Simpler Multicoloring of Triangle-free Hexagonal Graphs

Ignasi Sau
CNRS, LIRMM
161, rue Ada, 34095 Cedex 5
Montpellier, FRANCE
ignasi.sau@lirmm.fr

Petra Šparl
University of Maribor
FOV, Kidričeva 55a
SI-4000 Kranj, SLOVENIA
petra.sparl@fov.uni-mb.si

Janez Žerovnik
University of Ljubljana
FS, Aškerčeva 6
1000 Ljubljana, SLOVENIA
janez.zerovnik@imfm.uni-lj.si

Abstract

Given a graph G and a demand function $p : V(G) \to \mathbb{N}$, a proper n-$[p]$*coloring* is a mapping $f : V(G) \to 2^{\{1,\dots,n\}}$ such that $|f(v)| \geq p(v)$ for every vertex $v \in V(G)$ and $f(v) \cap f(u) = \emptyset$ for any two adjacent vertices u and v. The least integer n for which a proper n-$[p]$coloring exists, $\chi_p(G)$, is called the *multichromatic number* of G. Finding the multichromatic number of induced subgraphs of the triangular lattice (called *hexagonal graphs*) has applications in cellular networks. The *weighted clique number* of a graph G, $\omega_p(G)$, is the maximum weight of a clique in G, where the *weight* of a clique is the total demand of its vertices. McDiarmid and Reed [8] conjectured that $\chi_p(G) \leq (9/8)\omega_p(G) + C$ for triangle-free hexagonal graphs, where C is some absolute constant. In this article we provide an algorithm to find a 7-[3]coloring of triangle-free hexagonal graphs (that is, when $p(v) = 3$ for all $v \in V(G)$), which implies that $\chi_p(G) \leq (7/6)\omega_p(G) + C$. Our result constitutes a shorter alternative to the inductive proof of Havet [5] and improves the short proof of Sudeep and Vishwanathan [13], who proved the existence of a 14-[6]coloring. All steps of our algorithm take time linear in $|V(G)|$, except for the 4-coloring of an auxiliary planar graph. The new techniques may shed some light on the conjecture of McDiarmid and Reed [8].

Keywords: graph algorithm, approximation algorithm, graph coloring, frequency planning, cellular networks.

1 Introduction

Given an induced subgraph $G = (V, E)$ of the triangular lattice (called a *hexagonal graph*) together with a demand function $p : V(G) \to \mathbb{N}$, a proper n-$[p]$*coloring* of G (also called *multicoloring*) is a mapping $f : V(G) \to 2^{\{1,\dots,n\}}$ such that $|f(v)| \geq p(v)$ for every vertex $v \in V(G)$ and $f(v) \cap f(u) = \emptyset$ for any two adjacent vertices u and v. The least integer n for which a proper n-$[p]$coloring exists, denoted by $\chi_p(G)$, is called the *multichromatic number* of G. Another invariant of interest in this context is the *(weighted) clique number*, denoted by $\omega_p(G)$, defined as follows: The weight of a clique of G is the sum of the demands of its vertices and $\omega_p(G)$ is the maximum clique weight in G. Clearly, $\chi_p(G) \geq \omega_p(G)$. The bound $\chi_p(G) \leq (4/3)\omega_p(G) + C$ [8–10, 15], for some absolute constant C, is still the best known for both distributed and non-distributed models of computation.

