ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: AUTOMATYKA z.117 Nr kol. 1337

Marek KUBALE, Krzysztof MANUSZEWSKI
Politechnika Gdańska

ALGORYTMY SUMARYCZNEGO KOLOROWANIA GRAFÓW

Streszczenie. Suma chromatyczna $\Sigma(G)$ jest najmniejszą możliwą sumą kolorów spośród wszystkich legalnych pokolorowań wierzchołków grafu G liczbami naturalnymi. W niniejszej pracy przedstawiono wnioski płynące z przystosowania najbardziej znanych algorytmów kolorowania klasycznego do potrzeb kolorowania sumarycznego. Dla każdego takiego algorytmiu podano jego opis, złożoność obliczeniową, funkcję dobroci oraz najmniejsze trudne do kolorowania grafy. Rozważania ogólne poparte zostały doświadczeniami komputerowymi z implementacji i testowania tych algorytmów na identycznych seriach grafów pseudolosowych.

ALGORITHMS FOR THE CHROMATIC SUM OF A GRAPH

Summary. The chromatic sum $\Sigma(G)$ is the smallest total among all proper colorings of graph G using natural numbers. In the paper we give conclusions following from the adaptation of the most popular graph coloring algorithms to the need of chromatic sum coloring. For any such algorithm we give its description, computational complexity, performance guarantee and the smallest hard to color graphs. General considerations are supported by computational experience gained with implementation and profiling of analyzed algorithms on identical series of pseudorandom graphs.

1. Wprowadzenie

Suma chromatyczna $\Sigma(G)$, czyli najmniejsza możliwa suma kolorów spośród wszystkich legalnych pokolorowań wierzchołków grafu G liczbami naturalnymi, jest stosunkowo nowym parametrem grafu G. Potrzeba zajmowania się takim właśnie modelem kolorowania grafu wynika z zastosowania metody chromatycznej do rozwiązywania problemów szeregowania zadań jednostkowych. Łatwo bowiem zauważyć, że znalezienie pokolorowania chromatycznego odpowiada znalezieniu uszeregowania minimalizującego zarówno średni czas przepływu, jak i całkowitą długość uszeregowania.

Niestety, obliczenie wartości $\Sigma(G)$ jest problemem NP-trudnym. Dlatego w praktyce zmuszeni jesteśmy korzystać z istniejących algorytmów wielomianowych kolorowania

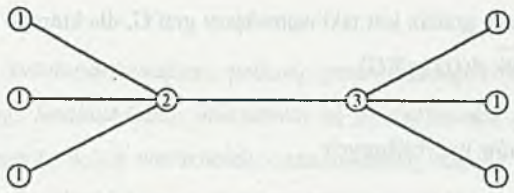
suboptymalnego zaadaptowanych do potrzeb kolorowania sumarycznego. Ponieważ każdy algorytm kolorowania klasycznego daje również pewne pokolorowanie sumaryczne, adaptacja taka jest oczywista i naturalna.

W niniejszej pracy przedstawimy wnioski wynikające z przystosowania najbardziej znanych algorytmów kolorowania klasycznego do potrzeb kolorowania sumarycznego. Będą to zarówno algorytmy sekwencyjne, jak i maksymalnych zbiorów niezależnych. Dla każdego takiego algorytmu podamy jego opis, złożoność obliczeniową, funkcję dobroci oraz najmniejsze trudne do kolorowania grafy. Rozważania ogólne zostaną poparte doświadczeniami komputerowymi z implementacji i testowania tych algorytmów na identycznych seriach grafów pseudolosowych.

2. Dotychczasowe wyniki

Niech $G=(V,E)$ będzie grafem spójnym ze zbiorem wierzchołków V i zbiorem krawędzi E. W dalszym ciągu przez $n=|V|$ oznaczać będziemy liczbę wierzchołków, zaś przez $m=|E|$ liczbę krawędzi. *Suma chromatyczna* $\Sigma(G)$ jest najmniejszą sumą $\Sigma_{v\in V}c(v)$ spośród wszystkich legalnych (tj. takich, że końce każdej krawędzi są różnobarwne) pokolorowań c wierzchołków grafu G przy użyciu liczb naturalnych, tzn. funkcji $c: V\rightarrow N$, takich że $c(u)\neq c(v)$, gdy $\{u,v\}\in E$. Pokolorowanie c, dla którego $\Sigma_{v\in V}c(v)=\Sigma(G)$, nazywamy *najlepszym*.

