

Piotr BOROWIECKI, Elżbieta SIDOROWICZ
Uniwersytet Zielonogórski

EFEKTYWNOŚĆ ALGORYTMÓW DYNAMICZNEGO KOLOROWANIA GRAFÓW - ZASTOSOWANIA W SIECIACH OPTYCZNYCH WDM

Streszczenie. W pracy tej formułujemy problem dynamicznego kolorowania grafów, analizujemy efektywność algorytmu zachłannego First-Fit (w skrócie FF) oraz wskazujemy na jego zastosowanie w problemie przydziału długości fali w sieciach optycznych WDM. W szczególności podajemy dolne i górne oszacowania dobroci algorytmu FF. Wskazujemy istnienie klas grafów G , dla których różnica pomiędzy wartością rozwiązania generowanego przez algorytm $FF(G)$ a wartością optymalną $OPT(G)$ może być dowolnie duża. Z drugiej strony dowodzimy, że dla dowolnego grafu G używanego przez nas w problemie przydziału długości fali zawsze zachodzi $FF(G) < 20PT(G)$.

EFFECTIVENESS OF DYNAMIC GRAPH COLORING ALGORITHMS - APPLICATIONS TO WDM OPTICAL NETWORKS

Summary. Within this paper we introduce a problem of dynamic graph coloring and analyze effectiveness of greedy algorithm First-Fit (FF for short). We point out an important application of a new model to wavelength assignment problem in WDM networks. In particular, we give lower and upper bounds on the performance ratio of FF. We prove that for some classes of graphs G , the difference between the solution value $FF(G)$ and optimum value $OPT(G)$ may be arbitrarily large. On the other hand, for all graphs, that we used in the wavelength assignment problem $FF(G) < 20PT(G)$ holds.

1. Wprowadzenie

W pracy tej formułujemy i analizujemy nowy problem - problem dynamicznego kolorowania grafów. Rozważania prowadzimy dla *grafów prostych*, czyli grafów nie zawierających pętli oraz wielokrotnych krawędzi. Dla grafu $G = (V, E)$ przez $V(G)$ oznaczamy zbiór wierzchołków, natomiast $E(G)$ jest zbiorem krawędzi. Rzędem grafu G jest liczba całkowita n równa $|V(G)|$. Kolorowanie wierzchołków grafu w klasycznym sformułowaniu polega na przypisaniu każdemu wierzchoł-

kowi v koloru $c(v)$ tak, aby dowolne dwa sąsiednie wierzchołki miały różne kolory. Pokolorowanie grafu przy użyciu k kolorów, nazywane jego k -pokolorowaniem, jest podziałem (V_1, V_2, \dots, V_k) zbioru wierzchołków na k podzbiorów niezależnych (klas kolorów). Problem kolorowania wierzchołków grafu jest problemem wyznaczenia takiego podziału, dla którego liczba klas kolorów jest najmniejsza. Liczbę tę oznaczamy $\chi(G)$ i nazywamy *liczbą chromatyczną* grafu G .

W odróżnieniu od tradycyjnego trybu kolorowania, w którym struktura grafu jest dana z góry, a kolejność przypisywania kolorów jest dowolna, w trybie dynamicznym algorytm nie ma wpływu na tę kolejność, a pewne fragmenty grafu mogą być ujawniane algorytmowi dopiero w trakcie kolorowania. Podczas ujawniania nowego wierzchołka v_i ujawniany jest również podzbiór krawędzi łączących v_i z niektórymi spośród zaprezentowanych wcześniej wierzchołków v_1, \dots, v_{i-1} . Prezentacja struktury grafu przebiega jak podczas kolorowania w trybie on-line, jednak o dynamice oprócz zmian wiedzy o grafie stanowi również nietrwałość kolorów. Podobnie jak w trybie on-line algorytm nie ma uprawnień do zmiany raz przypisanego koloru, może jednak stanąć wobec żądania ponownego pokolorowania wierzchołka, który z pewnych powodów utracił kolor. Dynamiczne kolorowanie grafu można interpretować jako dwuosobową grę *prezentera* i *malarza*. Ruchy graczy wykonywane są na przemian. W każdym ruchu prezenter może wywabić kolory dowolnych wierzchołków, ujawnić nowe wierzchołki wraz z incydentnymi krawędziami, a następnie musi zażądać od malarza pokolorowania wskazanego wierzchołka. W odpowiedzi malarz nadaje kolor wskazanemu wierzchołkowi. Celem malarza jest użycie jak najmniejszej liczby kolorów, podczas gdy prezenter stara się, aby było ich jak najwięcej. O zakończeniu gry decyduje prezenter. Malarz wygrywa, jeżeli użyje nie więcej niż $\chi(G)$ kolorów.

