University of Ljubljana
Institute of Mathematics, Physics and Mechanics
Department of Mathematics
Jadranska 19, 1 111 Ljubljana, Slovenia

Preprint series, Vol. 43 (2005), 988

2-LOCAL 7/6-COMPETITIVE ALGORITHM FOR MULTICOLORING A SUB-CLASS OF HEXAGONAL GRAPHS

Petra Šparl Janez Žerovnik

ISSN 1318-4865

September 8, 2005

Ljubljana, September 8, 2005

2-local 7/6-competitive algorithm for multicoloring a sub-class of hexagonal graphs

Petra Šparl
University of Maribor
FCE, Smetanova 17
2000 Maribor
SLOVENIA
petra.sparl@uni-mb.si

Janez Žerovnik
University of Maribor
FME, Smetanova 17
2000 Maribor
SLOVENIA
janez.zerovnik@uni-lj.si

Abstract

An important optimization problem in the design of cellular networks is to assign sets of frequencies to transmitters to avoid unacceptable interference. A cellular network is often modeled as a subgraph of the infinite triangular lattice. The distributed frequency assignment problem can be abstracted as a multicoloring problem on a weighted hexagonal graph, where the weight vector represents the number of calls to be assigned at vertices. In this paper we present a 2-local distributed algorithm for multicoloring triangle-free hexagonal graphs with no adjacent centers and with arbitrary demands. The algorithm is using only the local clique numbers at each vertex v of the given hexagonal graph, which can be computed from local information available at the vertex. We prove that the algorithm uses no more than $\lceil 7\omega(G)/6 \rceil + 5$ colors for any triangle-free hexagonal graph with no adjacent centers G, without explicitly computing the global clique number $\omega(G)$. Hence the competitive ratio of the algorithm is 7/6.

keywords: approximation algorithm, graph coloring, frequency planning, cellular networks, 2-local distributed algorithm

1 Introduction

A basic problem concerning cellular networks is to assign sets of frequencies (colors) to transmitters (vertices) to avoid unacceptable interference [1]. The number of frequencies demanded at a transmitter may vary between transmitters. In a usual cellular model, transmitters are centers of hexagonal cells and the corresponding adjacency graph is a subgraph of the infinite triangular lattice. An integer $d(v)$ is assigned to each vertex of the triangular lattice and will be called the *demand* of the vertex v. A *hexagonal graph* $G(V, E, d)$ is the vertex weighted graph induced on the subset of the triangular lattice of vertices with positive demand. Hexagonal graphs arise naturally in studies of cellular networks. A *proper multicoloring* of G is a mapping f from $V(G)$ to subsets of integers such that $|f(v)| \geq d(v)$ for any vertex $v \in G$ and $f(v) \cap f(u) = \emptyset$ for any pair of adjacent vertices u and v in the graph G. The minimal cardinality of a proper multicoloring of G, $\chi(G)$, is called the *multichromatic number*. Another invariant of interest in this context is the *(weighted) clique number*, $\omega(G)$, defined as follows: The weight of a clique of G is the sum of demands on its vertices and $\omega(G)$ is the maximal clique weight on G. Clearly, $\chi(G) \geq \omega(G)$. Recently, the bound $\chi(G) \leq (4/3)\omega(G)+C$ was independently proved by several authors [5, 6, 10]. All proofs are constructive thus implying the existence of 4/3-approximation algorithms. McDiarmid and Reed [5] also show that it is NP-complete to decide whether $\chi(G) = \omega(G)$. A distributed algorithm which guarantees the $\lceil (4/3)\omega(G) \rceil$ bound is reported by Narayanan and Shende [6, 7]. A framework for studying distributed online assignment in cellular networks was developed in [4]. In particular, competitive ratios of distributed algorithms which utilize information about increasingly larger neighborhoods are addressed. The best competitive ratios for 0-,1-,2- and 4-local algorithms reported are 3, 3/2, 17/12 and 4/3, respectively. An algorithm is k-local if the computation at a vertex v uses only information about the demands of vertices whose graph distance from v is less than or equal to k. A 2-local algorithm for multicoloring of hexagonal graphs which uses at most $\lceil (4/3)\omega(G) \rceil$ colors is given in [9].