Wydawać by się mogło, że pokolorowanie najlepsze osiągniemy wówczas, gdy znajdziemy pokolorowanie przy użyciu minimalnej liczby kolorów, a następnie pomalujemy największy zbiór niezależny kolorem 1, kolejny największy zbiór niezależny kolorem 2 itd. Jednakże Kubicka i Schwenk [6] pokazali, że nawet drzewa, czyli grafy 2-chromatyczne, mogą wymagać więcej niż dwóch kolorów dla osiągnięcia pokolorowania najlepszego. Przykład takiego drzewa pokazujemy na rys.1. Co więcej, dla każdego naturalnego k istnieje drzewo T_k wymagające k kolorów dla osiągnięcia sumy chromatycznej $\Sigma(T_k)$. Drzewo takie ma przynajmniej $n=((2+\sqrt{2})^{k-1}-(2-\sqrt{2})^{k-1})/\sqrt{2}$ wierzchołków [3]. Pokazano, że liczba takich drzew rośnie szybciej niż liczba wszystkich drzew n-wierzchołkowych, gdy $n\rightarrow\infty$.



Rys. 1. Drzewo wymagające 3 kolorów do najlepszego pokolorowania sumarycznego
Fig. 1. A tree requiring 3 colors for the best total coloring

Problem optymalnego pokolorowania sumarycznego jest NP-trudny [6]. Dlatego w praktyce skazani jesteśmy na korzystanie z oszacowań dolnych i górnych. Jest oczywiste, że $\Sigma(G)$ jest najmniejsze, gdy G jest pusty i największe, gdy G jest pełny. Dlatego

$$n \leq \Sigma(G) \leq n(n-1)/2. \tag{1}$$

Pierwsze nietrywialne oszacowanie z góry podane zostało przez Thomassena i in. [8] w postaci nierówności

$$\Sigma(G) \leq m + n. \tag{2}$$

Oszacowanie to zostało poprawione w tej samej pracy w sposób następujący:

$$\lceil \sqrt{8m} \rceil \leq \Sigma(G) \leq \lfloor 3(m+1)/2 \rfloor. \tag{3}$$

Oszacowanie powyższe można uściślić w przypadku drzew T, mianowicie

$$n+1 \leq \Sigma(T) \leq \lfloor 3n/2 \rfloor. \tag{4}$$

Zauważmy na marginesie, że każde drzewo można pomalować optymalnie w czasie $O(n)$ [6].

Niech $A(G)$ oznacza sumę kolorów użytych do pomalowania sumarycznego grafu G metodą A. Z każdym takim algorytmem kolorowania można związać funkcję dobroci A: $N \rightarrow R$ zdefiniowaną następująco

$$A(n) = \max\{A(G)/\Sigma(G): G \text{ jest rozmiaru } n\}. \tag{5}$$

Funkcja ta charakteryzuje jakość rozwiązań generowanych przez A w najgorszym przypadku danych. Rzecz jasna, wobec nierówności (1) mamy $A(n) = O(n)$. W następnym punkcie pokażemy, że funkcja ta jest w istocie liniowa dla większości klasycznych algorytmów kolorowania wierzchołków zaadaptowanych dla potrzeb kolorowania sumarycznego.