Dla wielu problemów występujących w rzeczywistych aplikacjach nie można przyjąć, że cała instancja problemu dana jest z góry, co więcej, praktyczną przydatność algorytmów ogranicza również założenie o trwałości raz wygenerowanego rozwiązania. Potrzeba tworzenia dynamicznych algorytmów jest więc całkiem naturalną konsekwencją dynamiki modelowanych procesów. Przykład jednego z bardzo wielu możliwych zastosowań kolorowania dynamicznego podajemy w ostatniej części tej pracy.

2. Problem dynamicznego kolorowania

Niech $\sigma = (\sigma_1, \dots, \sigma_r)$ będzie ciągiem ruchów prezentera, natomiast G_i grafem ujawnionym po i ruchach prezentera. Każdy ruch σ_i opisany jest czwórką $(V_i^\circ, V_i, E_i, v_i)$, gdzie V_i° jest zbiorem wierzchołków do wywabienia, V_i jest zbiorem nowych wierzchołków, E_i podzbiorem wszystkich krawędzi uv takich, że $u \in V_i$, $v \in V(G_{i-1}) \cup V_i$, natomiast v_i jest wierzchołkiem, wytypowanym do pokolorowania. Traktując ruchy prezentera jako żądania wobec malarza, który je obsługuje, problem dynamicznego kolorowania grafu możemy zapisać następująco:

Problem [Dynamiczne kolorowanie grafu]

Żądanie : $\sigma_i = (V_i^\circ, V_i, E_i, v_i)$, wywabić kolory wierzchołków V_i° oraz pokolorować wierzchołek v_i .

Obsługa : wywabienie kolorów wierzchołków zbioru V_i° oraz pokolorowanie v_i .

Cel : minimalizacja liczby użytych kolorów.

Należy zauważyć, że kolorowanie dynamiczne jest uogólnieniem kolorowania w trybie on-line. Jeżeli w każdym ruchu $V_i^\circ = \emptyset$ i $V_i = \{v_i\}$, to kolorujemy w trybie on-line, a ciąg ruchów prezentera jest jednoznacznie opisany przez permutację zbioru wierzchołków. Dla podkreślenia różnic pomiędzy kolorowaniem w trybie on-line a kolorowaniem dynamicznym rozważmy przykład dotyczący ścieżek.

Przykład 1. Niech dany będzie następujący ciąg ruchów prezentera:

$\sigma = ((\emptyset, \{v_2\}, \emptyset, v_2), (\emptyset, \{v_3\}, \{v_2v_3\}, v_3), (\{v_2\}, \{v_1\}, \{v_1v_2\}, v_1), (\emptyset, \emptyset, \emptyset, v_2))$. Podczas kolorowania grafu przez malarza stosującego zasadę przypisywania możliwie najmniejszego koloru wierzchołki otrzymują kolejno: $c(v_2) = 1$, $c(v_3) = 2$, $c(v_1) = 1$ oraz $c(v_2) = 3$. Zaprezentowany graf jest trójwierzchołkową ścieżką P_3 . Kolorowanie tego grafu tym samym algorytmem lecz w trybie on-line zawsze prowadzi do optymalnego pokolorowania dwoma kolorami. Wiadomo jednak, że już dla ścieżki P_4 każdy algorytm on-line można zmusić do użycia trzech kolorów. ■

