ELSEVIER

Discrete Mathematics 236 (2001) 151–165

DISCRETE MATHEMATICS

www.elsevier.com/locate/disc

The smallest hard-to-color graph for algorithm DSATUR

R. Janczewski *, M. Kubale, K. Manuszewski, K. Piwakowski[1]

Technical University of Gdańsk Foundations of Informatics Department, G. Narutowicza 11/12, 80-952 Gdańsk, Poland

Received 28 October 1997; revised 4 September 1998; accepted 5 March 2000

Abstract

For a given approximate coloring algorithm a graph is said to be slightly hard-to-color (SHC) if some implementation of the algorithm uses more colors than the chromatic number. Similarly, a graph is said to be hard-to-color (HC) if every implementation of the algorithm results in a non-optimal coloring. In the paper, we study the smallest of such graphs for the DSATUR vertex coloring algorithm. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Chromatic number; DSATUR algorithm; HC graph; SHC graph

AMS: 05C15

1. Introduction

As a rule, the performance of graph coloring heuristics is studied by giving asymptotic results. These are usually the worst-case performance guarantee and the worst-case time complexity. Both functions tell us what one can face at worst when using a given graph coloring algorithm if the number of vertices n goes to infinity. However, we do not know what is going on at the other end of the scale, say when $n < 10$. Therefore, Hansen and Kuplinsky [5] introduced the concept of the smallest hard-to-color graph and slightly hard-to-color graph. These are the smallest graphs which cannot be colored optimally by some approximation algorithms. The aim of studying such graphs is twofold. First, analyzing hard-to-color graphs makes it possible to obtain improved algorithms which avoid hard instances as far as possible. Second, it enables us to search for small benchmarks for comprehensive families of graph coloring algorithms (cf. [4]). These are the graphs which are hard to color for every algorithm in a given family.

* Corresponding author.
E-mail address: skalar@eti.pg.gda.pl (R. Janczewski).
[1] Supported by FNP and KBN.

PII: S0012-365X(00)00439-8

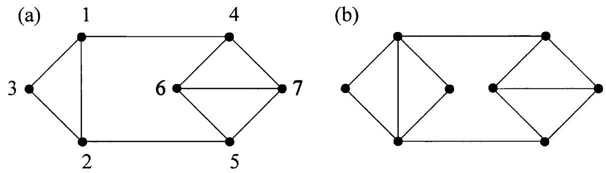


Fig. 1. The smallest graphs for algorithm DSATUR: (a) slightly HC; (b) HC.

A graph G is said to be *slightly hard-to-color* (SHC) with respect to an algorithm A if for some instance of it the number $A(G)$ of colors used satisfies $A(G) > \chi(G)$, where $\chi(G)$ is the chromatic number of G. We similarly define a *hard-to-color* (HC) graph as one for which every application of the algorithm (i.e. no matter what choice is made to break ties) results in a non-optimal coloring. Moreover, we define *smallest* graphs for which a given algorithm produces non-optimal colorings. More precisely, in the case of SHC graphs among all graphs $G=(V,E)$ we are looking for a graph $G'=(V',E')$ which realizes

$$|E'| = \min\{|E|: \text{ for some instance of } A, A(G) > \chi(G), \text{ and } |V| = n_0\},$$

where

$$n_0 = \min\{|V|: \text{ for some instance of } A, A(G) > \chi(G)\}.$$

HC graphs are defined in the same way with words ‘for which’ instead of ‘for some’.

So far, the only heuristics with the known smallest HC and SHC graphs have been the *largest-first* (LF) and the *smallest-last* (SL) sequential algorithms. Namely, Hansen and Kuplinsky [6] proved that path P_6 and the so-called envelope are the unique smallest SHC and HC graphs, respectively, for algorithm LF. Recently, Kubale et al. [8] showed that the prism and prismatoid are the unique smallest SHC and HC graphs, respectively, for algorithm SL. Babel and Tinhofer [1] studied a general connected sequential algorithm and proved fan F_5 to be the smallest SHC graph (a smallest HC graph for this method is unknown). These and some other smallest SHC/HC graphs were computationally verified in [7] and the problem of chromatic sum coloring was considered from a SHC/HC viewpoint in [4].

The remainder of this paper is organized as follows. In Section 2 we give preliminary results. In particular, we review the classes of graphs which are colored optimally by DSATUR. In Section 3, we introduce a concept of algorithm SLF and consider its SHC and HC graphs. The SLF algorithm is much easier to analyze than DSATUR and has the property that every graph which is SHC for DSATUR remains SHC for SLF and every HC graph for SLF is also HC for DSATUR. The main results of this paper are given in Section 4. We give there a proof that the graphs in Fig. 1 are the smallest SHC and HC graphs for algorithm DSATUR. Note that both graphs were mentioned as hard to color in [5,7] without giving a formal proof of their minimality.

2. Preliminary results

Throughout the paper $G=(V,E)$ is a simple connected graph. By $\Delta(G), \delta(G), m(G)$ and $n(G)$ we denote the maximum degree, minimum degree, size, and order of G, respectively. We shall drop the reference to the graph, if G is clear from the context.

