

Piotr BOROWIECKI  
Politechnika Zielonogórska

## EFEKTYWNOŚĆ ALGORYTMÓW KOLOROWANIA GRAFÓW W TRYBIE ON-LINE

**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. W niniejszej pracy rozważamy parametry, które pozwalają na ocenę oczekiwanego zachowania algorytmów kolorowania wierzchołków grafów w trybie on-line. W szczególności koncentrujemy się na oszacowaniach przeciętnej liczby chromatycznej, przeciętnej dobroci, funkcji przeciętnej dobroci i nowo zdefiniowanej podatności grafu dla algorytmu A. Podajemy rezultaty teoretyczne oraz wyniki eksperymentów obliczeniowych przeprowadzonych dla algorytmu First-Fit.

## EFFECTIVENESS OF ON-LINE GRAPH COLORING ALGORITHMS

**Summary.** It usually happens that worst case analysis leads to the results which are too pessimistic to be valuable in real life applications. In this paper we investigate an expected effectiveness of on-line graph coloring algorithms. The mean chromatic number, mean performance ratio and mean performance function as well as newly defined susceptibility of graph to algorithm A are analyzed. Experimental results for algorithm First-Fit are presented, too.

### 1. Wprowadzenie

Będziemy rozważać grafy bez pętli i 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  $|V(G)|$ . Przez  $N(v)$  oznaczamy zbiór wszystkich wierzchołków sąsiednich z wierzchołkiem  $v$ . Definicje innych pojęć z zakresu teorii grafów można znaleźć w [8]. W najbardziej tradycyjnej postaci *kolorowanie wierzchołków grafu* polega na przypisaniu każdemu wierzchołkowi takiego

koloru, aby dowolne dwa sąsiednie wierzchołki miały różne kolory. 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 grafu  $G$  i oznaczamy  $\chi(G)$ . Graf, dla którego istnieje  $k$ -pokolorowanie, nazywamy  $k$ -kolorowalnym, a graf o liczbie chromatycznej równej  $k$  nazywamy  $k$ -chromatycznym. Dowolne  $k$ -pokolorowanie grafu  $G$  tworzy podział  $(V_1, V_2, \dots, V_k)$  zbioru wierzchołków  $V(G)$  na  $k$  podzbiorów niezależnych, nazywanych klasami kolorów. Problem kolorowania wierzchołków grafu polega na wyznaczeniu podziału zbioru  $V(G)$  na najmniejszą liczbę klas kolorów. Większość znanych rezultatów dotyczy przypadku, w którym struktura kolorowanego grafu dana jest z góry. Do rozwiązania problemu (najczęściej tylko przybliżonego) można wtedy stosować algorytmy kolorowania działające w trybie off-line (patrz np. [13, 14]). Jednak jak pokazuje praktyka, duża grupa problemów występujących w rzeczywistości ma naturę nie pozwalającą na przyjęcie założenia o znajomości z góry struktury całego grafu. W rozwiązywaniu tego typu problemów każdy algorytm wymagający dostępu do całego grafu posiadałby poważnie ograniczoną przydatność praktyczną. Algorytmami budującymi rozwiązanie problemu oparte na niepełnej informacji o grafie są algorytmy działające w trybie on-line. Podczas kolorowania wierzchołków w trybie on-line, kolejne wierzchołki grafu prezentowane są na wejściu algorytmu kolorującego w niezależnym od algorytmu porządku  $(v_1, v_2, \dots, v_n)$ . 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  $v_1, \dots, v_{i-1}$ . Para  $\sigma_i = (v_i, E_i)$  stanowi *żądanie* pokolorowania wierzchołka  $v_i$ . Algorytm obsługujący żądanie musi nieodwracalnie, przed prezentacją następnego wierzchołka, przyporządkować prezentowanemu wierzchołkowi kolor  $c(v_i)$ . Graf  $G$  wraz z zadaniem uporządkowaniem  $\pi$  zbioru wierzchołków nazywamy *prezentacją on-line grafu  $G$*  i oznaczamy  $(G, \pi)$ .

