ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 136 Nr kol. 1556

Piotr BOROWIECKI
Uniwersytet Zielonogórski

EFEKTYWNOŚĆ ALGORYTMÓW KOLOROWANIA GRAFÓW W TRYBIE ON-LINE (II)

Streszczenie. Z praktycznego punktu widzenia analiza najgorszego przypadku okazuje się często zbyt pesymistyczna, natomiast zachowanie algorytmu dla spotykanych w rzeczywistości danych jest najczęściej dużo lepsze niż dla stosunkowo nielicznych instancji, decydujących o złej efektywności w najgorszym przypadku. Wśród parametrów pozwalających na ocenę oczekiwanego zachowania algorytmu kolorowania w trybie on-line koncentrujemy się na podatności grafu. Definiujemy operację zachowującą podatność dla algorytmu `First-Fit`. Dla tego samego algorytmu rozstrzygamy również problem istnienia grafów o dowolnie małej podatności. Dla znanej z wielu zastosowań rodziny grafów przedziałów prezentowane są wnioski płynące z eksperymentalnego porównania efektywności dwóch znanych algorytmów kolorowania on-line.

EFFECTIVENESS OF ON-LINE GRAPH COLORING ALGORITHMS (II)

Summary. It usually happens that worst case analysis leads to the results which are too pessimistic to be valuable in real live applications. In this paper we investigate an expected effectiveness of on-line graph coloring algorithms, in particular the susceptibility of graphs to algorithm `First-Fit` is analyzed. The operation preserving susceptibilty is defined and an existence of graphs having arbitrarily low susceptibility to `First-Fit`, is proved. For well known and widely applicable family of interval graphs, the results of comparative experimental study of expected behavior for two on-line coloring algorithms are given.

1. Wprowadzenie

Praca ta jest kontynuacją problematyki poruszonej w [2, 3] i dotyczy efektywności algorytmów kolorowania wierzchołków grafów w trybie on-line. W obu pracach zaprezentowano wyniki dotyczące zarówno oceny efektywności pesymistycznej, jak i oczekiwanej. W niniejszej pracy koncentrujemy się głównie na wprowadzonej w [3] podatności grafu oraz eksperymentalnym porównaniu efektywności dwóch znanych algorytmów kolorowania on-line, `First-Fit` i `Scattered Coloring`. Rozważane w tej pracy grafy są skończone i nie zawierają pętli ani wielokrotnych krawędzi. Dla danego grafu $G = (V, E)$ przez $V(G)$ oznaczamy zbiór wierzchołków, natomiast $E(G)$ oznacza zbiór krawędzi grafu G, przy czym *rzędem* grafu G nazywamy liczbę n równą $|V(G)|$. *Złączeniem* $G_1 + G_2$ *grafów* G_1 i G_2 jest graf $H = (V, E)$ taki, że $V(H) = V(G_1) \cup V(G_2)$, $E(H) = E(G_1) \cup E(G_2) \cup \{uv : u \in V(G_1), v \in V(G_2)\}$. Problem kolorowania wierzchołków grafu polega na przypisaniu każdemu wierzchołkowi $v \in V(G)$ takiego koloru $c(v)$, aby dowolne dwa sąsiednie wierzchołki miały różne kolory, a liczba użytych kolorów była jak najmniejsza. Pokolorowanie wierzchołków grafu przy użyciu k kolorów nazywamy jego *k-pokolorowaniem*, a najmniejszą liczbę k, dla której istnieje k-pokolorowanie grafu G, nazywamy *liczbą chromatyczną* i oznaczamy $\chi(G)$.