Algorithm DSATUR, along with LF and SL, belong to the three most popular vertex-coloring algorithms. In contrast to its two competitors, it is based on a dynamic ordering of vertices. More formally, let c be a partial coloring of the vertices of G and let v be an uncolored vertex. By $\rho(v)$ we mean the saturation degree of v, that is, the number of adjacent distinctly colored vertices. By $\deg(v)$ we denote the ordinary vertex degree. Algorithm DSATUR performs for each $i=1,\ldots,n$ the following two steps:

1. Select a vertex v_i with maximum saturation degree breaking ties by choosing a vertex of greater ordinary degree, i.e. $\rho(v_i)=\max\rho(V_i)$ and $\deg(v_i)=\max\{\deg(w)\colon w\in V_i \wedge \rho(w)=\max\rho(V_i)\}$, where $V_i=V\setminus\{v_1,\ldots,v_{i-1}\}$.
2. Color v_i greedily, i.e. so that color $c(v_i)=\min\{k\in N^+\colon c(v_j)\neq k$ for each $j<i$ such that $\{v_i,v_j\}\in E\}$.

Algorithm DSATUR, which has become the *de facto* standard among graph-coloring algorithms (cf. [9]), was given by Brélaz [2]. Turner [11] showed how this heuristic algorithm could be efficiently implemented in time $O(m\log n)$. Spinrad and Vijayan [10] constructed a family of 3-chromatic graphs for which the number of colors used by some implementation of DSATUR is $O(n)$. It is easy to see that

$$\mathrm{DSATUR}(G)=\max\{\rho(v_i)\colon 1\leqslant i\leqslant n\}+1,$$

where $v_1,\ldots,v_n$ is the sequence in which the vertices of G are colored in the course of the algorithm. We call such a valid sequence a DSATUR *sequence*. Any DSATUR sequence for which $\mathrm{DSATUR}(G)=\chi(G)$ is said to be *optimal.* Also, note that

$$\rho(v_i)\leqslant \deg_i(v_i),$$

where $\deg_i(v_i)$ is the degree of v_i in the subgraph of G induced by vertices $v_1,\ldots,v_i$.

The DSATUR algorithm has some nice properties. First of all, it optimally colors all bipartite graphs [2]. Next, it optimally colors the following k-degenerate graphs (i.e. such that each subgraph $G'\subset G$ has $\delta(G')\leqslant k$ for a fixed k): cycles, uni- and duocyclic graphs, cacti, trees of polygons, wheels, and the so-called necklaces. A *necklace* $N_{i_1,\ldots,i_k}$ is a graph whose edges constitute $k\geqslant 2$ chordless paths of lengths $i_1,\ldots,i_k$ joining the same pair of vertices. All paths are vertex disjoint except their endpoints. For example, cycle C_5 can be regarded as $N_{2,3}$ or $N_{1,4}$. On the other hand, DSATUR optimally colors almost all k-colorable graphs (see [11] for details).

In the following, we need a notion of the core. The *core* of G is a subgraph obtained by pruning away all pendant vertices successively until there are no vertices of degree 1. The following proposition enumerating the main classes of graphs that are colored optimally is straightforward.

Proposition 2.1. *No graph whose core is*:

(1) *a single vertex*,
(2) *a bipartite graph*,
(3) *a wheel*,
(4) *a complete multipartite graph*,
(5) *a cactus and*
(6) *a necklace*

is SHC for algorithm DSATUR (*and algorithm SLF of Section* 3).

Proposition 2.1(1) and (2) follows from the bipartiteness. Proposition 2.1(3) and (4) are obvious. Proposition 2.1(5) follows from the fact that cacti are graphs that can be decomposed into cycles any two of them have at most a vertex in common. If a vertex v gets color 3 then v is the final vertex of an odd cycle colored. Therefore, for any uncolored $v \in V \rho(v) \leqslant 2$ and no color 4 need be introduced. Proposition 2.1(6) follows from the fact that the only vertices of degree greater than 2 are path endpoints u, v which satisfy $\rho(u) \leqslant 1 \leqslant \rho(v) \leqslant 2$, where u is the vertex colored before v, and $\rho(v) = 2$ only if the necklace is 3-chromatic.

Let p and $n_1, \ldots, n_p$ be integers satisfying $p \geqslant 1$ and $n_1 \geqslant n_2 \geqslant \cdots \geqslant n_p \geqslant 3$. Let G be the class of graphs defined recursively by the rules: the n_i-cycle is in G for each $i = 1, \ldots, p$ and if G_1 and G_2 belong to G then so does any graph that can be formed from G_1 and G_2 by identifying an edge of G_1 with an edge of G_2. The graphs in G are called $(n_1, \ldots, n_p)$-*gon trees* or simply *trees of polygons.*

Proposition 2.2. *No tree of polygons is SHC for algorithm DSATUR* (*and algorithm SLF of Section* 3).

Proof. Let G be a tree of polygons. First of all observe that G is 2-degenerate and thus $\chi(G) \leqslant 3$. For the need of the proof, we introduce a *random saturation largest-first* (RSLF) algorithm. It repeatedly chooses an uncolored vertex v such that its saturation degree is maximum and puts v in any color class 1, 2 or 3 where it fits. If, however, none of them is available to v then the color is chosen according to the smallest-first principle.