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, podczas gdy 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.

Algorytm [ First-Fit ]

BEGIN

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

    REPEAT

        READ( $\sigma_i(v_i, E_i)$ );

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

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

$k := 1$ ;

        WHILE  $v_i.kolor$  nie jest przypisany DO

            IF nie istnieje  $u \in N(v_i)$  taki, że  $c(u) = k$

                THEN  $v_i.kolor := k$ ;

                ELSE  $k := k + 1$ ;

    UNTIL koniec ciągu  $\sigma$ ;

END.

Przykład 1. Rozważmy ciąg żądań  $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ , gdzie

$$\sigma_1 = (v_1, \emptyset), \sigma_2 = (v_4, \emptyset), \sigma_3 = (v_2, \{v_1 v_2\}), \sigma_4 = (v_3, \{v_3 v_2, v_3 v_4\}).$$

Jeżeli malarz koloruje wierzchołki zgodnie ze strategią stosowaną przez algorytm FF, to obsługa żądań z ciągu  $\sigma$  prowadzi do następującego pokolorowania:  $c(v_1) = 1, c(v_4) = 1, c(v_2) = 2, c(v_3) = 3$ . Zaprezentowany przez prezentera graf jest ścieżką  $P_4$ . Malarz przegrywa, ponieważ użył trzech kolorów, podczas gdy optymalna liczba kolorów wynosi  $\chi(P_4) = 2$ .

Liczbę kolorów użytych przez A do pokolorowania grafu  $G$  przy zadanym uporządkowaniu  $\pi$  oznaczamy przez  $A(G, \pi)$ . Największą liczbę kolorów użytych przez algorytm A do pokolorowania grafu  $G$ , wśród wszystkich możliwych prezentacji on-line

grafu  $G$ , oznaczamy  $\chi_A(G)$  i nazywamy liczbą on-line chromatyczną grafu  $G$  dla algorytmu  $A$ . Bardziej formalnie:

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

Liczba  $\chi_A(G)$  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), którą definiujemy jako:

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

gdzie sumowanie odbywa się po wszystkich uporządkowaniach  $\pi$  zbioru  $V(G)$ .

## 2. Ocena pesymistycznej efektywności kolorowania on-line

Ocena efektywności algorytmów kolorowania on-line opiera się zwykle na pojęciu *funkcji dobroci*  $\rho_A : Z^+ \rightarrow R$  mówiącej, jak bardzo rozwiązania generowane przez algorytm kolorowania on-line  $A$  mogą odbiegać od rozwiązania optymalnego:

$$\rho_A(n) = \max \chi_A(G) / \chi(G),$$

gdzie  $Z^+$ ,  $R$  oznaczają odpowiednio zbiór liczb całkowitych dodatnich i rzeczywistych, a maksimum brane jest po wszystkich grafach o  $n$  wierzchołkach. Jedno z najlepszych dolnych oszacowań funkcji dobroci wynika z twierdzenia podanego w pracy [7]. Dla modelu kolorowania rozważanego w niniejszej pracy oszacowanie to pokrywa się z wynikiem podanym wcześniej w [4] i wynosi

$$\rho_A(n) > \frac{\log_2 n}{2}.$$

Interesujący rezultat z pogranicza oceny pesymistycznej podano dla algorytmu FF w pracy [10]. Przy  $n$  dążącym do nieskończoności prawdopodobieństwo tego, że

$$\chi_{FF}(G) \leq (2 + \epsilon)\chi(G), \quad (1)$$

dąży do 1, innymi słowy własność ta zachodzi dla *prawie wszystkich grafów*. Najlepszy do tej pory algorytm kolorowania on-line dowolnych grafów (LST) znajdujemy w pracy [15], natomiast wyczerpujący przegląd znanych wyników zawarty jest w pracach [5, 12].