Dla danego algorytmu przybliżonego A *najmniejszy dość trudny* do kolorowania sumarycznego graf definiuje się jako najmniejszy w sensie liczby wierzchołków graf G, taki że pewna implementacja algorytmu A daje wynik $A(G) > \Sigma(G)$. Jeśli jest więcej takich grafów n-wierzchołkowych, wybiera się ten, który ma najmniej krawędzi. Natomiast *najmniejszym*

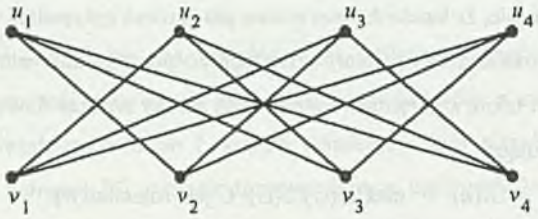
trudnym do kolorowania grafem jest taki najmniejszy graf G, dla którego każda implementacja algorytmu A daje wynik $A(G) > \Sigma(G)$.

3. Przegląd algorytmów przybliżonych

W niniejszym punkcie omawiamy cztery metody suboptymalnego kolorowania wierzchołków. Trzy pierwsze: *LF*, *SL*, *SLF* są to metody sekwencyjne, tzn. takie, że wierzchołki kolorowane są zachłannie zgodnie z pewną kolejnością. Ostatnia metoda *GIS* jest algorytmem zbiorów niezależnych. Popularność tych metod wynika z łatwości ich zaimplementowania na komputerze.

3.1. Metoda LF

W metodzie *pierwszy-największy*, zwanej w skrócie *LF* (ang. *Largest-First*), wierzchołki są najpierw porządkowane w kolejności nierosnących stopni, a następnie odbywa się właściwe kolorowanie sekwencyjne. Algorytm ten został po raz pierwszy podany przez Welsha i Powella [9]. Jego złożoność wynosi $O(m+n)$. Rodzinę najgorszych do kolorowania klasycznego grafów dla *LF* podał Johnson [5] (rys. 2). Grafy Johnsona wykorzystamy również w analizie dobroci kolorowania sumacyjnego.

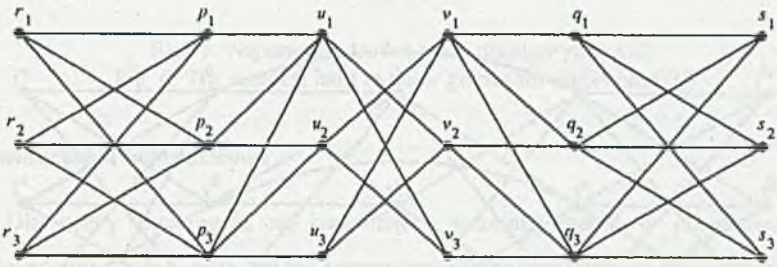


Rys. 2. Graf J_4 dla algorytmu *LF*
Fig. 2. Graph J_4 for algorithm *LF*

Niech $J_k = (V_k, E_k)$ będzie k-tym grafem Johnsona. Jeżeli wierzchołki kolorowane są w kolejności $u_1, v_1, \ldots, u_k, v_k$, to każda para otrzymuje nowy kolor. Zatem $LF(J_k) = 2(1+\ldots+k) = k(k+1)$. Jednakże $\Sigma(J_k) = k+2k = 3k$. Skoro $|V_k| = 2k$, to $LF(n) = O(n)$. Najmniejszym dość trudnym i zarazem trudnym do kolorowania grafem jest ścieżka P_3. Łatwo bowiem zauważyć, że $LF(P_3) = 5$, podczas gdy $\Sigma(P_3) = 4$.