To prove that RSLF$(G) \leqslant 3$, we use induction on the number k of polygons in G.

1. If $k = 1$ then $G = C_n$ and RSLF$(G) \leqslant 3$, since $\Delta = 2$.
2. Assume that the thesis is true for $k = 1, \ldots, p-1$ and let us check whether it holds for $k = p$.

Let G have p polygons. Then G can be represented as a graph H of $p-1$ polygons and a path $(w_1, \ldots, w_r)$, where only $w_1, w_r \in V(H)$ and $\{w_1, w_r\} \in E(H)$. Let $s = ((v_1, c(v_1)), \ldots, (v_n, c(v_n)))$ be any RSLF sequence of coloring G (v_1 is colored with $c(v_1)$, etc.) and let s' be the sequence obtained from s by deleting the vertices $w_2, \ldots, w_{r-1}$ along with their colors. We prove that s' is a valid RSLF sequence of coloring H. To do this it is enough to show that in the moment of coloring any vertex $v_i \in V(H)$ the following two conditions are satisfied: (a) v_i is a vertex with maximum $\rho(v_i)$ among all yet uncolored vertices of H and (b) assignment of color $c(v_i)$ is consistent with the rule of RSLF. Without loss of generality, we assume that w_1 is followed by w_r in s (and so in s'). Let us consider three cases.

Case 1: $v_i \neq w_1$ and $v_i \neq w_r$.

(a) Vertices $w_2, \ldots, w_{r-1}$ are not adjacent to v_i in G so in the moment of coloring the degree $\rho(v_i)$ is the same for the sequence s' of graph H and for the sequence s of graph G. The same concerns all uncolored vertices of H except w_1, w_r whose saturation degree in s' can be less than or equal to that in s of G. Since v_i has the maximal saturation degree while coloring G according to s, it has the maximal saturation degree while coloring H according to s' as well.

(b) Consistency of coloring v_i with the rules of RSLF follows from the fact that in the moment of assigning a color to v_i this vertex has exactly the same adjacent vertices colored identically in both cases.

Case 2: $v_i = w_1$.

(a) Let us consider two cases.

 (a.1) $v_1 \in V(H)$. Then in the moment of coloring v_i according to s of G the vertices yet uncolored in G have saturation degrees identical in s to that in s' of H. Since v_i has the maximal saturation degree while coloring G according to s, it has the maximal saturation degree while coloring H according to s'.

 (a.2) $v_1 \notin V(H)$. Then v_i is the first vertex in sequence s' and $\rho(v) = 0$ for all $v \in V(H)$.

(b) If $v_1 \in V(H)$ then the argument of Case 1(b) applies. If $v_1 \notin V(H)$ then the consistency is obvious.

Case 3: $v_i = w_r$.

(a) Let us consider two cases.

 (a.1) w_{r-1} *precedes* w_r *in* s *and* $c(w_1) \neq c(w_{r-1})$. Two subcases are possible.

  (a.1.1) w_{r-1} *is followed by* w_1 *in* s. Then $\{w_1, \ldots, w_{r-1}\}$ is the exact set preceding v_i in s. Thus in the beginning of s', we have w_1 and v_i. Hence in coloring according to s' vertex v_i has $\rho(v_i) = 1$ (in the moment of coloring) and the remaining uncolored vertices have $\rho(v) \leqslant 1$.

  (a.1.2) w_1 *is followed by* w_{r-1} *in* s. Then while coloring G in the moment of coloring vertex w_{r-1} has $\rho(w_{r-1}) = 1$. It is the maximal saturation degree, so the remaining uncolored vertices $v \in V(H)$ have $\rho(v) \leqslant 1$. After coloring vertex w_{r-1}, the new value of saturation degree of v_i is 2 (since w_1 is before v_i in s). Thus, v_i is the immediate successor of w_{r-1} in s. This implies that v_i has $\rho(v_i) = 1$ in s' (in the moment of coloring), but the remaining uncolored vertices $v \in V(H)$ have $\rho(v) \leqslant 1$.

(a.2) *w_r precedes w_{r-1} in s or $c(w_1) = c(w_{r-1})$.* Then, similar to (1.a), in both colorings according to s and s' in the moment of coloring v_i the saturation degrees of all vertices in $V(H)$ are identical except w_1, whose saturation can be less in s' than in s (but for sure not greater). Since v_i has maximal saturation degree while coloring G according to s, it has maximal saturation while coloring H according to s' (in the moment of coloring).

(b) If w_{r-1} precedes w_r and $c(w_1) \neq c(w_{r-1})$ then by Case 3(a.1) $c(v_i) \in \{1, 2, 3\}$. Otherwise see Case 1(b).

We have shown that s' is an RSLF sequence for H. So by induction hypothesis RSLF$(H) \leqslant 3$. Since s and s' color H in the same way, any RSLF sequence for G uses colors 1, 2, 3 for $w_2, \ldots, w_{r-1}$, because each of them is of degree 2. Therefore RSLF$(G) \leqslant 3$. Obviously, if $\chi(G) \leqslant 2$ then SLF$(G) = \chi(G)$ and DSATUR$(G) = \chi(G)$. If $\chi(G) = 3$ then RSLF$(G) = 3$. Since any SHC graph for SLF and DSATUR remains SHC for RSLF, Proposition 2.2 follows. □

3. Algorithm SLF

Algorithm *saturation largest-first* (SLF) is a simplification of DSATUR. It starts with coloring an arbitrary vertex, say v_1, with color 1 and then keeps the following invariant for each $i = 2, \ldots, n$:

1. The vertex to color in step i is an uncolored vertex with maximum saturation degree; if there are at least two such vertices then the next vertex to color is selected at random.
2. The chosen vertex, call it v_i, is colored greedily.