W odróżnieniu od szeroko opisywanego w literaturze kolorowania off-line (patrz np. [5, 7]) podczas kolorowania w trybie on-line struktura kolorowanego grafu nie jest znana z góry, a kolejne wierzchołki grafu prezentowane są na wejściu algorytmu kolorującego w niezależnym od algorytmu porządku $\pi = (v_1, v_2, \ldots, v_n)$. W momencie prezentacji wierzchołka v_i ujawniany jest także zbiór krawędzi $E_i \subseteq E(G)$, łączących v_i z wybranymi spośród zaprezentowanych wcześniej wierzchołków $(v_1, \ldots, v_{i-1})$. Przez $\mathrm{N}_\pi(v_i)$ oznaczamy zbiór wierzchołków, które są sąsiadami wierzchołka v_i i poprzedzają go w π, natomiast $\mathrm{C}_\pi(v_i)$ jest zbiorem różnych kolorów przypisanych wierzchołkom ze zbioru $\mathrm{N}_\pi(v_i)$. W ramach tej pracy przyjmujemy, że prawdopodobieństwo wystąpienia każdej permutacji zbioru $V(G)$ jest jednakowe. Kolorowanie grafu G w trybie on-line można interpretować jako grę dwóch przeciwników nazywanych *prezenterem* i *malarzem*. Prezenter odsłania kolejno wierzchołki grafu wraz z odpowiednimi krawędziami. Malarz nadaje kolory prezentowanym przez prezentera wierzchołkom. Celem malarza jest użycie jak najmniejszej liczby kolorów, natomiast prezenter, przeciwstawiając się temu, szuka takiego uporządkowania wierzchołków, które zmusi malarza do użycia jak największej liczby kolorów. Malarz wygrywa grę, jeżeli do pokolorowania całego grafu użyje $\chi(G)$ kolorów. Jeżeli użyje ich więcej, przegrywa. Jednym z najbardziej znanych algorytmów kolorowania on-line jest zachłanny algorytm `First-Fit` (w skrócie FF), który realizuje strategię zachłanną, przyporządkowując każdemu wierzchołkowi możliwie najmniejszy kolor.

Liczbę kolorów użytych przez A do pokolorowania grafu G przy zadanej permutacji π oznaczamy przez $\mathtt{A}(G,\pi)$. Największą liczbę kolorów użytych przez algorytm A do pokolorowania grafu G, wśród wszystkich możliwych permutacji $V(G)$, oznaczamy $\chi_{\mathtt{A}}(G)$ i nazywamy *liczbą on-line chromatyczną grafu G dla algorytmu* A. Bardziej formalnie:

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

Liczba $\chi_{\mathtt{A}}(G)$ mówi o zachowaniu algorytmu A w najgorszym przypadku. Przy założeniu, że wystąpienie każdej prezentacji on-line grafu G jest jednakowo prawdopodobne, miarą oczekiwanego zachowania algorytmu A jest *przeciętna liczba chromatyczna grafu G dla algorytmu* A (ang. mean chromatic number), która zdefiniowana jest jako:

$$\chi^{s}_{\mathtt{A}}(G) = \frac{1}{n!}\sum \mathtt{A}(G,\pi),$$

gdzie sumowanie odbywa się po wszystkich permutacjach zbioru $V(G)$. Niech $n_s(G,\mathtt{A})$ oznacza liczbę permutacji zbioru $V(G)$, które powodują, że algorytm A daje optymalne pokolorowanie grafu G, tzn. $\mathtt{A}(G,\pi) = \chi(G)$. Przez $n_f(G,\mathtt{A}) = n! - n_s(G,\mathtt{A})$ oznaczamy liczbę permutacji prowadzących do pokolorowań nieoptymalnych. Współczynnik $p_{\mathtt{A}}(G)$ nazywamy *podatnością grafu G dla algorytmu* A i definiujemy następująco:

$$p_{\mathtt{A}}(G) = 1 - \frac{n_f(G,\mathtt{A})}{n!} = \frac{n_s(G,\mathtt{A})}{n!}$$

2. Eksperymentalne porównanie efektywności algorytmów SCC i FF

Z praktycznego punktu widzenia pesymistyczna ocena algorytmu jest zwykle niewystarczająca. Znane są przykłady algorytmów świetnie spisujących się w praktyce pomimo "negatywnej" oceny pesymistycznej. Podobnie dla algorytmów kolorowania on-line rodzi się naturalne pytanie o ich rzeczywistą (obserwowalną w praktyce) efektywność. Dlatego ocenie eksperymentalnej, przeprowadzonej dla znanej z szerokich zastosowań rodziny grafów przedziałów (ang. interval graphs), poddane zostały dwa najpopularniejsze algorytmy kolorowania on-line - algorytm FF i dedykowany dla grafów przedziałów algorytm SCC (ang. Scattered Coloring) [8]. Rodzina *grafów przedziałów* to grafy $G = (V,E)$ o tej własności, że dla każdego z nich istnieje zbiór $\mathcal{J} = \{I(v) : v \in V(G)\}$