As stated originally by McDiarmid and Reed [8], the motivation for the study of multicoloring problems on hexagonal graphs was that hexagonal graphs arise naturally in studies of cellular networks, such as in the Philadelphia instances [12]. A fundamental problem concerning cellular networks is to assign sets of frequencies (colors) to transmitters (vertices) in order to avoid unacceptable interferences [4]. The number of frequencies demanded at a transmitter may vary between transmitters. A cellular network can be modeled in such a way that transmitters are centers of hexagonal cells and the corresponding adjacency graph is a subgraph of the infinite triangular lattice. An integer $p(v)$ is assigned to each vertex of the triangular lattice and is called the *demand* of vertex v. It should be noted that hexagonal graphs were indeed a good model for rural and for some early cellular networks, but on the other hand the networks in urban areas are usually much more complicated. As nowadays technology is constantly changing, the relation between the design of cellular networks and multicoloring of hexagonal graphs is now merely historical. However, it is not hard to imagine that call privacy could be required in some scenarios, and therefore a multicoloring model may be still relevant in some practical applications. Anyway, in the last decades this high technology application motivated a lot of mathematical work (cf. for instance [2, 5–9, 13, 15–17, 19, 20]), and some challenging problems remained open. We now proceed to discuss some of this work.

A framework for studying distributed online assignment in cellular networks was developed in [7], where 3/2-competitive 1-local, 17/12-competitive 2-local, and 4/3-competitive 4-local algorithms are outlined. Recall that an algorithm is *k-local* if the computation at any vertex v uses only information about the demands of vertices at distance at most k from v. Further, we say that an approximate algorithm for multicoloring is *r-competitive* if it yields the upper bound $\chi_p(G) \leq r\omega_p(G) + C$ for the multichromatic number of an arbitrary graph G, for some absolute constant C. Later, a 4/3-competitive 2-local algorithm was developed [15]. The best ratio for the 1-local case was first improved to 13/9 [2], later to 17/12 [18], 7/5 [19], and finally to 33/24 [20].

Better bounds can be obtained for triangle-free hexagonal graphs. The conjecture made by McDiarmid and Reed [8] is that $\chi_p(G) \leq (9/8)\omega_p(G) + C$ holds for triangle-free hexagonal graphs. In [6] a distributed algorithm for triangle free-hexagonal graphs with competitive ratio 5/4 is given. In [16] the authors report the existence of a 2-local distributed algorithm with competitive ratio 5/4, while an inductive proof for ratio 7/6 is reported in [5]. A 2-local 7/6-competitive algorithm for a sub-class of triangle-free hexagonal graphs is given in [17].

A special case of a proper multicoloring is when p is a constant function. For example, a 7-[3]coloring is an assignment of three colors between 1 and 7 to each vertex. In this paper we prove the following result.

Theorem 1 *There exists an algorithm for 7-[3]coloring triangle-free hexagonal graphs.*

The running time of the above algorithm is quadratic in the number of vertices of the input graph (cf. Section 4.2). Theorem 1 provides a shorter alternative proof to the inductive proof of Havet [5] and improves the short proof of [13] that implied the existence of a 14-[6]coloring. Note that in the case under study, we have that $\omega_p(G) = 6$. Using standard methods, one can derive from Theorem 1 the existence of an algorithm that uses at most $(7/6)\omega_p(G) + C$ colors, as we briefly discuss in Section 5.

The rest of the paper is organized as follows. In Section 2 we formally define some basic terminology. In Section 3 we present an algorithm for 7-[3]coloring an arbitrary triangle-free hexagonal graph G. The correctness of the algorithm is proved in Section 4.1 and its running time is discussed in Section 4.2. Finally, Section 5 concludes the article.

2 Preliminaries

The vertices of the triangular lattice can be represented as integer linear combinations $x\vec{p} + y\vec{q}$ of the two vectors $\vec{p} = (1, 0)$ and $\vec{q} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ of $\mathbb{R}^2$. Thus, we may identify the vertices of the 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 Figure 1(a).

There is a natural 3-coloring of the vertices of the infinite triangular lattice, which gives rise to the partition of the vertex set of any hexagonal graph into three independent sets *Red*, *Blue*, and *Green.* According to this partition, each vertex $v \in V(G)$ has its *base color,* namely *red* (**r**)*, blue* (**b**), or *green* (**g**), which is denoted by $c(v)$. Formally, we define

$$c(v) = (x + 2y) \pmod 3 + 1.$$

To avoid confusion, we define the constants $\mathbf{r} = -2$, $\mathbf{b} = -1$, $\mathbf{g} = 0$, and will use **r**, **b**, **g** when referring to the colors of the 7-coloring (with slight abuse of notation, our seven colors will range from -2 to 4), see Figure 1(b). We denote this 3-coloring by **rbg**-coloring.