Better bounds can be obtained for triangle-free hexagonal graphs: [3] provides a distributed algorithm with competitive ratio 5/4, later a distributed algorithm with competitive ratio 6/5 is given [11], while an inductive proof for 7/6 ratio is reported in [2]. McDiarmid and Reed conjectured that for triangle free hexagonal graphs the inequality $\chi(G) \leq (9/8)\omega(G) + C$ holds

[5]. If a graph is triangle-free, there is no set of three mutually adjacent vertices of positive demand. It is easy to see that the smallest induced odd cycle in this case is of length 9, hence the constant 9/8, which is the best possible ratio on C_9.

In this paper we give a 2-local algorithm for multicoloring triangle-free hexagonal graphs with no adjacent centers which uses at most $\lceil (7/6)\omega(G) \rceil + 5$ colors. As the algorithm is 2-local, no global information is assumed to be available. A vertex can initially communicate to its neighbors to obtain some local information. It is important to note that the computation time does not depend on the size of the graph. We also define the local clique number $\omega_1(v)$ at each vertex v of the graph, which can, by definition, be computed from local information available at the vertex. Directly from the definition it will follow that

$$\omega(G) = \max_{v \in G} \omega_1(v)$$

and hence the algorithm presented here will use no more than $\lceil (7/6)\omega(G) \rceil + 5$ colors, without explicitly computing the $\omega(G)$. We will also prove that the algorithm is 2-local. More formally, we will prove that

Theorem 1 *There is a 1-local distributed approximation algorithm for 7-[3]coloring of a triangle-free hexagonal graph with no adjacent centers. Time complexity of the algorithm at each vertex is constant.*

Theorem 2 *There is a 2-local distributed approximation algorithm for multicoloring of a triangle-free hexagonal graph with no adjacent centers which uses at most* $\lceil (7/6)\omega(G) \rceil + 5$ *colors. Time complexity of the algorithm at each vertex is constant.*

This is a partial answer to the open question whether a 7/6-competitive distributed algorithm for multicoloring triangle-free hexagonal graphs exists.

The paper is organized as follows. In the next section we formally define some basic terminology. In Section 3 the 1-local algorithm for 7-[3]coloring triangle-free hexagonal graphs with no adjacent centers and its correctness proof are given. In Section 4 we present the 2-local algorithm for multicoloring triangle-free hexagonal graphs with no adjacent centers with arbitrary demands.

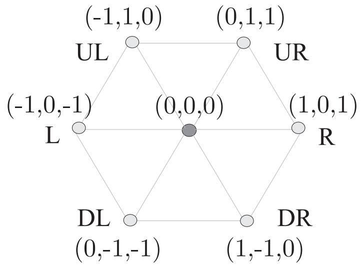

Figure 1: Definition of coordinates (i, j, k) on triangular lattice.

2 Preliminaries

A *weight function* on a graph G is a function from $V(G)$, the set of vertices of G, into the set of non-negative integers. Let p be a weight function on a graph G. An n-$[p]$*coloring* of G is a mapping f from $V(G)$ into the set of subsets of $\{1, 2, ..., n\}$, such that $|f(v)| = p(v)$ for every $v \in G$ and for any two adjacent vertices u and v of G, $f(u) \cap f(v) = \emptyset$. Accordingly, a 7-[3]coloring of G, which will be considered in Section 3, is a mapping $f : V(G) \to \mathcal{P}(\{1, 2, 3, 4, 5, 6, 7\})$, where $|f(v)| = 3$ for every $v \in G$ and $f(u) \cap f(v) = \emptyset$ for every $uv \in E(G)$. Here $\mathcal{P}(S)$ denotes the collection of all subsets of S.

Following [5], the vertices of triangular lattice can be represented as linear combination $x\vec{p} + y\vec{q}$ of the two vectors $\vec{p} = (1, 0)$ and $\vec{q} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$. Thus, we may identify vertices of triangular grid with pairs (x, y) of integers. Two vertices are adjacent when the Euclidean distance between them is one. Therefore each vertex (x, y) has six neighbors $(x \pm 1, y)$, $(x, y \pm 1)$, $(x + 1, y - 1)$ and $(x - 1, y + 1)$. For simplicity, we will refer to the neighbors as R (right), L (left), UR (up-right), DL (down-left), DR (down-right) and UL (up-left), respectively (see Fig. 1). For the 2-local implementation we introduce a three dimensional coordinate system $(i, j, k) := (x, y, x + y)$, where (x, y) are the coordinates as defined above (for details, see [8] or [9]).