The only difference from DSATUR is in the way ties are broken. Therefore the SLF algorithm has the following property.

Proposition 3.1. *Any HC graph for SLF is HC for DSATUR. Any SHC graph for DSATUR is SHC for SLF.*

Moreover, it colors optimally the same graphs that are listed in Proposition 2.1. For this reason Proposition 2.1 holds true for algorithm SLF as well. However, SLF yields arbitrarily bad results on 3-partite graphs shown in Fig. 2.

Proposition 3.2. *If G is SHC for the SLF algorithm then*

(i) $3 \leqslant \chi(G) \leqslant n(G) - 2$,
(ii) $\Delta(G) \geqslant 3$.

Proof. Property (i) follows from Proposition 2.1(2) and the fact that SLF uses n colors only on K_n. Property (ii) is obvious. □

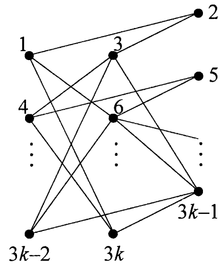


Fig. 2. Graph G_k for which $\mathrm{SLF}(G_k) = k + 1$.

The following corollary is a straightforward consequence of Brooks' bound [3].

Corollary 3.3. *If G is SHC for the SLF algorithm then* $\chi(G) \leqslant \Delta(G)$.

Proposition 3.4. *No graph with a spanning star is smallest SHC for the SLF algorithm.*

Proof. Suppose G is a smallest SHC graph for SLF with $\Delta = n - 1$. Let $(v_1, \ldots, v_n)$ be a non-optimal SLF sequence for G and let $\deg(v_k) = n - 1$. Then the sequence $(v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_n)$ is non-optimal for graph $G - v_k$, a contradiction to the hypothesis that G is the smallest. □

Proposition 3.5. *Let G be a graph with bridge e. If G is SHC for the SLF algorithm then at least one of connected components of* $G - e$ *is SHC for SLF.*

Proof. Let $(v_1, \ldots, v_n)$ be a non-optimal SLF sequence for G and let H_1 and H_2 denote the two connected components of G obtained after the deletion of edge e. Withouts loss of generality assume that $v_1 \in V(H_1)$. Since coloring of H_1 is independent of coloring of H_2, we can split $(v_1, \ldots, v_n)$ into two subsequences $(u_1, \ldots, u_k)$ and $(w_1, \ldots, w_l)$ such that $u_1, \ldots, u_k \in (H_1)$ and $w_1, \ldots, w_l \in V(H_2)$ and $(u_1, \ldots, u_k)$ is an SLF sequence for H_1. Moreover, the colors assigned to vertices $u_1, \ldots, u_k$ in both sequences are identical. Thus, if SLF fails within H_1, i.e. uses a color greater than $\chi(G)$, then H_1 is the desired subgraph. Otherwise, SLF fails within H_2. Let us consider this case in more detail. Let $e = \{u_i, w_1\}$. The following two subcases are possible.

Case 1: *Algorithm SLF has assigned color* 1 *to* w_1. Then $(w_1, \ldots, w_l)$ is an SLF sequence for H_2 and the colors assigned to vertices w_i are the same as used when coloring graph G.

Case 2: *Algorithm SLF has assigned color* 2 *to* w_1. Then $(w_2, w_1, w_3, \ldots, w_l)$ is an SLF sequence for H_2 and the colors assigned to vertices w_i are the same as used when coloring graph G, since w_2 was colored with 1.

It follows that H_2 is SHC for SLF. □

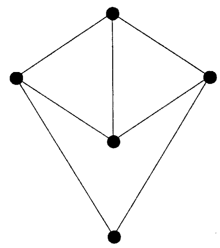


Fig. 3. Graph used in the proof of Proposition 3.7.

Proposition 3.6. *If G is SHC for the SLF algorithm then*

$$m(G) \geqslant (\chi(G)-2)(\chi(G)+1)/2+n(G).$$

Proof. Let G be SHC for SLF. By our assumption G is connected. Since there is an SLF sequence for which $\mathrm{SLF}(G) > \chi(G)$, graph G must have a vertex v with $\rho(v) \geqslant \chi(G)$. Moreover, v must be adjacent to $\chi(G)$ vertices each of which is colored differently. Thus, their degrees $\deg_i(v_i)$ must range from at least 0 to at least $\chi(G)-1$. Taking into account the fact that the saturation degree of the remaining vertices is at least 1 (in the moment of coloring), we get $m(G) \geqslant 0+\cdots+\chi(G)+n(G)-\chi(G)-1 = (\chi(G)-2)(\chi(G)+1)/2+n(G)$. □

Proposition 3.7. *No graph G with $n(G) \leqslant 5$ is SHC for the SLF algorithm.*