W dalszej części pracy podajemy i analizujemy zachłanny algorytm kolorowania dynamicznego. Ze względu na ścisły związek obu kolorowań, w analizie kolorowania dynamicznego korzystamy również z oszacowań parametrów zdefiniowanych dla trybu on-line. Najważniejszy z nich jest odpowiednik klasycznej liczby chromatycznej - *liczba on-line chromatyczna grafu G dla algorytmu A* , którą oznaczamy $\chi_A(G)$ i definiujemy jako największą liczbę kolorów użytych przez algorytm A do pokolorowania grafu G . Bardziej formalnie, jeżeli $A(G, \pi)$ oznacza liczbę kolorów użytych przez algorytm on-line A przetwarzający wierzchołki w kolejności zadanej przez permutację π , to

$$\chi_A(G) = \max_{\pi} A(G, \pi),$$

przy czym maksimum dotyczy wszystkich możliwych permutacji π . Analogicznie definiujemy *dynamiczną liczbę chromatyczną grafu G dla algorytmu A* , z tą różnicą, że maksimum brane jest po wszystkich możliwych ciągach ruchów prezentera.

$$\chi_A^d(G) = \max_{\sigma} A(G, \sigma).$$

Ponieważ π opisuje szczególny ciąg ruchów σ , więc pomiędzy zdefiniowanymi liczbami zachodzi następujący związek:

$$\chi(G) \leq \chi_A(G) \leq \chi_A^d(G). \quad (1)$$

Mówimy, że algorytm kolorowania dynamicznego A jest *efektywny* dla rodziny grafów \mathcal{G} , jeżeli istnieje funkcja $f(\chi)$ taka, że dla każdego $G \in \mathcal{G}$ i dowolnego ciągu ruchów prezentera liczba kolorów użytych przez A wynosi co najwyżej

$f(\chi(G))$. Jakość rozwiązań generowanych przez algorytmy kolorowania oceniamy między innymi na podstawie współczynnika $r_A(G) = A(G)/\chi(G)$ nazywanego *dobrocią algorytmu A dla grafu G*, przy czym w zależności od trybu kolorowania $A(G) = \chi_A^d(G)$ albo $A(G) = \chi_A(G)$ oraz na podstawie *funkcji dobroci* $\rho_A(n): \mathbb{N} \rightarrow \mathbb{R}$ algorytmu A zdefiniowanej jako $\rho_A(n) = \max r_A(G)$, gdzie maksimum brane jest po wszystkich niezomorficznych grafach o n wierzchołkach. Jeżeli ten sam algorytm jest stosowany w obu trybach kolorowania, to w celu wyróżnienia trybu dynamicznego w oznaczeniach dodajemy indeks d , odpowiednio $r_A^d(G)$ i $\rho_A^d(n)$.

3. Algorytm zachłanny

Jednym z najbardziej znanych algorytmów kolorowania jest algorytm zachłanny First-Fit, w skrócie FF. Algorytm ten pomimo liniowej funkcji dobroci zwykle dobrze sprawdza się w praktyce zarówno w trybie off-line jak i on-line [1, 3]. Stanowi on również podstawę działania wielu innych algorytmów. Podajemy tu wersję przeznaczoną do dynamicznego kolorowania grafów.

ALGORYTM First-Fit;

BEGIN

$V(G) \leftarrow \emptyset; E(G) \leftarrow \emptyset; i \leftarrow 0;$

REPEAT

$i \leftarrow i + 1;$

Odczytanie $(V_i^\circ, V_i, E_i, v_i);$

IF $V_i^\circ \neq \emptyset$ THEN Wywabienie_kolorów $(V_i^\circ);$

$V(G) \leftarrow V(G) \cup V_i;$

$E(G) \leftarrow E(G) \cup E_i;$

$v_i.kolor \leftarrow$ Najmniejszy_dopuszczalny_kolor $(v_i);$

UNTIL koniec $\sigma;$

END.

3.1. Oszacowanie liczby χ_{FF}^d dla dowolnych grafów

Algorytm FF zastosowany do kolorowania dynamicznego podlega podobnym ograniczeniom jak podczas kolorowania w trybie off-line oraz on-line. Na przykład, dla dowolnego wierzchołka v , kolor przypisany mu przez algorytm FF nigdy nie będzie większy niż $\deg(v) + 1$. Implikuje to następujące oszacowanie:

$$\chi_{FF}^d(G) \leq \Delta + 1, \quad (2)$$

gdzie Δ oznacza stopień grafu G . Ponadto, biorąc pod uwagę (1), możemy zapisać następujące twierdzenie:

Twierdzenie 1. *Jeżeli G jest dowolnym grafem prostym takim, że $\chi_{FF}(G) = \Delta + 1$, to $\chi_{FF}^d(G) = \chi_{FF}(G)$.*

Przykładami grafów, o których mowa w tym twierdzeniu, są grafy pełne K_n , ścieżki P_n o co najmniej czterech wierzchołkach, cykle C_n , $n \neq 4$ oraz grafy Johnsona i zdefiniowane w [2] drzewa kanoniczne. Dokładniejsze omówienie oszacowań liczby $\chi_{\text{FF}}(G)$ dla poszczególnych rodzin grafów znajduje się np. w pracach [1, 2].

3.2. Oszacowanie liczby χ_{FF}^d dla drzew

Szczególną rolę w analizie kolorowania dynamicznego zajmują *drzewa wymuszające*, których konstrukcja jest zbliżona do konstrukcji wspomnianych drzew kanonicznych. Pierwszymi trzema drzewami wymuszającymi są odpowiednio grafy K_1, K_2 i P_3 , przy czym korzeniem trzeciego drzewa jest środkowy wierzchołek. Kolejne drzewa $T_k, k > 3$ otrzymujemy biorąc dwie wierzchołkowo rozłączne kopie drzew T_{k-1} , nazywając je odpowiednio prawą i lewą kopią, a następnie łącząc ich korzenie krawędzią. Za korzeń drzewa T_k przyjmujemy korzeń lewej kopii. Stosując odpowiednią strategię dla drzewa wymuszającego T_k , można zmusić algorytm FF do użycia k kolorów. Mówi o tym następujący lemat:

Lemat 1. *Jeżeli T_k jest drzewem wymuszającym, to $\chi_{\text{FF}}^d(G) \geq k$.*

Dowód. Dla $k = 1, 2$ uzasadnienie jest łatwe. Strategię dla $k = 3$ znajdujemy w przykładzie 1. Przy założeniu, że lemat jest prawdziwy dla $k - 1$, prezenter ma strategię zmuszającą FF do przypisania korzeniowi drzewa T_{k-1} koloru $k - 1$. Niech x będzie korzeniem lewej kopii poddrzewa T_{k-1} , a y korzeniem prawej kopii. Stosując wspomnianą strategię kolejno dla lewej a następnie prawej kopii otrzymujemy kolor $k - 1$ w wierzchołku x oraz ponieważ x i y są połączone krawędzią, kolor k w wierzchołku y . \square

Ponieważ liczba chromatyczna każdego drzewa wynosi 2, więc różnica $\chi_{\text{FF}}^d(T) - \chi(T)$ może być dowolnie duża, a dobroć algorytmu FF w ogólnym przypadku nie może być ograniczona przez stałą (w punkcie 4., w kontekście przydziału kanałów w sieci WDM podamy przykład rodziny grafów, dla której dobroć algorytmu FF jest ograniczona przez 2). Z podanej konstrukcji drzew wymuszających wynika również, że liczba wierzchołków drzewa wymuszającego T_k dla $k \geq 3$ jest równa $3 \cdot 2^{k-3}$. Można więc zapisać następujące twierdzenie:

Twierdzenie 2. *Dla każdego naturalnego $n > 2$ istnieje takie drzewo T o n wierzchołkach, że $\chi_{\text{FF}}^d(T) \geq \log_2 n + c$, gdzie $c = 3 - \log_2 3$.*

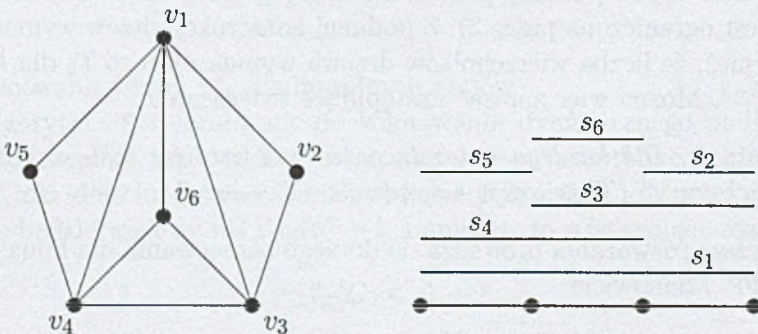
Powyższe rozważania prowadzą do dolnego oszacowania dla funkcji dobroci algorytmu FF. Mianowicie

$$\rho_{\text{FF}}^d(n) \geq \frac{\log_2 n}{2} + \frac{1}{2}(3 - \log_2 3)$$

albo krócej $\rho_{\text{FF}}^d(n) = \Omega(\log n)$.