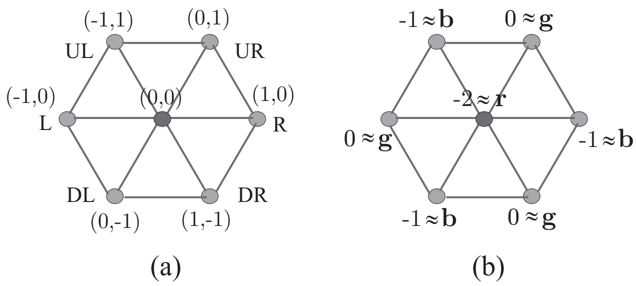

Figure 1: Coordinates and base colors.

Definition 2 *A vertex $v \in V(G)$ is a* right center *if it has at least two of its R, DL, and UL neighbors in G. Similarly, a vertex $v \in G$ is a* left center *if it has at least two of its L, DR, and UR neighbors in G.*

Note that in a triangle-free hexagonal graph, two centers of the same type (left or right) cannot be adjacent. Note also that each cycle contains centers of both types.

We need to introduce the following definitions.

Definition 3 *A vertex $v \in V(G)$ is* suitable *if it has neither L, UR, nor DR neighbor.*

Note that a suitable vertex is either a right center, has just one (R, UL, or DL) neighbor, or is an isolated vertex.

Definition 4 *Let c stand for red, blue, or green. A vertex v is c-*free *if its base color and the base colors of its neighbors are all different from c.*

By the above definitions and because we assume that there are no triangles, a center of a triangle-free graph is a c-free vertex. For example, a red right center and a blue left center are both green-free. Note that all neighbors of a center have the same base color.

Definition 5 *A path* $(x_1, x_2, \dots, x_s)$ *is called* left *(resp.* up-right, down-right*) if* x_{i+1} *is the left (resp. up-right, down-right) neighbor of* x_i *for* $i = 1, 2, \dots, s-1$*. A* tristar *is the union of one left* $(x, u_1, u_2, \dots, u_k)$*, one up-right* $(x, v_1, v_2, \dots, v_\ell)$*, and one down-right path* $(x, z_1, z_2, \dots, z_m)$*, for* $k, \ell, m \geq 0$*. The paths of a tristar will be called* rays*.*

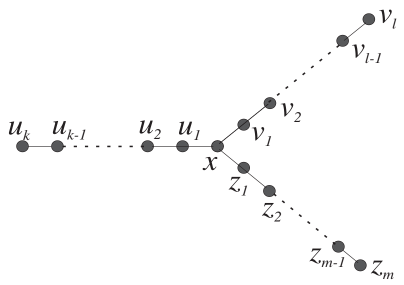

Figure 2: A tristar.

Note that a path and an isolated vertex are special cases of a tristar, where one or two of indices k, ℓ, and m equal to zero, or $k = \ell = m = 0$, respectively.

Let u_k, v_l, and z_m be the end-vertices of a tristar with common vertex x (see Figure 2). Note that there are only two possibilities for the parities of the lengths of the paths $(u_k, \dots, x, \dots, v_\ell)$, $(u_k, \dots, x, \dots, z_m)$, and $(v_\ell, \dots, x, \dots, z_m)$. Namely, either the lengths of all three paths are even (i.e., all three rays have the same length parity) or the lengths of two paths are odd and the length of the remaining path is even (i.e., the rays have different length parity).