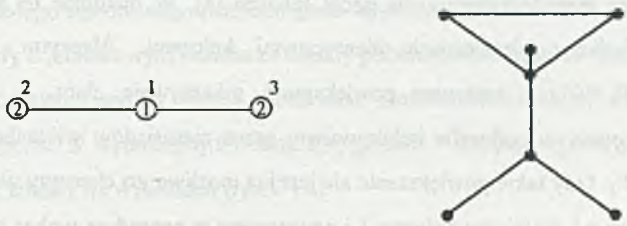
3.2. Metoda SL

W metodzie *ostatni-najmniejszy*, podanej przez Matulę i inn. [7] i nazwanej symbolicznie *SL* (ang. *Smallest-Last*), wierzchołki są porządkowane jako $v_1,\ldots,v_n$ w taki sposób, że dla każdego $i = 1,\ldots,n$ wierzchołek v_i ma minimalny stopień w podgrafie grafu G indukowanym przez wierzchołki $v_1,\ldots,v_i$. Algorytm *SL* można zrealizować w czasie $O(m+n)$. Rodzinę najgorszych do kolorowania grafów dwudzielnych podali Coleman i More [2]. Na rys.3. podajemy trzeci graf w tej rodzinie. Jeżeli wierzchołki grafu CM_k pomalujemy według następującej kolejności *SL*: $q_1,s_1,\ldots,q_k,s_k,p_1,r_1,\ldots,p_k,r_k,u_1,v_1,\ldots,u_k,v_k$, to każda para u_i,v_i dostanie kolor $i+1$, $i = 1,\ldots,k$. Zatem $SL(CM_k) = 2\cdot 3k+2\cdot(2+\ldots+k+1) = 6k+k(k+3) = k^2+9k$. Z drugiej strony $\Sigma(CM_k) = 3k+3\cdot 2k = 9k$, oraz $|V_k| = 6k$, zatem $SL(n) = O(n)$.



Rys. 3. Graf CM_3 dla algorytmu *SL*
Fig. 3. Graph CM_3 for algorithm *SL*

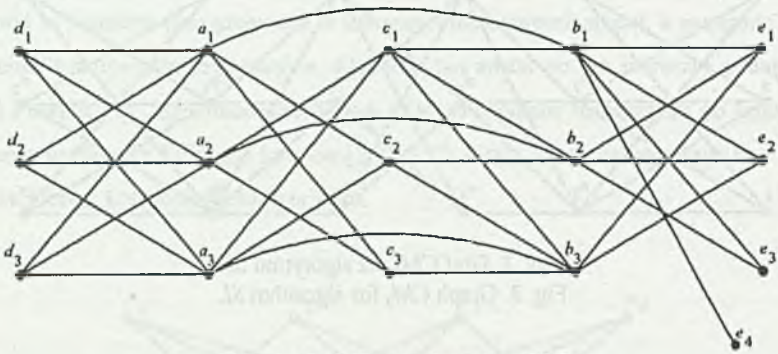
Grafy najmniejszy dość trudny i najmniejszy trudny dla algorytmu *SL* są pokazane na rys.4.



Rys. 4. Najmniejsze trudne grafy dla algorytmu *SL*
Fig. 4. The smallest hard to color graphs for algorithm *SL*

3.3. Metoda SLF

Dynamiczny sposób porządkowania wierzchołków zaproponował Brélaz [1]. W algorytmie *saturacyjny LF* (*SLF*) jako v_1 bierzemy wierzchołek o największym stopniu i przydzielamy mu kolor 1, a następnie dla każdego $i = 2,...,n$ jako v_i wybieramy wierzchołek o największym stopniu nasycenia i kolorujemy go najniższym możliwym kolorem. Jeżeli jest więcej niż jeden taki wierzchołek, to jako v_i brany jest wierzchołek o największym stopniu zwykłym. Algorytm *SLF* może być wykonany w czasie $O(m\log n)$. Rodzinę trudnych do kolorowania grafów stanowią grafy 3-chromatyczne jak na rys.5. Jeżeli wierzchołki grafu G_k malowane są w kolejności $a_1,b_1,c_1,...,a_k,b_k,c_k,d_1,...,d_k,e_1,...,e_{2k-2}$, to $SLF(G_k) = 4k+k(k+1)+k(k+3)/2+2k-2 = O(k^2)$. Jednakże $\Sigma(G_k) = (2k+2k-2)+2k+3k = O(k)$, oraz $|V_k| = 6k-2 = O(k)$, więc $SLF(n) = O(n)$. Ponownie, najmniejszym dość trudnym i trudnym grafem dla *SLF* jest P_3.