Proof. Clearly, no graph with at most 4 vertices is SHC for SLF. By Proposition 3.6 if $n=5$ and G is SHC then $m \geqslant 7$. All graphs K_5 without one or two edges have a spanning star. The only graph K_5 without three edges and a spanning star is shown in Fig. 3. It is easy to see that every SLF sequence for this graph is optimal. □

Lemma 3.8. *The graph shown in Fig. 4(a) is the unique smallest SHC graph for the SLF algorithm.*

Proof. By Propositions 2.1(2), 3.6, 3.7 and the fact that the graph in Fig. 4(a) is SHC for SLF it follows that a smallest SHC graph must have $m=8$ and $n=6$. By Proposition 3.5 it cannot have a vertex of degree 1. There are only 4 graphs with degree sequence $(3,3,3,3,2,2)$, 4 graphs with degree sequence $(4,3,3,2,2,2)$ and 2 graphs with degree sequence $(4,4,2,2,2,2)$. They are shown in Fig. 4. One can easily check that the graph of Fig. 4(a) is the only SHC graph for SLF. □

The reader can verify that all 6-vertex SHC graphs for algorithm SLF contain the smallest SHC graph, that is, each of them is a supergraph of the graph shown in Fig. 4(a). There are only 3 such graphs and they are all shown in Fig. 5.

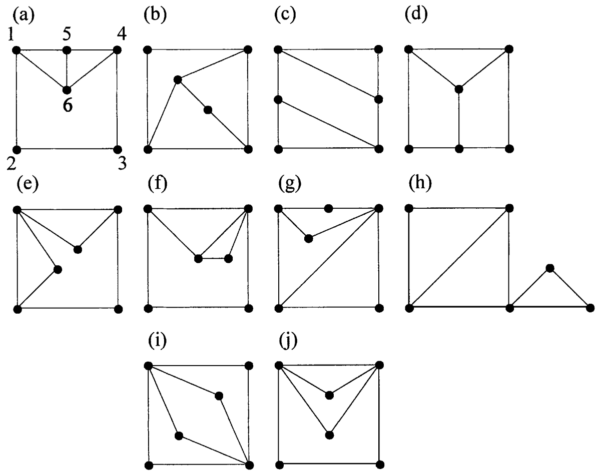


Fig. 4. Possible candidates for the smallest SHC graph for SLF: (a) SHC graph; (b) bipartite; (c) (4, 3, 3)-gon tree; (d) non-SHC; (e) (4, 3, 3)-gon tree; (f) (4, 3, 3)-gon tree; (g) non-SHC; (h) non-SHC; (i) bipartite; (j) necklace.

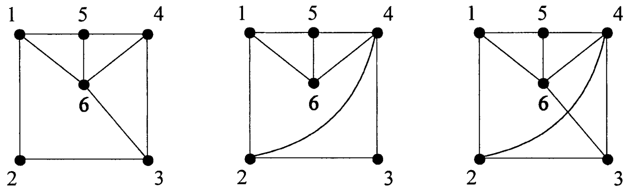


Fig. 5. Six-vertex SHC graphs for SLF with at least 9 edges.

4. Algorithm DSATUR

By Proposition 3.1 and Lemma 3.8 no graph smaller than that of Fig. 4(a) is SHC for the DSATUR algorithm. The reader can verify that the graph of Fig. 4(a) is not SHC for DSATUR either.

Proposition 4.1. *If G is SHC for the DSATUR algorithm with $\chi(G) = n(G) - 3$ and $\Delta(G) \leqslant n(G) - 2$, then*

(i) $\Delta(G) \geqslant n(G) - 3$
(ii) *G has $n(G) - 3$ vertices of degree at least $n(G) - 3$, which generate a clique or a clique without one edge.*

Proof. Property (i) follows from Proposition 3.1 and Corollary 3.3. To prove (ii) first observe that DSATUR will use at least $\chi(G) + 1 = n - 2$ colors on G. It cannot use n or $n - 1$ colors, since otherwise it would have a clique on at least $n - 2$ vertices and consequently $\chi(G) \neq n-3$. So DSATUR uses exactly $n-2$ colors. Obviously, no three vertices can be assigned the same color. Hence, two vertices will be colored with color 1, another two vertices will be colored with a color from $2,\dots,n-2$ and the remaining vertices will be colored with unique colors. Let $c(v_1)=c(v_k)=1$, $c(v_i)=c(v_j)>1$, and let $S=\{v_2,\dots,v_n\}\setminus\{v_i,v_j,v_k\}$. The vertices of S form a clique of order $n-4$, because they are colored uniquely. Moreover, each of them is adjacent to at least one of v_1, v_k and one of v_i, v_j. Therefore, the degree of any vertex from S is at least $n-3$. This clearly concerns vertex v_1 as well. Thus $n-3$ vertices of $S \cup \{v_1\}$ all have degree at least $n-3$ and constitute a clique or a clique without an edge (at v_1). □

Proposition 4.2. *If G is SHC for the DSATUR algorithm and* $\Delta(G)=n(G)-2$, *then there is an SHC graph for SLF of order* $n(G)-2$.