### 3. Ocena oczekiwanej efektywności kolorowania on-line

Przeciętna liczba chromatyczna stanowi podstawę do zdefiniowania przeciętnej dobroci i funkcji przeciętnej dobroci. *Przeciętną dobroć algorytmu A dla grafu G* definiujemy następująco:

$$r_A^s(G) = \frac{\chi_A^s(G)}{\chi(G)}.$$

*Funkcja przeciętnej dobroci*  $\rho_A^s(n) : Z^+ \rightarrow R$  zdefiniowana jest jako:

$$\rho_A^s(n) = \max r_A^s(G),$$

gdzie maksimum brane jest po wszystkich grafach o  $n$  wierzchołkach. Podobnie jak dla  $\rho_A(n)$ , dla dowolnego algorytmu A mamy  $1 \leq \rho_A^s(n) \leq n$ .

Teoretyczne oszacowania przeciętnej liczby chromatycznej znane są jedynie dla bardzo specyficznych rodzin grafów. Anthony i Biggs [2] wykazali, że dla ścieżek i cykli o parzystej długości  $\chi_A^s$  zmierza do 3 przy  $n$  dążącym do nieskończoności. Nie są natomiast znane żadne ogólne oszacowania funkcji  $\rho_A^s(n)$ . Fakt ten stanowił motywację do przeprowadzenia eksperymentu obliczeniowego, którego głównym celem było wyznaczenie dokładnej wartości funkcji  $\rho_A^s(n)$  w możliwie szerokim zakresie wartości  $n$ . Eksperyment przeprowadzony został na komputerze IBM PC Pentium 166 MHz, a wyczerpującemu kolorowaniu algorytmem FF poddano wszystkie nieizomorficzne i spójne grafy o  $n$  wierzchołkach,  $3 \leq n \leq 9$ . Jednym z podstawowych problemów napotkanych podczas planowania eksperymentu była niedostępność katalogu wszystkich nieizomorficznych grafów. Problem efektywnego wygenerowania takiego katalogu nie jest z pewnością trywialny, chociażby ze względu na brak wielomianowych algorytmów i ostatecznego rozstrzygnięcia, czy problem sprawdzania izomorfizmu dwóch grafów jest NP-zupełny (patrz [9] problem GT48). Duży problem sprawia także liczba grafów, rosnąca wykładniczo ze wzrostem liczby wierzchołków, co łącznie z czasem kolorowania stanowiło ograniczenie decydujące ostatecznie o zawężeniu eksperymentu wyczerpującego kolorowania do  $n \leq 9$ .

#### 3.1. Generowanie grafów

Do wygenerowania zbioru nieizomorficznych, spójnych grafów zastosowano program makeg z pakietu NAUTY ver. 1.9, który ma obecnie najlepsze referencje wśród

znanych implementacji algorytmów generowania katalogów grafów nieizomorficznych. Do badań naukowych pakiet dostępny jest bezpłatnie w internecie pod adresem <http://cs.anu.edu.au/people/bdm>. Generowane grafy pogrupowane zostały według liczby wierzchołków, tworząc rodziny  $\mathcal{G}_n$ ,  $n = 3, \dots, 10$ . Liczby wygenerowanych grafów dla poszczególnych rodzin zestawiono w tab. 1. Dla porównania, podano także liczebność rodzin  $\mathcal{N}_n$  wszystkich grafów nieizomorficznych (włączając grafy niespójne).