In particular this enables a definition of the *parity* of a vertex with respect to a line given below. Note that the triangular lattice is composed of three sets of parallel straight lines.

We omit a straightforward proof of the following proposition:

Proposition 3 *The following statements hold:*
- each line which goes from bottom-left to top-right has the first coordinate,

i, constant,
- each horizontal line has the second coordinate, j, constant,
- each line which goes from top-left to bottom-right has the third coordinate, k, constant.

Definition 4 *The* parity *of a vertex $v \in G$.*
A vertex v with coordinates (i, j, k)
- is odd *with respect to its L neighbor with coordinates $(i-1, j, k-1)$ if $i \equiv 1 (\mathrm{mod}\, 2)$,*
- is even *with respect to its L neighbor with coordinates $(i-1, j, k-1)$ if $i \equiv 0 (\mathrm{mod}\, 2)$,*
- is odd *with respect to its R neighbor with coordinates $(i+1, j, k+1)$ if $i \equiv 1 (\mathrm{mod}\, 2)$,*
- is even *with respect to its R neighbor with coordinates $(i+1, j, k+1)$ if $i \equiv 0 (\mathrm{mod}\, 2)$.*

Similarly: a vertex v with coordinates (i, j, k) is *odd* [*even*] with respect to its UL neighbor with coordinates $(i-1, j+1, k)$ (DR neighbor with coordinates $(i+1, j-1, k)$) if $j \equiv 1 (\mathrm{mod}\, 2)$ [$j \equiv 0 (\mathrm{mod}\, 2)$]. A vertex v with coordinates (i, j, k) is *odd* [*even*] with respect to its UR neighbor with coordinates $(i, j+1, k+1)$ (DL neighbor with coordinates $(i, j-1, k-1)$) if $k \equiv 1 (\mathrm{mod}\, 2)$ [$k \equiv 0 (\mathrm{mod}\, 2)$].

Note that the parity of v with respect to its L (UL, DL) neighbor is the same as the parity with respect to its R (DR, UR) neighbor. Hence, we may also think about the parity of v with respect to a line.

There are two different colorings of the infinite triangular lattice which give rise to a partition of the vertex set of any hexagonal graph into independent sets. The first one is a 3-coloring, which gives three independent sets *Red*, *Blue* and *Green*, such that if x is in Red (resp. Blue or Green) set then its right neighbor is in Blue (resp. Green or Reed) set. According to this partition each vertex has its *base color,* namely *red* (R)*, blue* (B) or *green* (G). Let $c(v)$ stand for the base color of vertex $v \in G$. For the later reference let us define $R := -2$, $B := -1$ and $G := 0$. The second coloring is a 4-coloring, which gives four independent sets $1, 2, 3$ and 4 (see Fig. 2). According to this partition each vertex $v \in G$ has its *base number* denoted by $n_1(v)$.

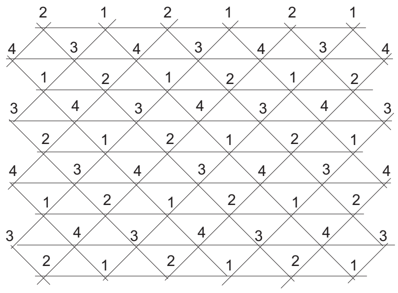

Figure 2: A 4-coloring of a triangular lattice.

Definition 5 *For every vertex v in triangle-free hexagonal graph G let*

$$\omega_1(v) = \max\{d(v) + d(u) \mid u \textit{ is a neighbor of } v \textit{ in } G\}$$

be the 1-local clique number of the vertex v and $p_1(v) = \left\lceil \frac{\omega_1(v)}{6} \right\rceil$.

Definition 6 *Vertex $v \in G$ is* heavy *if $d(v) > 3p_1(v)$ and v is* light *if $d(v) \le 3p_1(v)$.*

Definition 7 *Vertex $v \in G$ is a* center *if it has at least two neighbors in G, which are not in the same line.*

Note that all neighbors of a center have the same base color and pairwise different base number. The following lemmas follow directly from Definition 5.