Proof. Let $(v_1,\dots,v_n)$ be a non-optimal DSATUR sequence for G. Obviously v_1 is the first vertex of degree $n-2$ and let v_k be the only vertex non-adjacent to v_1. Then $(v_2,\dots,v_{k-1},v_{k+1},\dots,v_n)$ is a non-optimal SLF sequence for graph $G-v_k-v_1$, since in G all vertices except v_k are adjacent to v_1, and v_1, v_k are the only vertices which receive color 1. □

Corollary 4.3. *If G is SHC for the DSATUR algorithm and* $n(G) \leqslant 7$, *then* $3 \leqslant \chi(G) \leqslant \Delta(G) \leqslant n(G)-3$.

Proposition 4.4. *Let G be SHC for the DSATUR algorithm. If G is* 2*-edge-connected then either* $\delta(G) \geqslant 3$ *or there is a vertex v of degree* 2 *such that* $G-v$ *is SHC for SLF.*

Proof. Let G be a bridgeless SHC graph for DSATUR. If $\delta<3$ then $\delta=2$, since G has no pendant edges. Let $H_1,\dots,H_k$ be connected components obtained from G by removing all vertices of degree 2. Consider two cases.

Case 1: $k=1$. Let v be any vertex of degree 2 in G and let u,w be the two neighbors of v. If at least one of u, w is of degree 2 then there is a non-optimal DSATUR sequence $(v_1,\dots,v_{n-1},v)$ for G and $(v_1,\dots,v_{n-1})$ is a non-optimal SLF sequence for $G-v$. If u, $w \in V(H_1)$ then $\deg(u) \geqslant \deg(w) \geqslant 3$ in G and both neighbors will be colored before v. Therefore $G-v$ is SHC for SLF.

Case 2: $k>1$. Let $S_1,\dots,S_l$ be the set of paths obtained after deleting the edges of $H_1,\dots,H_k$ from G. We assume that subgraphs $H_1,\dots,H_k$ are indexed in order in which DSATUR begins their coloring, and paths $S_1,\dots,S_l$ are indexed in order in which DSATUR completes their coloring. Since G is bridgeless so $l \geqslant k$. Notice that DSATUR begins coloring of G starting from a vertex of H_1, and completes its coloring before choosing a vertex of some S_j. Then it completes the coloring of S_1 before processing H_2 to which this path is adjoined to, etc. The last path colored S_l has a vertex v which is colored after its two neighbors have been colored. Therefore, as previously, $G-v$ is SHC for SLF. □

Corollary 4.5. *A smallest SHC graph for the DSATUR algorithm contains at least 7 vertices and 9 edges.*

Proof. By Proposition 3.7 no 5-vertex graph is smallest SHC for SLF and DSATUR. Suppose that G is a 6-vertex SHC graph for DSATUR. Then by Corollary 4.3 $\chi(G)=\Delta(G)=3$. If G is 3-regular then it is the prism (for a definition of the prism see e.g. [8]). However, the prism is not an SHC graph for DSATUR. If G is not 3-regular then $\delta(G)=2$. Clearly, G cannot have a bridge. So by Proposition 4.4 there is a vertex v with $\deg(v)=2$ such that $G-v$ is SHC for SLF. But $G-v$ has 5 vertices, a contradiction. Thus, $n(G)\geqslant 7$ and by Propositions 3.1 and 3.6 $m(G)\geqslant 9$. □

Proposition 4.6. *If G is SHC for DSATUR and $n(G)=7$ then $\delta(G)\geqslant 2$.*

Proof. Let $(v_1,\dots,v_7)$ be a non-optimal DSATUR sequence for G. Suppose that $\delta(G)=1$ and let v be a vertex with degree 1. Since DSATUR colors pendant vertices at latest, we may assume that $v=v_7$. Then $(v_1,\dots,v_6)$ is a non-optimal SLF sequence for $G-v$. However, there exist only four 6-vertex SHC graphs for SLF (Figs. 4(a) and 5) and there is no way to attach a pendant edge (i.e. an edge attached to an existing vertex whose second endpoint is a new vertex of degree 1) to any of them so that the resulting graph is SHC for DSATUR. □

Now we are ready to prove the following

Theorem 4.7. *The graph of Fig. 1(a) is the unique smallest SHC graph for the DSATUR algorithm.*

Proof. Let G be a smallest SHC graph for DSATUR. Since graph of Fig. 1(a) is SHC for DSATUR, $n(G)=7$ and $m(G)\leqslant 10$. Since G cannot be cubic and is bridgeless (by Proposition 3.5), by Proposition 4.4 it must have a vertex of degree 2 whose deletion results in a slightly HC graph for SLF. However, the smallest SHC graph for SLF has 6 vertices and 8 edges and there is only one way to attach to it a new vertex adjacent to two old vertices so that the new graph is SHC for DSATUR. This is just the graph shown in Fig. 1(a). □

Proposition 4.8. *If G is a 7-vertex SHC graph for the DSATUR algorithm and $m(G)\geqslant 11$ then $\delta(G)\geqslant 3$.*

Proof. Let G be such an SHC graph for DSATUR. Suppose to the contrary that $\delta<3$. By Proposition 3.5 G contains no bridge so it must have a vertex of degree 2 whose deletion results in an SHC graph for SLF. However, there is no way to attach a new vertex adjacent to two old vertices of graphs in Fig. 5 so that the resulting graph is SHC for DSATUR. □