4. Przydział długości fali w sieci WDM

Postęp w technologii wytwarzania laserów umożliwił wykorzystanie jednego włókna światłowodowego do równoczesnego przesyłania wielu strumieni danych transmitowanych niezależnie na różnych długościach fal świetlnych. Technika ta nazywana jest *zwielokrotnieniem falowym*, w skrócie WDM (ang. *Wavelength Division Multiplexing*). Uproszczonym modelem sieci typu WDM jest *graf sieci*, którego wierzchołki odpowiadają węzłom sieci, a krawędzie reprezentują dwukierunkowe łącza. Do transmisji danych pomiędzy dwoma węzłami s i t wykorzystywane są łącza odpowiadające krawędziom ścieżki $p = (v_s, \dots, v_t)$ wytyczonej w grafie sieci pomiędzy wierzchołkami v_s i v_t podczas inicjowania sesji komunikacyjnej. Zakładamy, że węzły sieci nie mają możliwości konwersji długości fali, zatem łącza przyporządkowane sesji muszą ją obsłużyć korzystając z jednej długości fali. Problem przydziału długości fali polega na przypisaniu każdej sesji jednej długości fali tak, aby *sesje w konflikcie*, tzn. wykorzystujące przynajmniej jedno wspólne łącze, otrzymały różne długości fal, przy czym liczba wykorzystanych długości fal powinna być jak najmniejsza. *Grafem konfliktów zbioru sesji* nazywamy graf, w którym każdemu wierzchołkowi odpowiada jedna sesja, natomiast dwa różne wierzchołki są sąsiednie wtedy i tylko wtedy, gdy odpowiadające im sesje są w konflikcie (jest to graf przecięć krawędzi ścieżek wytyczonych w grafie sieci G). W przypadku gdy zbiór ścieżek zawiera wszystkie możliwe różne ścieżki grafu sieci, to graf ten nazywamy *grafem konfliktów dla sieci G* (patrz rys. 1). Problem przydziału długości fali można rozwiązać kolorując wierzchołki grafu konfliktów sesji. W rozważanym przez nas modelu dynamicznym wywabianie kolorów pozwala na symulację zakończenia sesji, podczas gdy żądanie pokolorowania kolejnego wierzchołka inicjuje nową sesję. Prezentowane tu rezultaty dotyczą przypadku, w którym w dowolnej chwili pomiędzy parą węzłów ustanowiona jest co najwyżej jedna sesja komunikacyjna.



Rys. 1. Graf konfliktów i sesje dla sieci P_4

Twierdzenie 3. *Jeżeli G_k jest n -wierzchołkowym grafem konfliktów dla sieci P_k , $k \geq 2$, to $\chi_{FF}^d(G_k) = n$.*

Dowód. Ze względu na dużą złożoność i objętość formalnego dowodu przedstawiamy jedynie strategię prezentera dla $n = 6$ (oznaczenia zgodnie z rys. 1). Uważny Czytelnik będzie w stanie rozszerzyć ją na pozostałe przypadki. Relacja przebiegu gry wygląda następująco: kolejno prezentuj i koloruj wierzchołki v_1, v_4, v_6, v_3 , wywab kolor wierzchołka v_4 oraz zaprezentuj i pokoloruj v_5 , pokoloruj ponownie v_4 , wywab kolor wierzchołka v_1 oraz zaprezentuj i pokoloruj v_2 , pokoloruj wierzchołek v_1 . Wierzchołkowi v_1 przypisany został kolor 6. \square

Okazuje się jednak, że nawet wtedy, gdy algorytm FF przypisze każdej sesji różny kolor, to liczba użytych kolorów nie przekroczy dwukrotnej wartości rozwiązania optymalnego. Mówi o tym kolejne twierdzenie.