Tablica 1  
Ilość nieizomorficznych grafów o  $n$   
wierzchołkach i czasy generowania

| $n$ | $ \mathcal{G}_n $ | $t$ [sek.] | $ \mathcal{N}_n $ |
|-----|-------------------|------------|-------------------|
| 3   | 2                 | –          | 4                 |
| 4   | 6                 | –          | 11                |
| 5   | 21                | –          | 34                |
| 6   | 112               | –          | 156               |
| 7   | 853               | 0.11       | 1044              |
| 8   | 11117             | 0.82       | 12346             |
| 9   | 261080            | 17.46      | 274668            |
| 10  | 11716571          | 705.96     | 12005168          |

Dzięki bardzo dużej sprawności pakietu NAUTY, wygenerowanie katalogu wszystkich niezbędnych grafów ( $n = 1, \dots, 10$ ) na wspomnianym komputerze zajęło zaledwie około 12 min. Jednak jak wynika z oszacowań, wygenerowanie katalogu nieizomorficznych, spójnych grafów 11-wierzchołkowych zajęłoby już około 20 godz. Dla porównania podajemy, że samych grafów 24-krawędziowych byłoby wtedy 68 803 769.

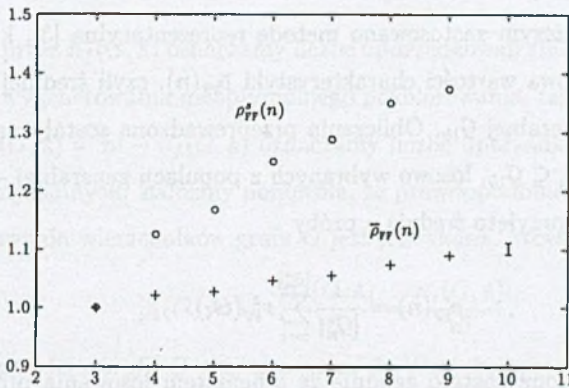
### 3.2. Kolorowanie grafów rzędu $n \leq 9$

W eksperymencie zastosowano metodę *wyczerpującego kolorowania* każdego grafu, tzn. generowano pokolorowanie grafu dla jego każdej prezentacji on-line. Pozwoliło to na wyznaczenie  $\Sigma \text{FF}(G, \pi)$ , a w konsekwencji obliczenie dokładnej wartości liczby  $\chi_{\text{FF}}^*(G)$  dla każdego z rozpatrywanych grafów. Ponieważ dla dowolnego grafu, wśród wszystkich jego prezentacji on-line, zawsze istnieje taka, która prowadzi do wygenerowania przez FF pokolorowania optymalnego, więc wyniki wyczerpującego kolorowania umożliwiły wyznaczenie liczby  $\chi(G)$  każdego grafu. W ten sposób wyznaczenie  $\rho_{\text{FF}}^*(n)$  sprowadzone zostało do przeszukania zbiorów wyników kolorowania i odszukania w nich grafów o największej wartości  $\tau_{\text{FF}}^*(G)$ . Jako uzupełnienie obliczone zostały wartości *funkcji średniej dobroci*  $\bar{\rho}_{\text{FF}}(n)$ , zde-

finiowanej jako:

$$\bar{\rho}_{FF}(n) = \frac{\sum \tau_{FF}^s(G)}{|\mathcal{G}_n|}.$$

Algorytm kolorowania FF zaimplementowano w języku C++. Już wstępne oszacowanie czasu niezbędnego do przeprowadzenia obliczeń wykazało, że przeprowadzenie eksperymentu, polegającego na wyczerpującym kolorowaniu każdego grafu jest realne jedynie dla  $n \leq 9$ . Wyczerpujące pokolorowanie wszystkich grafów dla  $n = 1, \dots, 8$  zajęło zaledwie półtorej godziny, jednak pokolorowanie grafów 9-wierzchołkowych wymagało już około 260 godzin, tj. prawie 11 dni.