How many 7-vertex graphs denser than that of Fig. 1(a) are SHC for DSATUR? If G is such a graph then it must fulfill the following inequalities $3 \leqslant \chi(G) \leqslant \Delta(G) \leqslant 4$ and $3 \leqslant \delta(G) \leqslant 4$. In addition, if G is 4-chromatic then it must have 4 vertices of degree 4 which generate K_4 or $K_4 - e$. There are 14 graphs with $\chi(G) = 3$ and 4 graphs with $\chi(G) = 4$ satisfying these presumptions. All these graphs are given in the appendix. None of them appears to be HC for DSATUR, though the first two are SHC for this algorithm. Therefore we claim the following

Corollary 4.9. *A smallest HC graph for the DSATUR algorithm contains at least* 8 *vertices.*

Now let us focus on hard-to-color graphs for DSATUR.

Lemma 4.10. *The graph G of Fig.* 1(b) *is HC for the DSATUR algorithm.*

Proof. At first, algorithm DSATUR will color two adjacent vertices of degree 4 with colors 1 and 2. After that two non-adjacent vertices of degree 2 will have the same saturation degree 2, so they will be colored next with color 3. Finally, DSATUR will color the remaining vertices with colors $1, 2, 3$ and 4. Consequently, DSATUR (G) is always 4, though $\chi(G) = 3$. □

The following proposition is obvious.

Proposition 4.11. *If v is pendant in an HC graph G for the DSATUR algorithm then $G - v$ is SHC for DSATUR.*

Proposition 4.12. *No graph with a vertex of degree* 1 *or two adjacent vertices of degree* 2 *is smallest HC for the DSATUR algorithm.*

Proof. Let G be a smallest HC graph for DSATUR. Let us first consider the case that v is pendant in G. Then by Proposition 4.11 $G - v$ must be SHC. But there are only 3 SHC graphs with 7 vertices and none of them becomes HC after attachment to a pendant edge. So let us consider the case that G has an edge e incident with two vertices of degree 2. Then $G - e$ remains HC for DSATUR, a contradiction to the hypothesis that G is the smallest. □

Proposition 4.13. *No graph smaller than that of Fig.* 1(b) *is HC for the DSATUR algorithm.*

Proof. Suppose by the way of contradiction that G is such a graph. Obviously, it must have $n = 8$ vertices and $m \leqslant 11$ edges. Since it cannot have a bridge so by Proposition 4.4 G must have a vertex v of degree 2 such that $G - v$ is SHC for SLF. However, SLF has only 4 non-isomorphic SHC graphs with 7 vertices and 9 edges: three obtained by attaching a pendant edge to vertices 1, 2 and 5 of the graph in Fig. 4(a) and the fourth one shown in Fig. 6(a). Clearly, if a graph has a leaf than any attachment of

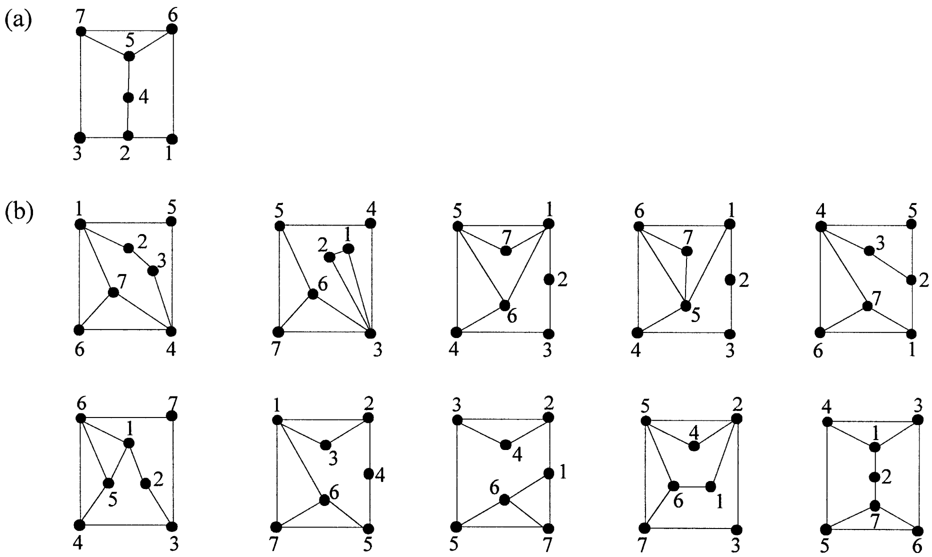


Fig. 6. Sparse 7-vertex SHC graphs for algorithm SLF: (a) with $m=9$; (b) with $m=10$.

a new vertex adjacent to two old vertices results in a supergraph with a leaf or two adjacent vertices of degree 2. By Proposition 4.12 such a graph cannot be smallest HC for DSATUR. Thus the only graph to consider is that of Fig. 6(a). The reader can verify that no attachment of a vertex of degree 2 to it results in an HC graph for DSATUR. □

Now we present the main result of this paper.