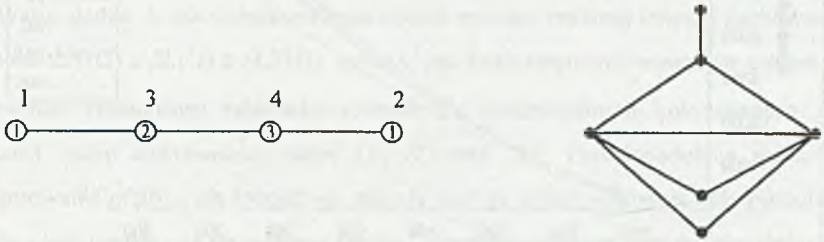


Rys. 5. Graf G_3 dla algorytmu *SLF*
Fig. 5. Graph G_3 for algorithm *SLF*

3.4. Metoda GIS

Odmienną metodę kolorowania podał Johnson [4]. W metodzie tej kolorujemy całe zbiory wierzchołków niezależnych identycznymi kolorami. Algorytm zaczynamy od podstawienia $V_1 = \emptyset$, a następnie powiększamy sukcesywnie zbiór V_1 o wierzchołek minimalnego stopnia w podgrafie indukowanym przez niesąsiadów wierzchołków aktualnie należących do V_1. Gdy takie powiększenie nie jest już możliwe, to elementy w V_1, których jest przynajmniej $\lfloor \log_2 n \rfloor$, malujemy kolorem 1 i powtarzamy tę procedurę wobec grafu rozpiętego na V-V_1 itd. Z uwagi na zachłanny sposób dobierania wierzchołków do kolejnych zbiorów niezależnych, metodę tę nazywamy algorytmem *zachłannym zbiorów niezależnych GIS* (ang. *Greedy Independent Sets*). Algorytm *GIS* może być zrealizowany w czasie $O(mn)$.

Johnson [5] pokazał, że algorytm *GIS* koloruje każdy graf k-barwny co najwyżej $\lceil n/\log n \rceil$ kolorami. A zatem suma takiego kolorowania nie przekracza $O(n^2/\log n)$, co wobec oszacowania dolnego (1) oznacza, że $GIS(n) = O(n/\log n)$. Grafy najmniejszy dość trudny i najmniejszy trudny dla algorytmu *GIS* są pokazane na rys.6.

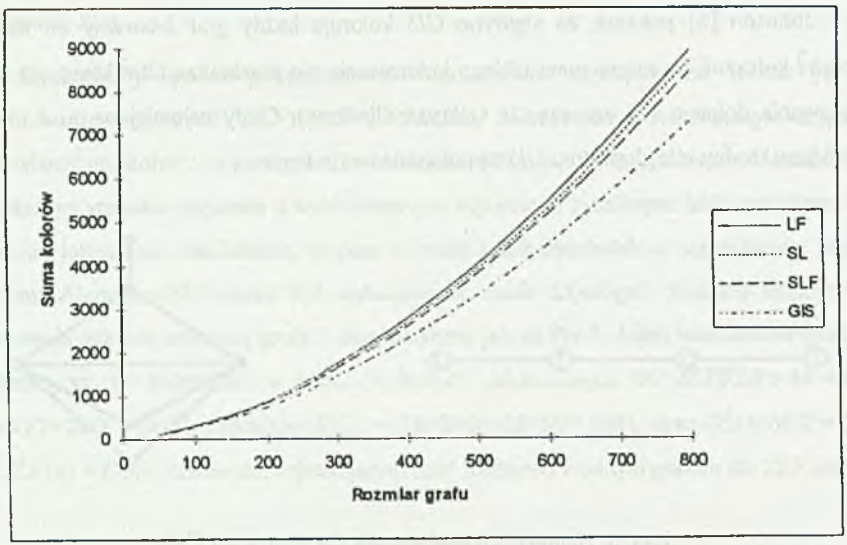


Rys. 6. Najmniejsze trudne grafy dla algorytmu *GIS*
Fig. 6. The smallest hard to color graphs for algorithm *GIS*