Lemma 8 *For every vertex $v \in G$, $\omega_1(v) \le 6p_1(v)$.*

Lemma 9 *For arbitrary neighbors u and v, $d(u)+d(v) \le \min\{\omega_1(u), \omega_1(v)\}$.*

Lemma 10 *For arbitrary neighbors u and v of G we have*

$$d(u) + d(v) \le 6 \min\{p_1(u), p_1(v)\}.$$

3 1-local 7-[3]coloring

We will describe a 1-local algorithm which 7-[3]colors every hexagonal graph without triangles G. We assume that each vertex knows its position (coordinates (i, j, k)) and the coordinates of its neighbors. The algorithm will use three base colors R, B and G and four base numbers $1, 2, 3$ and 4.

Recall that $c(v)$ and $n_1(v)$ mean the base color and the base number of vertex $v \in G$ respectively. Note that if the base color of $v \in G$ is $c(v)$, then the base color of its neighbor is either $c(v) + 1 (\bmod 3)$ or $c(v) - 1 (\bmod 3)$. Note that vertices of each straight line of the triangular lattice have only two different base numbers which alternate (for example, there is a straight line whose vertices have base numbers equal to 1 or 2 alternatively). The base color and the base number of a vertex $v \in G$ can be computed from the coordinates, for example, using the rules $c(v) = i + 2j (\bmod 3)$ and $n_1(v) = i (\bmod 2) + 2(j (\bmod 2)) + 1$ for vertex v with coordinates (i, j, k).

3.1 The 7-[3]coloring algorithm

Every vertex of the algorithm will get a subset of three colors from the set of seven colors labeled by -2 (*red*), -1 (*blue*), 0 (*green*), $1, 2, 3$ and 4. The 3-coloring of vertices depends on the fact whether a vertex is a center or not. Suppose that vertex v is not a center and is not isolated. Therefore v is on a straight line. Let $n_2(v)$ stand for the base number of neighbors of v. Recall that both neighbors of v (if it has two of them) have the same base number. Let define the set of numbers $n(v) := \{1, 2, 3, 4\} \backslash \{n_1(v), n_2(v)\}$ for a vertex v of G called *free numbers* of v.

For the easier understanding of the algorithm let say that for vertex $v \in G$ the base color, which is not equal neither to base color of v nor to base color of its neighbors is called a *free color* of v. For example, if v is a red center with blue neighbors, its free color is green (it is not difficult to see that vertex $v \in G$ with two neighbors in the same line has no free color).

The principle of the 3-coloring is the following: every vertex $v \in G$ gets its base color, its base number and borrow either its free color or one of its free numbers.

Case 1: **(**v **is a center or isolated;** *v borrows its free color***):**
IF $c(u) = c(v) + 1 (\bmod 3)$ for every neighbor u of v or v is isolated THEN

v gets colors $\{c(v), n_1(v), c(v) - 1 (\bmod 3)\}$
ELSE (for every neighbor u of v, $c(u) = c(v) - 1 (\bmod 3)$)
v gets colors $\{c(v), n_1(v), c(v) + 1 (\bmod 3)\}$,

Case 2: **(v is not a center,** *v borrows one of its free numbers***):**
IF v is odd an a line THEN
v gets colors $\{n_1(v), \min\{n(v)\}, c(v)\}$,
IF v is even an a line THEN
v gets colors $\{n_1(v), \max\{n(v)\}, c(v)\}$.

3.2 Correctness proof

Let us first argue that the algorithm is 1-local:
- it is 1-local to decide if v is on a line and to compute its parity,
- it is 1-local to decide if v is a center,

The details are straightforward and we omit them.

We will show now that the coloring is proper. Note that the violations of the properness of the coloring can occur only between two neighbors of G. Therefore, we will consider all possible cases where two vertices are neighbors in G.

1. Let x be from the Case 1. If x is isolated then no conflicts can occur. Let x be a center. In this case x can have a neighbor y which falls into the:

   (1a) Case 1. This is not possible, because we consider only graphs with no adjacent centers.