Rys. 1. Funkcje  $\rho_{FF}^s(n)$  i  $\bar{\rho}_{FF}(n)$  oraz przedział ufności dla  $\bar{\rho}_{FF}(10)$

Fig. 1. Functions  $\rho_{FF}^s(n)$  and  $\bar{\rho}_{FF}(n)$  and confidence interval for  $\bar{\rho}_{FF}(10)$

Tablica 2

Wyniki eksperymentu dla  $n \leq 9$

| $n$ | $ \mathcal{G}_n $ | $\rho_{FF}^s(n)$ | $\tau_{FF}^s(G) > 1$ [szt.] | $\tau_{FF}^s(G) > 1$ [%] | $\bar{\rho}_{FF}(n)$ |
|-----|-------------------|------------------|-----------------------------|--------------------------|----------------------|
| 3   | 2                 | 1.0000           | 0                           | 0.00                     | 1.0000               |
| 4   | 6                 | 1.1250           | 1                           | 16.67                    | 1.0208               |
| 5   | 21                | 1.1667           | 5                           | 23.81                    | 1.0265               |
| 6   | 112               | 1.2500           | 51                          | 45.54                    | 1.0455               |
| 7   | 853               | 1.2875           | 531                         | 62.25                    | 1.0553               |
| 8   | 11117             | 1.3490           | 8882                        | 79.90                    | 1.0739               |
| 9   | 261080            | 1.3730           | 239606                      | 91.78                    | 1.0903               |

Rezultaty badań otrzymane w wyniku wyczerpującego kolorowania zestawione są na rys. 1 i w tab. 2. Zauważmy, że na podstawie nierówności 1, dla prawie wszystkich grafów mamy  $r_{FF}^*(G) \leq 2 + \epsilon$ . W następnym punkcie przedstawiamy wyniki estymacji przedziałowej wartości  $\bar{\rho}_{FF}(10)$ , która przeprowadzona została na podstawie losowo wybranej próby.

### 3.3. Kolorowanie grafów rzędu $n=10$

Wygenerowanie wyczerpujących pokolorowań wszystkich grafów 10-wierzchołkowych na wykorzystanym w tym eksperymencie komputerze zajęłoby około 10 lat. Jest oczywiste, że koszty przeprowadzenia takiego eksperymentu są zbyt wysokie, a skrócenie czasu przez zwiększenie mocy obliczeniowej komputera czy rozproszenie obliczeń na kilka równoległe pracujących stacji roboczych nie stanowi rozwiązania problemu.

W związku z powyższym zastosowano metodę reprezentacyjną [3], której celem była estymacja przedziałowa wartości charakterystyki  $\bar{\rho}_{FF}(n)$ , czyli średniej wartości cechy  $r_{FF}^*(G)$  w populacji generalnej  $\mathcal{G}_{10}$ . Obliczenia przeprowadzone zostały na próbie grafów 10-wierzchołkowych  $\mathcal{G}_{10}^* \subseteq \mathcal{G}_{10}$ , losowo wybranych z populacji generalnej. Jako estymator charakterystyki  $\bar{\rho}_{FF}(n)$  przyjęto średnią z próby

$$\rho_{FF}^*(n) = \frac{1}{|\mathcal{G}_n^*|} \sum_{i=1}^{|\mathcal{G}_n^*|} r_{FF}^*(G_i).$$

Losowanie przeprowadzone zostało zgodnie ze schematem losowania prostego, tzn. kolejne grafy wybierane były do próby sekwencyjnie, bez zwracania i z zachowaniem warunku, że każdy pozostawiony w populacji graf, w każdym momencie losowania ma to samo prawdopodobieństwo wylosowania. Wymagana liczebność próby wyznaczona została na podstawie próby wstępnej, przy ustalonym  $\alpha$ , będącym prawdopodobieństwem zdarzenia polegającego na tym, że  $|\bar{\rho}_{FF}(n) - \rho_{FF}^*(n)|$  jest większe od pewnej ustalonej wartości  $d$ . Jest to jeden z najbardziej obiektywnych i niezawodnych sposobów oceny minimalnej liczebności próby. W tab. 3 przedstawiamy wyniki otrzymane dla grafów 10-wierzchołkowych oraz wyniki otrzymane podczas weryfikacji przyjętej metody, którą przeprowadzono dla populacji  $\mathcal{G}_7$ ,  $\mathcal{G}_8$  i  $\mathcal{G}_9$ .