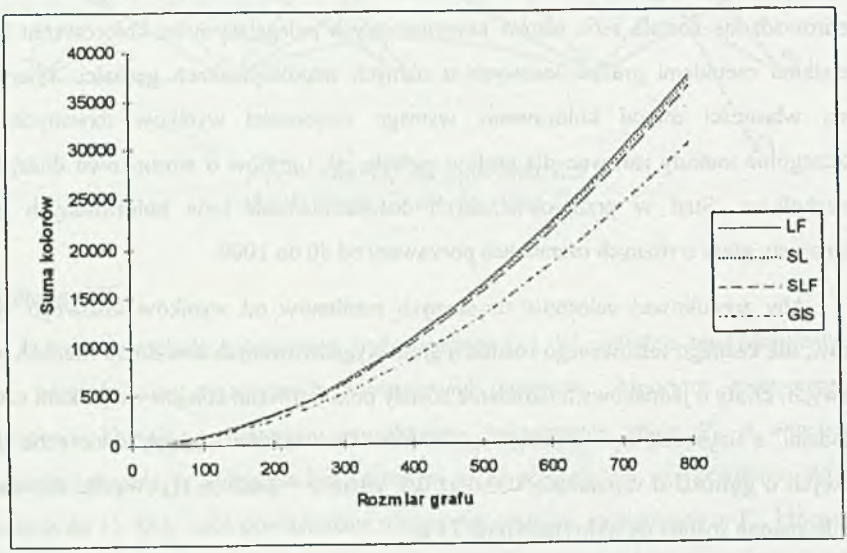
4. Doświadczenia komputerowe

Dla oceny zachowania się omawianych wcześniej metod w przypadku średnim przeprowadzona została seria testów komputerowych polegających na kolorowaniu kolejno wszystkimi metodami grafów losowych o różnych współczynnikach gęstości. Wiarygodna ocena własności metod kolorowania wymaga znajomości wyników dawanych przez poszczególne metody zarówno dla grafów małych, jak i grafów o stosunkowo dużej liczbie wierzchołków. Stąd w przeprowadzonych doświadczeniach serie kolorowanych grafów obejmowały grafy o różnych rozmiarach począwszy od 50 do 1000.

Aby zredukować zależność uzyskanych rezultatów od wyników losowego wyboru grafów, dla każdego testowanego rozmiaru grafu wygenerowanych zostało 50 różnych grafów losowych. Grafy o jednakowym rozmiarze zostały pokolorowane kolejno wszystkimi czterema metodami, a uzyskane wyniki zostały uśrednione. Doświadczenia objęły dwie serie grafów losowych o gęstości d wynoszącej 0,1 oraz 0,5, gdzie $d = 2m/(n(n-1))$. Wyniki doświadczeń przedstawione zostały na wykresach (rys. 7 i 8).



Rys. 7. Efektywność kolorowania sumacyjnego dla grafów losowych z d=0,1
Fig. 7. The efficiency of chromatic sum coloring for random graphs with d=0,1



Rys. 8. Efektywność kolorowania sumacyjnego dla grafów losowych z d=0,5
Fig. 8. The efficiency of chromatic sum coloring for random graphs with d=0,5

Jak wynika z przeprowadzonych doświadczeń, wszystkie testowane metody sekwencyjne, tj. *LF*, *SL* oraz *SLF*, dają wyniki zbliżone w sensie sumy kolorów użytych dla kolorowania grafów losowych. Widać jednak niezależną od rozmiaru grafu relację pomiędzy średnimi wynikami uzyskiwanymi przez poszczególne algorytmy. Stosunkowo najgorsze efekty daje najprostsza heurystyka *LF*, nieco lepsze heurystyka *SL* i kolejno heurystyka saturacyjna - *SLF*. Warto dodać, że dla wszystkich testowanych wartości zmiennej losowej zachowana była zależność $LF(G) \geq SL(G) \geq SLF(G)$, gdzie G jest konkretnym wylosowanym grafem. Fakt zachowania wspomnianej zależności również dla poszczególnych kolorowanych grafów wzmacnia ocenę efektywności metod *LF*, *SL* oraz *SLF*. Prawdopodobnie możliwe jest skonstruowanie grafów, dla których np. metoda *SLF* da gorsze efekty niż obie pozostałe, ale grafy te - jeśli istnieją - są stosunkowo rzadkie. Algorytm zachłanny zbiorów niezależnych *GIS* znacznie odbiega wynikami od pozostałych metod, dając regularnie lepsze pokolorowania. Uzyskiwane przez ten algorytm wyniki są wyraźnie o kilkanaście procent lepsze niż wyniki osiągane przez najlepszą z rodziny heurystyk sekwencyjnych, tj. metodę *SLF*.