przedziałów na osi rzeczywistej takich, że dla dwóch różnych przedziałów $I(v_i), I(v_j)$ zachodzi $I(v_i) \cap I(v_j) \neq \emptyset \Leftrightarrow v_iv_j \in E(G)$. Zbiór $\mathcal{J}$ nazywamy *reprezentacją przedziałową* grafu G. Algorytm SCC gwarantuje uzyskanie rozwiązań, dla których

$$\chi_{\mathrm{SCC}}(G) \leq 3\chi(G) - 2 \tag{1}$$

Ponieważ w pracy [6] udowodniono, że ograniczenia (1) nie można już poprawić, więc w sensie najgorszego przypadku algorytm ten jest najlepszy wśród wszystkich algorytmów kolorowania grafów przedziałów w trybie on-line. Przypomnijmy, że dla algorytmu FF w [9] podano następujące dolne oszacowanie:

$$\chi_{\mathrm{FF}}(G) \geq 4.45\chi(G) \tag{2}$$

2.1. Algortm SCC

Algorytm SCC przyporządkowuje każdy wierzchołek v_i do pewnego podzbioru $W_s \subseteq V(G)$ nazywanego *grupą*. Wewnątrz grupy W_s wierzchołek przypisywany jest do jednego z *poziomów* $P_p \subseteq W_s$. Przydział do poziomu odbywa się zgodnie z algorytmem FF, natomiast o przydziale do grupy W_s decyduje liczba klikowa ω podgrafu indukowanego przez $\bigcup_{j=0}^{s} W_j \cup \{v_i\}$. Kolor przypisany każdemu wierzchołkowi jest więc parą (s, p).

Algorytm [SCC - **Scattered Coloring**]

BEGIN

  INITIALIZE (W_j := $\emptyset$, $V(G)$:= $\emptyset$, $E(G)$:= $\emptyset$);

  REPEAT

    READ(v_i, E_i);

    $V(G)$:= $V(G) \cup \{v_i\}$;

    $E(G)$:= $E(G) \cup E_i$;

    s := 0;

    WHILE $\omega(G[(\bigcup_{j=0}^{s} W_j) \cup \{v_i\}]) > s + 1$ DO

      $s := s + 1$;

    v_i.*grupa* := s;

    v_i.*poziom* := First-Fit($G[W_s], v_i$);

    W_s := $W_s \cup \{v_i\}$;

  UNTIL koniec ciągu wierzchołków;

END.

W pracy [8] algorytm SCC zapisano dla danych wejściowych będących ciągiem przedziałów. Wersja prezentowana w tej pracy sformułowana jest dla grafu przedziałów, przy czym warto zauważyć, że liczbę klikową ω grafu przedziałów można wyznaczyć w czasie liniowym.