   (1b) Case 2. Let suppose that $c(y) = c(x) + 1 (\bmod 3)$. Then x gets colors $\{c(x), n_1(x), c(x) - 1 (\bmod 3)\}$ and y may be assigned either colors $\{n_1(y), \min\{n(v)\}, c(y)\}$ or $\{n_1(y), \max\{n(v)\}, c(y)\}$. Using $n_1(x) \neq n_1(y)$, we see that the assigned sets of colors of vertices x and y have an empty intersection. When $c(y) = c(x) + 1 (\bmod 3)$ similar arguments shows that the assigned sets of colors of vertices x and y are disjoint.

2. Let x be from the Case 2. In this case x can have a neighbor y which falls into the:

   (2a) Case 1. This is already discussed in (1b).

(2b) Case 2. Without loss of generality we may suppose that x is odd on a line (this implies that y is even on a line). Then x and y get colors $\{n_1(x), \min\{n(x)\}, c(x)\}$ and $\{n_1(y), \max\{n(y)\}, c(y)\}$, respectively. As $n(x) = n(y)$ the assigned sets of colors have an empty intersection.

Since any two neighboring vertices get disjoint sets of colors, the 7-[3]coloring is proper.

4 2-local $[p]$coloring

The set of colors at every vertex v of G will consists of $7p_1(v)$ colors. For technical reasons colors will be divided into seven subsets called *palettes.* Each of the palettes has $p_1(v)$ *hues*, namely there are $p_1(v)$ hues of every base color (red, blue and green) and $p_1(v)$ hues of every base number (1, 2, 3 or 4). Let c stand for $-2, -1, 0, 1, 2, 3$ or 4. Some subsets of hues of the c palette at vertex $v \in G$ can be identified with sets $(c, [i, j])$, where $i \leq j$, $1 \leq i$ and $j \leq p_2(v)$. We will say that we start the c palette from *low* if we start by the color $(c, 1)$ and continue upwards. Similarly we start the c palette from *high* if we start by the color $(c, p_2(v))$ and continue downwards.

The 7-[3]coloring will be used to define a proper $[p]$coloring with at most $\lceil (7/6)\omega(G) \rceil + 5$ colors (Theorem 2). Let us denote the colors assigned by 7-[3]coloring to a vertex $v \in G$ as $c_1(v)$, $c_2(v)$ and $c_3(v)$.

4.1 The $[p]$coloring algorithm

The algorithm will consist of three steps. The first one (Step 0) is the initialization step, while the second one (Step 1) performs the 7-[3]coloring of an input graph G, which is needed to define the order of color palettes in the main step (Step 2). The input of the algorithm is a triangle-free weighted hexagonal graph with no adjacent centers, where every vertex knows its own demand and demands of all its neighbors.

Step 0: For each vertex $v \in G$ compute $p_1(v)$.

Step 1: Apply the algorithm for 7-[3]coloring (to define the order of color palettes in the Step 2).

Step 2: Every vertex $v \in G$ first takes all colors from palettes $c_1(v), c_2(v)$ and $c_3(v)$ one after another, always starting the palette from low. Vertex can take colors from the following palette only when it has taken all colors from the previous one (first of all v takes colors from palette $c_1(v)$, starting the palette from low, as long the palette is empty, then v takes colors from palette $c_2(v)$, again starting from low and so on.) Let $C = \{c_1(v), c_2(v), c_3(v)\}$. If v is heavy, then v takes additional colors in turns from three palettes (starting the palette from high) as follows:

Case 1 : (v *is a center*)
IF $C = \{c(v), n_1(v), c(v) \mp 1 (\bmod 3)\}$ THEN v takes additional colors in turns from:
$c(v) \pm 1 (\bmod 3)$ palette (high) and (if needed) from $n_1(v)+1 (\bmod 4)$ palette (high) and (if needed) from $n_1(v)+2 (\bmod 4)$ palette (high).

Case 2: (v *is odd* [*even*] *on a line and has exactly one neighbor* u, *which is a center*)
IF $C = \{n_1(v), \min\{n(v)\}\ [\max\{n(v)\}], c(v)\}$ THEN v takes additional colors in turns from:
$\{r, b, g\} \backslash \{c(v), c(u)\}$ palette (high) and (if needed) $\max\{n(v)\}\ [\min\{n(v)\}]$ palette (high) and (if needed) $n_2(v)$ palette (high).