Twierdzenie 4. *Jeżeli G_k jest grafem konfliktów dla sieci P_k , $k \geq 2$, to*

$$r_{\text{FF}}^d(G_k) = 2 - \frac{1}{\lfloor \frac{k+1}{2} \rfloor}.$$

Dowód. Z twierdzenia 3 wiadomo, że $\chi_{\text{FF}}^d(G_k)$ równa jest liczbie różnych ścieżek w grafie P_k , która wynosi $k(k-1)/2$. Pamiętając, że rozważane tu grafy konfliktów sieci są grafami przecięć przedziałów, których liczba chromatyczna równa jest liczbie klikowej, można zauważyć, jak rozszerzenie grafu sieci z P_k do P_{k+1} wpływa na ich liczbę chromatyczną. Otrzymujemy $\chi(G_2) = 1$ i $\chi(G_k) = \chi(G_{k-1}) + \lfloor (k-1)/2 \rfloor$ dla $k > 2$. Stąd $\chi(G_k) = \sum_{i=2}^k \lfloor (i-1)/2 \rfloor$. Wartość sumy obliczamy niezależnie dla parzystych i nieparzystych wartości k , otrzymując odpowiednio $k^2/4$ i $(k^2-1)/4$. Daje to $r_{\text{FF}}^d(G_k) = 2\frac{k-1}{k}$, dla k parzystych oraz $r_{\text{FF}}^d(G_k) = 2\frac{k}{k+1}$ dla nieparzystych. Spostrzeżenie, że obie wartości są równe dla nieparzystego k i następującego po nim parzystego $k+1$, prowadzi do postaci wyrażenia użytej w tezie twierdzenia. \square

Wniosek 1. *Algorytm kolorowania dynamicznego FF jest algorytmem efektywnym dla grafów konfliktów sieci o strukturze ścieżki.*

Z powyższego twierdzenia wynika również górne oszacowaniem dla liczby χ_{FF} w podrodzinie grafów przedziałów, którą tworzą grafy konfliktów dla sieci o strukturze ścieżki. Najlepsze obecnie oszacowanie górne dla dowolnych grafów przedziałów podano w [4] i wynosi ono $\chi_{\text{FF}}(G) \leq 10\chi(G)$, jednak na podstawie twierdzenia 4 i nierówności 1 można stwierdzić, że $\chi_{\text{FF}}(G_k) < 2\chi(G_k)$.

LITERATURA

1. Borowiecki P.: Kolorowanie w trybie on-line. W: M. Kubale red., Optymalizacja dyskretna. Modele i metody kolorowania grafów, WNT, Warszawa 2002, s. 53–71.
2. Gyárfás A., Lehel J.: First-Fit and on-line chromatic number of families of graphs. *Ars Combinatoria* **29C**, 1990, p. 168–176.

3. McDiarmid C.J.H.: Coloring random graphs badly. W: R.J. Wilson red., Graph Theory and Combinatorics, Pitman, 1979, p. 76–86.
4. Pemmaraju S.V., Raman R., Varadarajan K.: Buffer minimization using max-coloring, Proc. 15th ACM-SIAM Symposium on Discrete Algorithms, 2004.

Recenzent: Prof. dr hab. inż. Marek Kubale

Abstract

Dynamics is an inherent feature of many real life systems thus it's quite natural to define their theoretical models in a manner that reflects it. In this paper we investigate a dynamic approach to a well known problem of graph coloring. We point out an important application of a new model to wavelength assignment problem in WDM optical networks, where WDM stands for Wavelength Division Multiplexing. In dynamic graph coloring the graph we color is not given in advance and new vertices together with some adjacent edges are presented to algorithm during the coloring process. Moreover some vertices may lose their colors i.e., an algorithm may be asked to color them again. We formally define dynamic graph coloring problem, dynamic chromatic number and analyze effectiveness of dynamic greedy algorithm First-Fit (FF for short). Lower and upper bounds on the performance ratio of FF are given and we prove that for some classes of graphs G , the difference between the solution value $FF(G)$ and optimum value $OPT(G)$ may be arbitrarily large. On the other hand, for all graphs that we used to study the wavelength assignment problem we had $FF(G) < 2OPT(G)$.