2.2. Wyniki eksperymentu

Na podstawie wyników eksperymentu można stwierdzić, że pomimo przeznaczenia algorytmu SCC dla grafów przedziałów w praktyce spisuje się on dla nich gorzej niż działający dla wszystkich grafów algorytm FF. Badania przeprowadzono dla dwóch typów instancji. W pierwszym eksperymencie każdy wygenerowany losowo graf przedziałów pokolorowany został algorytmami FF i SCC, dla wszystkich możliwych permutacji wierzchołków. Pozwoliło to na dokładne wyznaczenie przeciętnej liczby chromatycznej i podatności każdego z badanych grafów. Ze względu na dużą liczbę permutacji wierzchołków eksperyment można było przeprowadzić jedynie dla grafów o niskim rzędzie. Wśród przebadanych w eksperymencie losowych prób, po 40 grafów o $n = 5, 6, \ldots, 10$ wierzchołkach, nie napotkano takich, dla których przeciętna liczba chromatyczna dla algorytmu SCC byłaby mniejsza niż dla FF. W drugim eksperymencie algorytmy porównano w warunkach bardziej zbliżonych do rzeczywistych aplikacji. Każdy wygenerowany losowo graf pokolorowano tylko dla jednego, losowego uporządkowania wierzchołków. Dzięki tak wygenerowanym instancjom można było zaobserwować różnice w działaniu obu algorytmów również dla grafów o większej liczbie wierzchołków. Przebadano instancje, dla których grafy miały $n = 10, 20, \ldots, 100$ wierzchołków (po 20 tys dla każdego n). Dla $n = 10$ algorytm FF zwyciężył (użył mniej kolorów niż SCC) dla ok. 9% instancji i tyle samo kolorów co SCC dla pozostałych 91%. Jednak dla $n = 20$, zanotowano już ok. 36% zwycięstw FF i ok. 64% remisów. Dla $n = 40$ algorytm FF wygrywa dla ok. 88% instancji, remisuje dla ok.12% i przegrywa w jednym przypadku. Przy $n = 100$ zwycięstwa FF stają się niemal regułą – ponad 99%. Pewnym wytłumaczeniem przewagi algorytmu FF jest fakt, że grafy skonstruowane w celu wykazania dolnego ograniczenia (2) są grafami o wysokim rzędzie i mają bardzo specyficzną strukturę. Biorąc pod uwagę fakt, że użycie maksymalnej liczby kolorów będzie niezbędne tylko dla niewielkiej liczby permutacji zbioru wierzchołków widać, że prawdopodobieństwo losowego wygenerowania takiej instancji jest bardzo niewielkie. Dobra przeciętna efektywność FF znajduje potwierdzenie w wynikach zaprezentowanych w [3]. Rezultaty te stanowią z pewnością istotną motywację do poszukiwania kolejnych algorytmów dla grafów przedziałów oraz do dalszego badania własności algorytmu FF.

3. Podatność grafów

Niech $\mathcal{G}_{n,k} = \{G : |V(G)| = n, \chi(G) = k\}$ i $G_{n,k}$ niech będzie grafem realizującym najmniejszą podatność dla FF w rodzinie $\mathcal{G}_{n,k}$. W pracy [3], w której wprowadzono pojęcie podatności, podane zostały ogólne dolne oszacowania tego parametru oraz grafy o najmniejszych podatnościach dla algorytmu FF (najmniejszych wśród grafów o określonej liczbie wierzchołków). Były to grafy $G_{6,3}, G_{7,3}, G_{8,4}, G_{9,3}$. W tablicy 3 prezentujemy poszerzone zestawienie wartości najmniejszych podatności dla algorytmu FF, osiąganych dla wszystkich grafów z rodzin $\mathcal{G}_{n,k}$, $n = 6, \ldots, 9$. Dalsze rozszerzenie zestawienia podanego w tablicy 3 wymaga wygenerowania dla pewnego n katalogu wszystkich grafów nieizomorficznych i pokolorowania każdego z nich dla wszystkich $n!$ permutacji zbioru wierzchołków. Przykładowo, dla grafów 10-wierzchołkowych należałoby wykonać ok. $4.25 \cdot 10^{13}$ kolorowań.

Tablica 1

Najmniejsze podatności grafów dla algorytmu FF

$\chi(G)$	$n=6$	$n=7$	$n=8$	$n=9$
2	0.5000	0.4250	0.3177	0.2663
3	0.4083	0.3099	0.2031	0.1092
4	0.7500	0.4083	0.2017	0.1298
5	1.0000	0.7500	0.4083	0.2017
6	1.0000	1.0000	0.7500	0.4083
7	–	1.0000	1.0000	0.7500
8	–	–	1.0000	1.0000
9	–	–	–	1.0000

Na szczególną uwagę zasługuje wyróżniająca się podatność grafów 2-chromatycznych oraz rodziny grafów o jednakowych podatnościach, np. $P = \{G_{6,3}, G_{7,4}, G_{8,5}, G_{9,6}\}$.

Dokładna analiza struktury grafów z rodziny P pozwala zauważyć, że $G_{9,6} = G_{8,5} + K_1$, $G_{8,5} = G_{7,4} + K_1$ i $G_{7,4} = G_{6,3} + K_1$. Obserwacja ta skłania do postawienia hipotezy, że operacja złączenia grafu G z grafem pełnym K_n daje w wyniku graf o podatności równej $p_{FF}(G)$ (zauważmy, że wspomniana wcześniej rodzina grafów przedziałów jest domknięta ze względu na operację złączenia z grafem pełnym). Aby rozstrzygnąć tę hipotezę udowodnimy następujący lemat.