Case 3: (v *is odd [even] on a line between two centers*)
IF $C = \{n_1(v), \min\{n(v)\}\ [\max\{n(v)\}], c(v)\}$ THEN v takes additional colors in turns from:
$\max\{n(v)\}\ [\min\{n(v)\}]$ palette (high) and (if needed) $n_2(v)$ palette (high) and (if needed) $c(v) + 1\ (\bmod 3)$ palette (high).

Case 4: (v is *odd* [*even*] *on a line and its neighbor is not a center*)
IF $C = \{n_1(v), \min\{n(v)\}\ [\max\{n(v)\}], c(v)\}$ THEN v takes additional colors in turns from:
$c(v) + 1 (\bmod 3)$ palette (high) and (if needed) $c(v) + 2 (\bmod 3)$ palette (high) and (if needed) $n_2(v)$ palette (high).

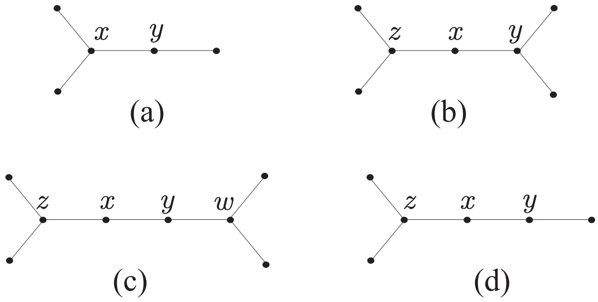

Figure 3: Possible neighbors of vertex x in G.

4.2 Correctness proof

Due to definition of a heavy vertex, it is obvious that only $d(v)$ and $p_1(v)$ are needed at vertex $v \in G$ to know if it is heavy or not. For knowing if vertex is a neighbor of a center it needs a 2-local information. Therefore, since 7-[3]coloring of G is 1-local, the $[p]$coloring of G is 2-local.

Let us show that the coloring is proper. Like in the 7-[3] case, we have to consider all possible cases of Step 2, where two vertices of G are neighbors. Recall that a vertex can take colors from the following palette only when it has taken all colors from the previous one. Moreover, it starts the first three palettes from low and continues upwards and, if heavy, it starts the second three palettes from high and continues downwards. The vertex stops by taking colors from palettes immediately after its demand is fulfilled. For brevity, we will omit writing $\pmod 3$ for base colors and $\pmod 4$ for base numbers in the following tables.

1. Let x be from the Case 1. In this case x can have a neighbor y, which falls into one of the following cases:

   (1a) Case 2 (see Fig. 3(a)). In this situation vertices x and y takes colors from palettes as follows: when $c(y) = c(x) - 1 \pmod 3$ then

x	y
$c(x)$	$n_1(y)$
$n_1(x)$	$\min\{n(y)\}$ $[\max\{n(y)\}]$
$c(x)+1$	$c(y)=c(x)-1$
$c(x)-1$	$\{r,b,g\}\backslash\{c(x),c(y)\}=c(x)+1$
$n_1(x)+1$	$\max\{n(y)\}$ $[\min\{n(y)\}]$
$n_1(x)+2$	$n_2(y)=n_1(x)$

otherwise ($c(y)=c(x)+1(\bmod 3)$)

x	y
$c(x)$	$n_1(y)$
$n_1(x)$	$\min\{n(y)\}$ $[\max\{n(y)\}]$
$c(x)-1$	$c(y)=c(x)+1$
$c(x)+1$	$\{r,b,g\}\backslash\{c(x),c(y)\}=c(x)-1$
$n_1(x)+1$	$\max\{n(y)\}$ $[\min\{n(y)\}]$
$n_1(x)+2$	$n_2(y)=n_1(x)$

Note that $n_1(x)\neq n_1(y)$ and $n_1(x)\notin\{n(y)\}$. By Lemma 10 we have $d(x)+d(y)\leq 6\min\{p_1(x),p_1(y)\}$. Therefore, following the fact that vertices x and y start the first three palettes from low and the second three palettes from high and the following palette may be started only when the previous one is emptied, no conflict can occur.