Theorem 4.14. *The graph of Fig.* 1(b) *is the unique smallest HC graph for the DSATUR algorithm.*

Proof. Suppose that the graph of Fig. 1(b) is not unique and let G be a smallest HC graph for DSATUR which is different from that of Fig. 1(b). Thus, G must have 8 vertices and 12 edges. By Proposition 4.12, there are two cases to consider.

Case 1. $\delta(G)=2$. Since G is bridgeless, by Proposition 4.4 there exists v such that $G-v$ is SHC for SLF, where $\deg(v)=2$. However, there are 11 7-vertex SHC graphs for SLF with at most 10 edges and minimum degree of at least 2. They are shown in Fig. 6. None of them can be augmented to a graph which is different from that of Fig. 1(b) and HC for DSATUR.

Case 2: $\delta(G)=3$. Such a graph must be cubic, 3-chromatic and Hamiltonian (a smallest non-Hamiltonian cubic graph is the Petersen graph). There are only four ways to complete C_8 with 4 edges in such a way that the resulting graph meets these presumptions, as shown in Fig. 7. It is easy to see that no graph of Fig. 7 is HC for DSATUR. □

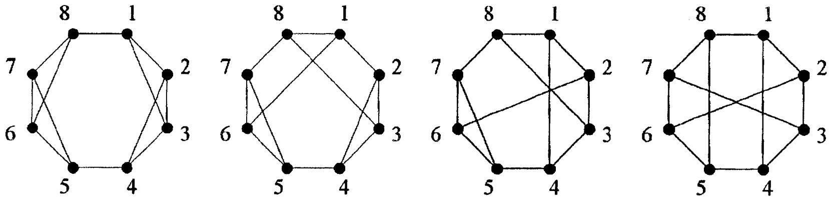
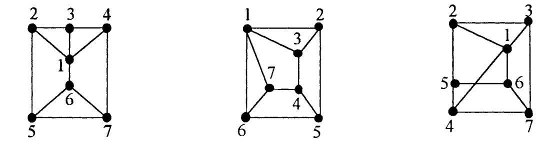


Fig. 7. Cubic 3-chromatic graphs for Theorem 4.14.

Appendix

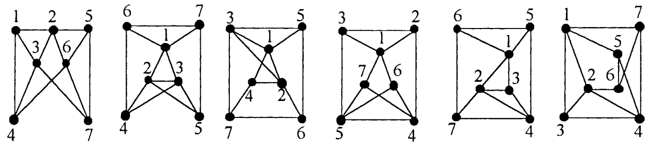
7-vertex graphs satisfying inequalities $3 \leqslant \chi(G) \leqslant \Delta(G) \leqslant 4, 3 \leqslant \delta(G) \leqslant 4$ and the conditions of Proposition 4.1. The vertices are numbered according to optimal DSATUR sequences.

m=11, χ=3

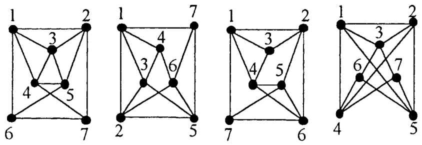


SHC: (1,3,2,4,6,5,7) SHC: (1,2,3,7,6,5,4)

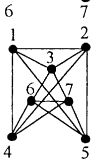
m=12, χ=3



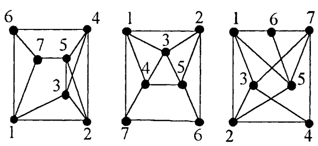
m=13, χ=3



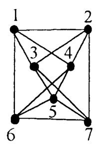
m=14, χ=3



m=13, χ=4



m=14, χ=4



References

[1] L. Babel, G. Tinhofer, Hard-to-color graphs for connected sequential colorings, Ann Discrete Math. 51 (1994) 3–25.
[2] D. Brélaz, New methods to color the vertices of a graph, Comm. ACM 22 (1979) 251–256.
[3] R.L. Brooks, On colouring the nodes of a network, Proc. Cambridge Philos. Soc. 37 (1941) 194–197.
[4] W. Głazek, M. Kubale, K. Manuszewski, The quest for small benchmarks for the chromatic sum problem, Arch. Cont. Sci. 7 (1998) 25–36.
[5] P. Hansen, J. Kuplinsky, Slightly hard-to-color graphs, Congr. Numer. 78 (1990) 81–98.
[6] P. Hansen, J. Kuplinsky, The smallest hard-to-color graph, Discrete Math. 96 (1991) 199–212.
[7] M. Kubale, J. Pakulski, A catalogue of the smallest hard-to-color graphs, Proceedings of the International Conference on Operations Research'94, Berlin, 1994, pp. 133–138.
[8] M. Kubale, J. Pakulski, K. Piwakowski, The smallest hard-to-color graph for the SL algorithm, Discrete Math. 164 (1997) 197–212.
[9] E.C. Sewell, An improved algorithm for exact graph coloring, in: D.S. Johnson, M.A. Trick (Eds.), Cliques, Coloring and Satisfiability, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, 1996, pp. 359–373.
[10] J.P. Spinrad, G. Vijayan, Worst case analysis of a graph coloring algorithm, Discrete Appl. Math. 12 (1985) 89–92.
[11] J.S. Turner, Almost all k-colorable graphs are easy to color, J. Algorithms 9 (1988) 63–82.