Lemat 1. *Jeżeli G jest dowolnym grafem i H jest grafem otrzymanym przez złączenie G z K_1, to $p_{FF}(H) = p_{FF}(G)$.*

Dowód. Niech $V(G) = \{v_1, v_2, \ldots, v_n\}$ i $V(K_1) = \{x\}$ oraz niech $c_\pi(v)$ oznacza kolor przypisany wierzchołkowi v przez algorytm FF w permutacji π. Udowodnimy, że $n_s(H, \mathrm{FF}) = (n+1) \cdot n_s(G, \mathrm{FF})$. Dla każdej permutacji $\pi = (v_1, v_2, \ldots, v_n)$ zbioru wierzchołków grafu G przeanalizujemy wszystkie otrzymane z niej permutacje $\pi_i = (v_1, v_2, \ldots, v_i, x, v_{i+1}, \ldots, v_n)$ zbioru $V(H)$. Należy zauważyć, że wszystkie permutacje zbioru $(n+1)$-elementowego można otrzymać z permutacji zbioru n-elementowego w powyższy sposób. Jeżeli w permutacji π graf G pokolorowany został k kolorami, to kolorując graf H w permutacji $\pi_0 = (x, v_1, v_2, \ldots, v_n)$, algorytm FF użyje $k+1$ kolorów, przypisując wierzchołkowi x kolor 1, a wszystkim pozostałym wierzchołkom, ze względu na sąsiedztwo z x, kolory $c_\pi(v_i) + 1$. Zauważmy również, że dla każdej permutacji $\pi_i = (v_1, v_2, \ldots, v_i, x, v_{i+1}, \ldots, v_n)$, $0 < i \leq n$, wierzchołki ciągu $D_{\pi_i} = \{v_1, \ldots, v_i\}$ otrzymają te same kolory co podczas kolorowania grafu G w permutacji π, natomiast wierzchołek x otrzyma kolor $t+1$, gdzie $t = \max_{j \leq i}\{c_\pi(v_j)\}$. Pozostaje przeanalizowanie kolorowania wierzchołków $v_{i+1}, \ldots, v_n$. Niech A_π, B_π będą ciągami zdefiniowanymi następująco:

$$A_\pi = (v_j : c_\pi(v_j) \leq t, j \geq i+1),$$
$$B_\pi = (v_j : c_\pi(v_j) > t, j \geq i+1)$$

Przypadek 1. Niech $v_{i+1} \in A_\pi$. Ponieważ wierzchołek v_{i+1} otrzymał w π kolor s, to wiemy, że $s \notin \mathrm{C}_\pi(v_{i+1})$, a ponieważ $\mathrm{N}_{\pi_i}(v_{i+1}) = \mathrm{N}_\pi(v_{i+1}) \cup \{x\}$ i x otrzymał kolor $t+1$, więc oczywiście w zbiorze $\mathrm{N}_\pi(v_{i+1})$ dalej brakuje wierzchołka w kolorze s i algorytm FF przypisze v_{i+1} ten sam kolor. Weźmy teraz dowolny wierzchołek v' z ciągu A_π i załóżmy, że wszystkim wierzchołkom z ciągu A_π, poprzedzającym v', algorytm FF nadał w π_i te same kolory co w π. Będziemy mogli indukcyjnie stwierdzić, że v' otrzyma ten sam kolor w obu permutacjach, jeżeli wykażemy, że każdy wierzchołek $u \in B_\pi$, poprzedzający v', otrzymał w π_i kolor $c_{\pi_i}(u) = c_\pi(u) + 1$.