(1b) Case 3 (see Fig. 3 (b)). In this situation vertex y has another neighbor z which is also a center. Vertices x, y and z takes colors from palettes as follows: when $c(y)=c(x)-1(\bmod 3)$ (note that this implies $c(z)=c(y)-1\ (\bmod 3)$) then

x	y	z
$c(x)$	$n_1(y)$	$c(z)=c(y)-1$
$n_1(x)=n_2(y)$	$\min\{n(y)\}$ $[\max\{n(y)\}]$	$n_1(z)=n_2(y)$
$c(x)+1=c(y)-1$	$c(y)$	$c(z)-1=c(y)+1$
$c(x)-1=c(y)$	$\max\{n(y)\}$ $[\min\{n(y)\}]$	$c(z)+1=c(y)$
$n_1(x)+1$	$n_2(x)$	$n_1(z)+1$
$n_1(x)+2$	$c(y)+1$	$n_1(z)+1$

otherwise ($c(y)=c(x)+1(\bmod 3)$ and $c(z)=c(y)+1\ (\bmod 3)$)

x	y	z
$c(x) = c(y) - 1$	$n_1(y)$	$c(z) = c(y) + 1$
$n_1(x) = n_2(y)$	$\min\{n(y)\}$ $[\max\{n(y)\}]$	$n_1(z) = n_2(y)$
$c(x) - 1 = c(y) + 1$	$c(y)$	$c(z) + 1 = c(y) - 1$
$c(x) + 1 = c(y)$	$\max\{n(y)\}$ $[\min\{n(y)\}]$	$c(z) - 1 = c(y)$
$n_1(x) + 1$	$n_2(x)$	$n_1(z) + 1$
$n_1(x) + 2$	$c(y) + 1$	$n_1(z) + 1$

2. Let x be from the Case 2. Note that, by definition, x has a neighbor z which is a center. If z is the only neighbor of x which is a center, then sets of colors assigned to x and z can not cause any conflicts, see (1a). In this case x can have (beside neighbor z) a neighbor y which falls into the one of the following cases:

   (2a) Case 2. Note that, by definition, y has a neighbor w which is a center and $c(z) = c(w)$ (see Fig. 3(c)). In this situation x and y take colors from palettes as follows:
   When $c(y) = c(x) - 1 \pmod 3$ (note that this implies $c(x) = c(z) - 1 \pmod 3$) then

x	y
$n_1(x)$	$n_1(y) = n_2(x)$
$\min\{n(x)\}$ $[\max\{n(x)\}]$	$\max\{n(y)\}$ $[\min\{n(y)\}]$
$c(x)$	$c(y) = c(x) - 1$
$\{r, b, g\} \backslash \{c(x), c(u)\} = c(x) - 1$	$\{r, b, g\} \backslash \{c(y), c(v)\} = c(x)$
$\max\{n(x)\}$ $[\min\{n(x)\}]$	$\min\{n(y)\}$ $[\max\{n(y)\}]$
$n_2(x)$	$n_2(y) = n_1(x)$

   otherwise ($c(y) = c(x) + 1 \pmod 3$ and $c(x) = c(z) + 1 \pmod 3$)

x	y
$n_1(x)$	$n_1(y) = n_2(x)$
$\min\{n(x)\}$	$\max\{n(y)\} = \min\{n(y)\}$
$c(x)$	$c(y) = c(x) + 1$
$\{r, b, g\} \backslash \{c(x), c(u)\} = c(x) + 1$	$\{r, b, g\} \backslash \{c(y), c(v)\} = c(x)$
$\max\{n(x)\}$	$\min\{n(y)\}$ $[\max\{n(y)\}]$
$n_2(x)$	$n_2(y) = n_1(x)$

Following the equality $n(x) = n(y)$ we see that no conflict can occur between the assigned colors.

