywalności; [3]

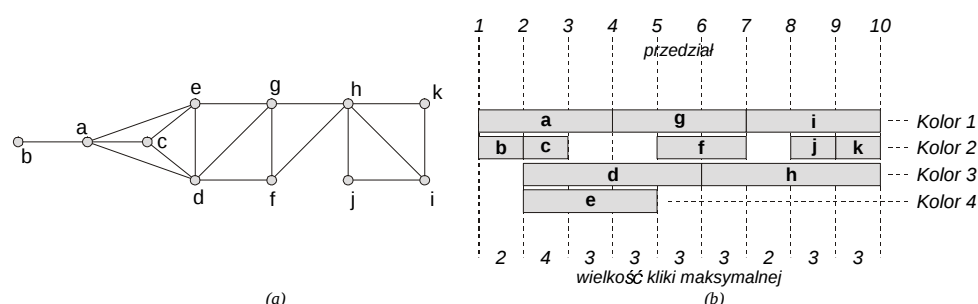
Grafy interwałowe

Rozpoznawanie grafów interwałowych można oprzeć na założeniu, że graf jest grafem interwałowym, jeśli dopełnienie grafu trójkątnego jest grafem porównywalności [3]. Możliwe jest użycie prezentowanych wcześniej algorytmów. Należy pamiętać, że rozpoznanie przynależności do grafów interwałowych będzie prawdopodobnie używane łącznie z rozpoznawaniem dwóch powyższych podklas grafów doskonałych. Istnieją jeszcze inne algorytmy pozwalające w szybki sposób rozpoznać grafy interwałowe. [4]

4. WNIOSKI

Użyteczność metod grafowych wynika z faktu, że wiele operacji wykonywanych na grafach można zinterpretować dla potrzeb projektowania. Przykładowo, jeżeli graf reprezentuje potencjalną równoległość operacji (wierzchołki grafu reprezentują operacje, zaś krawędzie między wierzchołkami odpowiadają potencjalnej równoległości realizacji odpowiednich operacji), to kolorowanie takiego grafu odpowiada dekompozycji na niezależne bloki (np. składowe automatowe). Oprócz kolorowania najczęściej stosowaną operacją jest wyszukiwanie minimalnego pokrycia klikami oraz maksymalnych klik grafu.

Rys. 6.b przedstawia przykładowy graf interwałowy będący wynikiem działania algorytmu kolorowania Left_Edge(). [3] Można zauważyć jak wiele elementów i w jak prostej formie przedstawia. Poziomo ułożone przedziały reprezentują zbiory niezależne, pozwalające jednocześnie odczytać kolorowanie. Zliczając ilość warstw w pionie łatwo można obliczyć wielkość największej maksymalnej kliky w grafie a co za tym idzie dolne ograniczenie ilości kolorów. Informacje te można użyć do pokrycia grafu klikami.



Rys. 6. Graf interwałowy (a) i jego reprezentacja przedziałowa (b)

Na podstawie analizy większej liczby sieci Petriego, opisującej rzeczywiste układy sterowania można wysnuć przypuszczenie, że relacje uporządkowania przestrzeni stanów lokalnych sieci jest opisana grafem doskonałym. Wniosek ten może potwierdzać pośrednio praca [1], w której przeprowadzono znaczną liczbę testów na grafach opisujących rzeczywiste układy cyfrowe. Wyniki prac eksperymentalnych w tej dziedzinie wskazują, że znacząca większość rzeczywistych układów cyfrowych opisanych grafami należącymi do klasy grafów doskonałych daje się właściwie pokolorować w prosty sposób w czasie wielomianowym.

Algorytmy rozpoznawania grafów trójkątnych, porównywalności i interwałowych zostały zrealizowane praktycznie z wykorzystaniem języka C.

LITERATURA

- [1] O. Coudert: *Exact Coloring of Real-Life Graphs is Easy*, Synopsys, Inc. 700 East Middlefield Rd., Mountain View, CA 94043, Proc. of the 34th Design Automation Conference DAC,, Anaheim, CA, USA, June 1997, pp. 121-126
- [2] G. De Micheli: *Synteza i optymalizacja układów cyfrowych*, Wydawnictwa Naukowo-Techniczne, Warszawa 1998
- [3] M. C. Golumbic: *Algorithmic Graph Theory and Perfect Graphs*, Courant Institute of Mathematical Science, New York University, Academic Press 1980
- [4] G. S. Leuker: *Interval graph algorithms. Ph. D. thesis*, Princeton University
- [5] G. Cornuejols, L. Xinming, K. Vuskovic: *A polynomial algorithm for recognizing perfect graphs*, Proceeding 44th Annual IEEE Symposium, 2003

- [6] D. L. Springer, D. E. Thomas: *Exploiting the Special Structure of Conflict and comparability Graphs in high-Level Synthesis*, 1994
- [7] K. Mielcarek: *Grafy doskonałe jako alternatywa dla automatycznej syntezy i optymalizacji układów cyfrowych*, Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim: IV Krajowa Konferencja. Kraków, Polska, 2003
- [8] praca zbiorowa pod redakcją M. Kubale: *Optymalizacja dyskretna. Modele i metody kolorowania grafów*, WNT 2002
- [9] M. Adamski: *Metodologia projektowania reprogramowalnych sterowników logicznych z wykorzystaniem elementów CPLD i FPGA*, Reprogramowalne Układy Cyfrowe RUC 98, Szczecin 12-13.03.1998, ss. 15-22, 1998
- [10] D. Gajski: *System Level Design Flow: What is needed and what is not*, 2002
- [11] D. Gajski, D. Shin: *Scheduling in RTL Design methodology*, 2002
- [12] D. Shin, A. Gerstlauer, R. Dömer, D. Gajski: *C-based Interactive RTL Design Methodology*