Niech zatem v_j będzie pierwszym wierzchołkiem ciągu B_π. Wiadomo, że $c_\pi(v_j) = t+1$. Ponieważ kolory wierzchołków $v_1, \ldots, v_i$, i na podstawie poczynionego wyżej założenia wszystkich poprzedzających v_j ze zbioru A_π nie uległy w π_i zmianie, więc $\{c : 1 \leq c \leq t\} \subseteq C_{\pi_i}(v_j)$. Ponadto x ma kolor $t+1$ i jest sąsiadem v_j, zatem FF przypisze v_j kolor $t+2$. Weźmy zatem wierzchołek $v'' \in B_\pi$ o kolorze $c_\pi(v'') = r$ bezpośrednio poprzedzający v' i załóżmy, że wszystkie wierzchołki ciągu B_π, poprzedzające v'', otrzymały w π_i kolor o jeden większy niż w π. Widzimy, że $C_{\pi_i}(v'')$ zawiera wszystkie kolory $t+2, \ldots, r$. Ponadto z poprzedniego założenia (dla A_π) i pokolorowania wierzchołków zbioru D_{π_i} wynika, że $C_{\pi_i}(v'')$ zawiera także kolory $1, \ldots, t$, co w połączeniu z kolorem $t+1$ przypisanym wierzchołkowi x daje kolory $1, \ldots, r$. Stąd indukcyjnie wnioskujemy, że algorytm FF w permutacji π_i przypisze wierzchołkowi v'' kolor $r+1$.

Przypadek 2. Jeżeli $v_{i+1} \in B_\pi$, to dowód przebiega analogicznie do przypadku 1.

Zatem, jeżeli graf G w permutacji π pokolorowany został k kolorami, to do pokolorowania grafu H w każdej permutacji π_i, otrzymanej z π, niezbędne jest $k+1$ kolorów. Ostatecznie, ponieważ $\chi(H) = \chi(G) + 1$, więc każda permutacja π_i spośród $n+1$ permutacji otrzymanych z permutacji π prowadzi do optymalnego pokolorowania H wtedy i tylko wtedy, gdy π prowadzi do optymalnego pokolorowania grafu G. ∎

Powyższy lemat pozwala na udowodnienie następującego twierdzenia.

Twierdzenie 1. *Dla każdego grafu G i $n \geq 1$ zachodzi* $p_{\mathrm{FF}}(G + K_n) = p_{\mathrm{FF}}(G)$.

Dowód. Jeżeli $n = 1$, to twierdzenie jest prawdziwe na podstawie lematu 1. Załóżmy, że $p_{\mathrm{FF}}(G + K_{n-1}) = p_{\mathrm{FF}}(G)$. Ponieważ $K_n = K_{n-1} + K_1$, więc korzystając z lematu 1 i założenia indukcyjnego otrzymujemy $p_{\mathrm{FF}}(G + K_n) = p_{\mathrm{FF}}((G + K_{n-1}) + K_1) = p_{\mathrm{FF}}(G + K_{n-1}) = p_{\mathrm{FF}}(G)$. ∎

Wniosek 1. *Dla każdego grafu G istnieje nieskończenie wiele grafów G_i, $i = 1, 2, \ldots$ takich, że* $p_{\mathrm{FF}}(G_i) = p_{\mathrm{FF}}(G)$.

Można wskazać związek pomiędzy podatnością grafu G dla algorytmu kolorowania on-line A oraz liczbami $\chi^s_A(G)$, $\chi_A(G)$ i $\chi(G)$. Zależność

$$\chi^s_A(G) \leq \chi_A(G) - p_A(G)(\chi_A(G) - \chi(G)) \tag{3}$$

otrzymujemy wychodząc z definicji przeciętnej liczby chromatycznej.

Dzięki powyższej nierówności oraz twierdzeniu podanemu w pracy [1], można rozstrzygnąć problem istnienia grafów o dowolnie małej podatności dla algorytmu FF. Mówi o tym następujące twierdzenie.

Twierdzenie 2. *Dla każdego $\epsilon > 0$ istnieje graf G taki, że $p_{FF}(G) < \epsilon$.*

Dowód. Niech C_{2k} będzie cyklem o parzystej długości $n = 2k$, $k = 1, 2, \ldots$. Ponieważ $\chi(C_{2k}) = 2$ i $\chi_{FF}(C_{2k}) = 3$, więc z (3) mamy

$$p_{FF}(C_{2k}) \leq 3 - \chi^s_{FF}(C_{2k})$$

Z [1] wiemy, że przy k dążącym do nieskończoności $3 - \chi^s_{FF}(C_{2k}) \sim 2/\alpha^{2k}$, gdzie α jest dodatnim rzeczywistym rozwiązaniem równania $\cosh x = x \sinh x$ i w przybliżeniu wynosi 1.9968. Dla każdego $\epsilon > 0$ istnieje k takie, że $2/\alpha^{2k} < \epsilon$. ∎

Stosując rekurencyjnie operację złączenia grafu G z grafem K_1, otrzymujemy $((\ldots(G+K_1)+K_1)+\ldots+K_1)$, czyli kolejne grafy o tej samej podatności dla FF, lecz o coraz większej gęstości $d(G)$, gdzie $d(G) = |E(G)|/\binom{|V(G)|}{2}$. Wynika stąd następujący wniosek.