(2b) Case 4 (see Fig. 3(d)). In this situation x and y take colors from palettes as follows: when $c(y) = c(x) - 1 \pmod 3$ then

x	y
$n_1(x)$	$n_1(y) = n_2(x)$
$\min\{n(x)\}$ $[\max\{n(x)\}]$	$\max\{n(y)\}$ $[\min\{n(y)\}]$
$c(x)$	$c(y) = c(x) - 1$
$\{r, b, g\}\setminus\{c(x), c(u)\} = c(x) - 1$	$c(y) + 1 = c(x)$
$\max\{n(x)\}$ $[\min\{n(x)\}]$	$c(y) + 2 = c(x) + 1$
$n_2(x)$	$n_2(y) = n_1(x)$

otherwise $(c(y) = c(x) + 1 \pmod 3)$

x	y
$n_1(x)$	$n_1(y) = n_2(x)$
$\min\{n(x)\}$ $[\max\{n(x)\}]$	$\max\{n(y)\}$ $[\min\{n(y)\}]$
$c(x)$	$c(y) = c(x) + 1$
$\{r, b, g\}\setminus\{c(x), c(u)\} = c(x) + 1$	$c(y) + 1 = c(x) - 1$
$\max\{n(x)\}$ $[\min\{n(x)\}]$	$c(y) + 2 = c(x)$
$n_2(x)$	$n_2(y) = n_1(x)$

Similar as in the previous case no conflict can occur.

3. Let x be from the Case 3. This is already discussed in (1b).

4. Let x be from the Case 4. In this case x can have a neighbor y which falls into the Case 2. This is already discussed in (2b).

Since $d(x) + d(y) \leq 6 \min\{p_1(x), p_1(y)\}$, by Lemma 10, and the first three palettes at a vertex start by the low colors, meanwhile the second three palettes start by the high colors, no conflict can occur in neither of the above cases.

At every vertex $v \in G$ we used at most $7p_1(v)$ colors. Since (by simple calculation) we have $7p_1(v) = 7\left\lceil \frac{\omega_1(v)}{6} \right\rceil \leq \left\lceil \frac{7\omega_1(v)}{6} \right\rceil + 5$ and considering $\omega(G) = \max\{\omega_1(v) \mid v \in G\}$, our 2-local algorithm for multicoloring triangle-free hexagonal graphs with no adjacent centers uses at most $\left\lceil \frac{7\omega(G)}{6} \right\rceil + 5$ colors. The statement of Theorem 2 follows.

References

[1] W.K.Hale, Frequency Assignment, Proceedings of the IEEE 68 (1980), 1497-1514.

[2] F.Havet, Channel Assignment and Multicoloring of the Induced Subgraphs of the Triangular Lattice, Discrete Mathematics 233 (2001), 219-231.

[3] F.Havet and J.Žerovnik, Finding a Five Bicolouring of a Triangle-free Subgraph of the Triangular Lattice, Discrete Mathematics 244 (2002), 103-108.

[4] J.Janssen, D.Krizanc, L.Narayanan and S.Shende, Distributed Online Frequency Assignment in Cellular Networks, Journal of Algorithms 36 (2000), 119-151.

[5] C.McDiarmid and B.Reed, Channel Assignment and Weighted Colouring, Networks 36 (2000), 114-117.

[6] L.Narayanan and S.Shende, Static Frequency Assignment in Cellular Networks, Algorithmica 29 (2001) 396-410. Earlier version in Sirocco 97, (Proceedings of the 4th International Colloquium on structural information and communication complexity, Ascona, Switzerland), D.Krizanc and P.Widmayer (eds.), Carleton Scientific 1997, pp.215-227.

[7] L.Narayanan and S.Shende, Corrigendum to Static Frequency Assignment in Cellular Networks, Algorithmica 32 (2002), 697.

[8] P.Šparl, S.Ubeda and J.Žerovnik, Upper bounds for the span of frequency plans in cellular networks, International Journal of Applied Mathematics 9 (2002) no.2, 115-139.

[9] P.Šparl and J.Žerovnik, 2-Local 4/3-Competitive Algorithm For Multicoloring Hexagonal Graphs, Journal of Algorithms 55 (2005), 29-41.

[10] S.Ubeda and J.Žerovnik, Upper bounds for the span of triangular lattice graphs: application to frequency planing for cellular networks, Research report No. 97-28, ENS Lyon, September 1997.

[11] J. Žerovnik, A distributed 6/5-competitive algorithm for multicoloring triangle-free hexagonal graphs, Preprint series-Institute of Mathematics Physics and Mechanics, 42 (2004) No. 954, 1-12.