Przedział ufności dla  $n = 10$  zaznaczony jest również na rys. 1. Zauważmy, że eksperyment dla  $n = 9$  przeprowadzony został dla wartości  $d = 0.004$ . Jednak wymagana w tym przypadku liczebność próby okazała się znacznie większa niż w pozostałych przypadkach.



Stąd dla  $n = 10$  badania przeprowadzone zostały przy  $d = 0.01$ .

Tablica 3

Wyniki eksperymentu uzyskane metodą  
reprezentacyjną dla  $n = 7, \dots, 10$

| $\alpha = 0.05$ |                     |       |                   |                   |
|-----------------|---------------------|-------|-------------------|-------------------|
| $n$             | $ \mathcal{G}_n^* $ | $d$   | $\bar{P}_{FF}(n)$ | przedział ufności |
| 7               | 179                 | 0.01  | 1.0553            | 1.0425–1.0625     |
| 8               | 231                 | 0.01  | 1.0739            | 1.0627–1.0817     |
| 9               | 1511                | 0.004 | 1.0903            | 1.0865–1.0945     |
| 10              | 235                 | 0.01  | –                 | 1.0921–1.1121     |

#### 4. Grafy niepodatne dla algorytmu First-Fit

Dla grafu  $G$  przez  $n_f(G, A)$  oznaczamy liczbę uporządkowań zbioru  $V(G)$  zmuszających algorytm  $A$  do wygenerowania nieoptymalnego pokolorowania, tzn. takiego, że  $A(G, \pi) > \chi(G)$ . Przez  $n_s(G, A) = n! - n_f(G, A)$  oznaczamy liczbę uporządkowań prowadzących do pokolorowań optymalnych. Załóżmy ponownie, że prawdopodobieństwo wystąpienia każdego uporządkowania wierzchołków grafu  $G$  jest jednakowe. Wówczas współczynnik

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

nazywamy *podatnością grafu  $G$  dla algorytmu  $A$* . Wartość tego współczynnika mówi o prawdopodobieństwie wygenerowania przez algorytm  $A$  optymalnego pokolorowania grafu  $G$ .

Rozważmy algorytm FF. Niech  $k = \chi(G)$ , a podział  $(V_1, V_2, \dots, V_k)$  niech będzie optymalnym  $k$ -pokolorowaniem jednoznacznie kolorowalnego grafu  $G$  o  $n$  wierzchołkach. Oznaczając  $a_i = |V_i|$ ,  $i = 1, \dots, k$ , dolne oszacowanie podatności grafu  $G$  możemy zapisać następująco:

$$p_{FF}(G) \geq \frac{k! \cdot a_1! \cdot \dots \cdot a_k!}{n!}.$$

Zauważmy, że ponieważ  $\sum_{i=1}^k a_i = n$ , więc wartość powyższego wyrażenia jest najmniejsza, gdy  $|a_i - a_j| \leq 1$ . Otrzymujemy stąd następujący lemat:

**Lemat 1.** *Jeżeli  $G$  jest grafem  $n$ -wierzchołkowym, jednoznacznie  $k$ -chromatycznym, to  $p_{FF}(G) \geq f(n, k)$ , gdzie*

$$f(n, k) = \frac{k! \cdot k \cdot \lfloor \frac{n}{k} \rfloor!}{n!}.$$

Niech  $G$  będzie grafem, który jest  $k$ -chromatyczny i ma  $r$  różnych  $k$ -pokolorowań  $(V_1^{(j)}, V_2^{(j)}, \dots, V_k^{(j)})$ ,  $j = 1, \dots, r$ . Oznaczmy  $a_i^{(j)} = |V_i^{(j)}|$ . Wtedy podatność

$$p_{\text{FF}}(G) \geq k! \cdot \sum_{j=1}^r \frac{a_1^{(j)}! \cdot \dots \cdot a_k^{(j)}!}{n!}.$$

Stąd ostatecznie możemy zapisać twierdzenie:

**Twierdzenie 1.** *Jeżeli  $G$  jest grafem  $n$ -wierzchołkowym,  $k$ -chromatycznym, posiadającym  $r$  różnych, optymalnych pokolorowań, to*

$$p_{\text{FF}}(G) \geq r \cdot f(n, k).$$

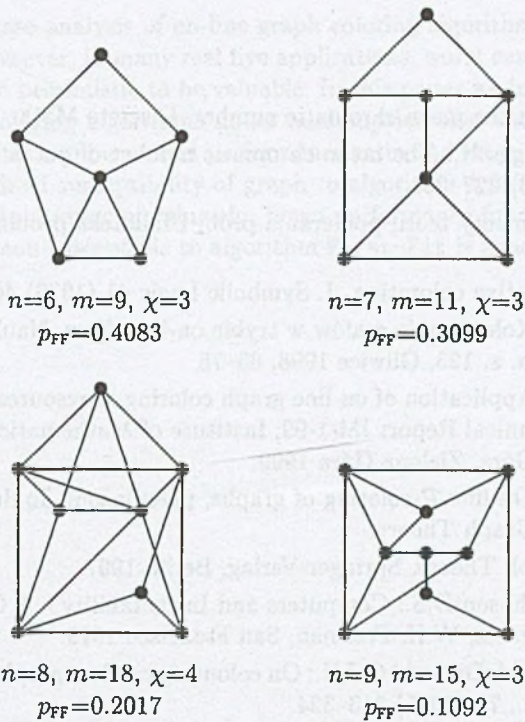
#### 4.1. Generowanie katalogu grafów niepodatnych

Jeżeli wystąpienie każdego uporządkowania wierzchołków grafu  $G$  jest jednakowo prawdopodobne i wartość współczynnika podatności  $p_A(G) < 0.5$ , to  $G$  nazywamy *grafem niepodatnym dla algorytmu A*. Wyniki wyczerpujących pokolorowań grafów wykorzystane zostały do wygenerowania katalogu grafów niepodatnych dla algorytmu FF. Zestawienie ilości  $h$  wyznaczonych grafów, wraz z najmniejszą wartością podatności, dla każdego  $n = 6, \dots, 9$  podajemy w tab. 4. Warto w tym miejscu zapytać o istnienie grafów o podatności równej zero. Problem ten pozostaje w ścisłym związku z *grafami trudnymi do kolorowania dla algorytmu A* (ang. hard to color), zdefiniowanymi w pracy [11] jako grafy, dla których dowolna implementacja algorytmu A prowadzi do wygenerowania nieoptymalnego pokolorowania. Jak widać, są to grafy o zerowej podatności.

Tablica 4  
Ilości grafów niepodatnych i  
najmniejsze podatności  $n = 6, \dots, 9$

| $n$ | $h$ [szt.] | $h$ [%] | $\min_{G \in \mathcal{G}_n} p_{\text{FF}}(G)$ |
|-----|------------|---------|---|
| 6   | 2          | 1.79    | 0.4083  |
| 7   | 58         | 6.80    | 0.3099  |
| 8   | 1673       | 15.05   | 0.2017  |
| 9   | 62355      | 23.88   | 0.1092  |

W pracy [16] udowodniono, że najmniejszym (w sensie liczby wierzchołków i krawędzi) trudnym do kolorowania grafem dla algorytmu LST jest graf  $2K_{17}$ . Natomiast dla algorytmu FF wiadomo, że dla każdego grafu można podać przynajmniej jedną prezentację



Rys. 2. Grafy o najmniejszych podatnościach dla algorytmu FF

Fig. 2. Graphs with minimum susceptibility for algorithm FF

on-line prowadzącą do pokolorowania optymalnego. Stąd wynika, że nie istnieje graf trudny do kolorowania dla algorytmu FF, a wartość  $\min_{G \in \mathcal{G}_n} p_{FF}(G)$  nigdy nie osiągnie 0. Pomimo tego z wyników przedstawionych w tabeli 4 widać, że dla algorytmu FF istnieje wiele grafów niepodatnych, a więc z praktycznego punktu widzenia niemal równie kłopotliwych co grafy trudne do kolorowania dla FF. Z przeprowadzonych badań wynika również, że dla  $n = 1, \dots, 5$  zachodzi następujące twierdzenie:

**Twierdzenie 2.** *Jeżeli  $|V(G)| \leq 5$ , to  $p_{FF}(G) \geq 0.5$ .*

Zatem dla wszystkich grafów o liczbie wierzchołków nie przekraczającej 5, bardziej prawdopodobne jest, że algorytm FF wygeneruje pokolorowanie optymalne niż nieoptymalne. Dla każdej rodziny  $\mathcal{G}_n$ ,  $n = 6, \dots, 9$  wyznaczony został również graf o najmniejszej podatności dla FF. Wyznaczone grafy przedstawiamy na rys. 2. Grafy o niskim współczynniku podatności są szczególnie interesujące z punktu widzenia zastosowań, w których ten sam graf jest wielokrotnie kolorowany przy różnych permutacjach wierzchołków [6].

## LITERATURA

1. Anthony M.: On the mean chromatic number, *Discrete Math.* 125 (1994) 11–14.
2. Anthony M., Biggs N.: The mean chromatic number of paths and cycles, *Discrete Math.* 120 (1993) 227–231.
3. Barnett V.: *Elementy teorii pobierania prób*, Biblioteka problemów, t. 275, PWN, Warszawa 1982.
4. Bean D.R.: Effective coloration, *J. Symbolic Logic* 41 (1976) 469–480.
5. Borowiecki P.: Kolorowanie grafów w trybie on-line, *Zesz. Nauk. Politech. Śląskiej, ser. Automatyka*, z. 123, Gliwice 1998, 65–75.
6. Borowiecki P.: Application of on-line graph coloring in resource constrained on-line scheduling, Technical Report IM-1-99, Institute of Mathematics, Technical University of Zielona Góra, Zielona Góra 1999.
7. Borowiecki P.: On-line  $\mathcal{P}$ -coloring of graphs, przedłożone do druku w *Discussiones Mathematicae Graph Theory*.
8. Diestel R.: *Graph Theory*, Springer-Verlag, Berlin 1997.
9. Garey M.R., Johnson D.S.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco 1979.
10. Grimmett G.R., McDiarmid C.J.H.: On colouring random graphs, *Math. Proc. Cambridge Phil. Soc.*, 77 (1975) 313–324.
11. Hansen P., Kuplinsky J.: The smallest hard-to-color graph, *Discrete Math.* 96 (1991) 199–212.
12. Kierstead H.A.: Coloring Graphs On-line, in: A. Fiat and G.J. Woeginger eds., *Online Algorithms - The State of the Art*, LNCS 1442, Springer Verlag, Berlin 1998, 281–305.
13. Kubale M.: Problem kolorowania wierzchołków grafów. Przegląd algorytmów i zastosowań, *Zesz. Nauk. Politech. Śląskiej, ser. Automatyka*, z. 114, Gliwice 1994, 187–198.
14. Kubale M.: *Introduction to Computational Complexity and Algorithmic Graph Coloring*, Wydawnictwo GTN, Gdańsk 1998.
15. Lovász L., Saks M., Trotter W.T.: An on-line graph coloring algorithm with sublinear performance ratio, *Discrete Math.* 75 (1989) 319–325.
16. Manuszewski K.: *Grafy algorytmicznie trudne do kolorowania*, rozprawa doktorska, Politechnika Gdańska, Wydział ETI, Gdańsk 1997.

Recenzent: Prof. dr hab. inż. M. Kubale

## 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 live 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. Mean chromatic number, mean performance ratio and mean performance function as well as newly defined susceptibility of graph to algorithm A are analyzed. As a result of experimental study a range of values for mean performance function is calculated and a catalog of graphs non-susceptible to algorithm First-Fit is generated.