Wniosek 2. *Dla każdego $\epsilon > 0$ i $r \in (0,1)$ istnieje graf G taki, że $p_{FF}(G) < \epsilon$ i $d(G) > r$.*

Podane twierdzenia pozwalają na wyciągnięcie istotnych wniosków dotyczących projektowania algorytmów przydziału zasobów, wykorzystujących kolorowanie wierzchołków grafu konfliktów algorytmem FF. Jeżeli graf G jest tworzonym na bieżąco grafem konfliktów zasobowych, kolorowanym w trybie on-line, to dla algorytmu FF prawdopodobieństwo otrzymania optymalnego rozwiązania może być bardzo małe, nawet wtedy, gdy gęstość grafu konfliktów jest zbliżona do gęstości grafu pełnego (przypomnijmy, że $p_{FF}(K_n) = 1$). Z drugiej strony, potrafimy podać grafy o małej gęstości (np. cykle) i dowolnie małej podatności. Stąd wniosek, że metody wstępnego przetwarzania żądań przydziału zasobów w trybie on-line (np. odrzucanie wybranych żądań, replikacja pewnych zasobów), w których *jedynym* kryterium jest minimalizacja liczby występujących konfliktów (liczby krawędzi), mogą być nieefektywne.

LITERATURA

1. Anthony M., Biggs N.: The mean chromatic number of paths and cycles, Discrete Math. **120** (1993) s. 227–231.
2. Borowiecki P.: Kolorowanie grafów w trybie on-line, Zeszyty Naukowe Polit. Śl., ser. Automatyka, z. 123, Gliwice, 1998, s. 65–75.
3. Borowiecki P.: Efektywność algorytmów kolorowania grafów w trybie on-line, Zeszyty Naukowe Polit. Śl., ser. Automatyka, z. 131, Gliwice 2000, s. 12–23.
4. Diestel R.: Graph Theory, Springer-Verlag, 1997.
5. Jensen T.R., Toft B.: Graph Coloring Problems, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, 1995.
6. Kierstead H.A., Trotter W.T.: An extremal problem in recursive combinatorics, Congressus Numerantium **33** (1981) s. 143–153.
7. Kubale M.: Problem kolorowania wierzchołków grafów. Przegląd algorytmów i zastosowań, Zeszyty Naukowe Polit. Śl., ser. Automatyka, z. 114, Gliwice 1994, s. 187–198.
8. Ślusarek M.: A coloring algorithm for interval graphs, Mathematical Foundations of Computer Science '89, LNCS 379, Springer, 1989, s. 471–480.
9. Ślusarek M.: A Lower Bound for the First-Fit Coloring of Interval Graphs, Zeszyty Naukowe Uniwersytetu Jagiellońskiego, Prace Informatyczne z. 5, Kraków 1993, s. 25–32.

Recenzent: Prof. dr hab. inż. Jerzy Klamka

Abstract

An on-line algorithm receives a sequence of requests and immediately services each request before the next one is considered. It is assumed that the sequence is not known in advance. Worst case analysis of on-line graph coloring algorithms is usually based on "bad" sequences. However, in many real-life applications, worst case analysis leads to the results which are too pessimistic to be valuable. In this paper we investigate an expected behavior of graph coloring algorithms under assumption that each sequence is equally likely. In particular, the susceptibility of graphs to algorithm `First-Fit` is analyzed. We define the operation preserving susceptibilty and prove the existence of graphs having arbitrarily low susceptibility to algorithm `First-Fit`. The experimental comparative study results of expected behavior of algoritms `First-Fit` and `Scattered Coloring` for interval graphs are given, too.