Rozważając teoretycznie sposób działania poszczególnych metod, wydaje się oczywiste, że algorytm *LF* powinien dawać stosunkowo najgorsze rezultaty. Przy kolorowaniu wierzchołków o dużym stopniu należałoby bowiem oczekiwać nadania pojedynczemu wierzchołkowi stosunkowo najwyższego z używanych kolorów, tak aby jak największa ilość jego sąsiadów mogła dostać kolory o niskich numerach. Tymczasem metoda *LF* postępuje dokładnie odwrotnie. Metody *SL* oraz *SLF* stosują nieco bardziej wyrafinowaną strategię przydziału kolorów, ale ich efektywność nie odbiega znacząco od wyników osiąganych przez *LF*. Całkiem inaczej jest w przypadku heurystyki *GIS*. Strategia tej metody przewiduje taki dobór wierzchołków, aby możliwie jak największa liczba wierzchołków otrzymała kolor 1, potem 2 itd., co wydaje się doskonale odpowiadać wymogom narzucanym przez sumacyjny model kolorowania grafów.

LITERATURA

1. Brélaz D.; New methods to color the vertices of a graph, Comm. ACM, vol. 22, 1979, pp. 251-256.
2. Coleman T.F., More J.J.; Estimation of sparse Jacobian matrices and graph coloring problems, SIAM J. Numer. Anal., vol. 20, 1983, pp. 187-209.
3. Erdös P., Kubicka E., Schwenk A.J.; Graphs that require many colors to achieve their chromatic sum, Congr. Numer., vol. 71, 1990, pp. 17-28.

4. Johnson D.S., Approximation algorithms for combinatorial problems, J. Comp. Syst. Sci., vol. 9, 1974, pp. 256-278.

5. Johnson D.S., Worst-case behavior of graph coloring algorithms, Proc. 5th Southeastern Conf. on Combinatorics, Graph Theory and Computing, Utilitas Mathematicae, Winnipeg, 1974, pp. 513-527.

6. Kubicka E., Schwenk A.J., An introduction to chromatic sums, Proc. 7th Ann. ACM Computer Science Conf., ACM Press, 1989, pp. 39-45.

7. Matula D.W., Marble G., Isaacson D., Graph coloring algorithms, in: Graph Theory and Computing, Academic Press, New York, 1972, pp. 109-122.

8. Thomassen C., Erdös P., Alavi Y., Malde P.J., Schwenk A.J., Tight bounds on the chromatic sum of a connected graph, J. Graph Theory, vol. 13, 1989, pp. 353-357.

9. Welsh D.J., Powell M.B., An upper bound for the chromatic number of a graph and its application to timetabling problem, Comp. J., vol. 10, 1967, pp. 85-68.

Recenzent: Prof.dr hab.inż. Jerzy Klamka

Wpłynęło do Redakcji do 30.06.1996r.

Abstract

The chromatic sum $\Sigma(G)$, i.e. the smallest total among all proper colorings of graph G using natural numbers, is a relatively new graph invariant. The need for considering such a graph coloring model follows from its application to solve problems of scheduling unit execution time tasks. This is so because finding a best chromatic sum solution corresponds to finding a schedule that minimizes the mean flow time and total completion time.

Unfortunately, finding the value of $\Sigma(G)$ is NP-hard. Therefore we are forced to use polynomial-time approximation algorithms. Since any heuristic graph coloring algorithm also gives some total coloring, such an adaptation is straightforward and natural.

In this paper we give results following from the adaptation of the most popular graph coloring algorithms to the need of chromatic sum coloring. For any such algorithm we give its description, computational complexity, performance guarantee and the smallest hard to color graphs. General considerations are supported by computational experience gained with implementation and profiling of analyzed algorithms on identical series of pseudorandom graphs.