3 The algorithm

We describe in this section an algorithm which 7-[3]colors a triangle-free hexagonal graph G. The algorithm uses seven colors, more precisely, the three base colors **r**, **b**, and **g** and four additional colors $1, 2, 3$, and 4. Recall that $c(v)$ denotes the base color of vertex $v \in V(G)$. Loosely speaking, the algorithm is composed of the following four steps:

- Assignment of the base **rbg**-coloring to the whole graph G;
- Partial coloring of suitable vertices (see Definition 3);
- Creation of an auxiliary graph, which is planar and thus 4-colorable;
- Extension of the obtained auxiliary graph coloring to the vertices with higher demand in such a way that the final 7-[3]coloring is proper.

We are now ready to give a precise description of the algorithm.

Input: A triangle-free hexagonal graph $G = (V, E, p)$, with constant demand $p(v) = 3$ for every $v \in V$ and given coordinates of vertices.
Output: A proper 7-[3]coloring of G.

Step 1: (rbg-coloring):
Assign the base **rbg**-coloring to the graph G. This reduces the demand $p(v)$ by one for every vertex $v \in V(G)$. Therefore, the new demand is equal to $p_1(v) = 2$ for every vertex $v \in V(G)$. Let $G_1 = (V, E, p_1)$.

Step 2: (Suitable vertices):

All suitable vertices are assigned the free color, i.e., a c-free suitable vertex is assigned color c. Hence the new demands are $p_2(v) = 1$ for suitable vertices and $p_2(v) = 2$ for any other vertex $v \in V$. The obtained graph is denoted by $G_2 = (V, E, p_2)$.

Remark 6 *Let S be a tristar in G_2 such that $p_2(v) = 2$ for every $v \in V(S)$, with rays $(x, u_1, u_2, \ldots, u_k)$, $(x, v_1, v_2, \ldots, v_\ell)$, and $(x, z_1, z_2, \ldots, z_m)$. Note that the only neighbors w of vertices of S in graph G_2 with $p_2(w) = 1$ can be vertices u (an L neighbor of u_k), v (an UR neighbor of v_ℓ), and z (a DR neighbor of z_m); see Figure 3. Indeed, in all other cases either vertices u_k, v_ℓ, or z_m are not the end-vertices of S with $p_2(\cdot) = 2$ or there is a contradiction with the assumption that G is triangle-free. Note that S can have none, one, two, or all three neighbors u, v, and z in G_2, and these neighbors must be right centers. A tristar S together with its neighbors is called an* extended tristar*, denoted by $\hat{S}$. Therefore, for an extended tristar $\hat{S}$ it holds $V(\hat{S}) \setminus V(S) \subseteq \{u, v, z\}$.*

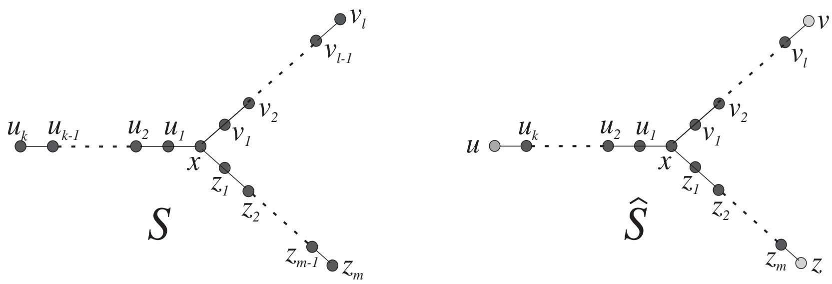

Figure 3: A tristar S and an extended tristar $\hat{S}$.

Step 3: (4-coloring of the vertices of demand 1 in G_2):

Create auxiliary graphs $\bar{G}_3 = (\bar{V}_3, \bar{E}_3)$ and $\widetilde{G} = (\widetilde{V}_3, \widetilde{E})$ as follows:

Let $\bar{V}_3 = \{v \in V(G) : p_2(v) = 1\}$. Note that there are no edges in G between any two vertices in $\bar{V}_3$.

Define the graph $\bar{G}_3$ by adding the following edges among the vertices in $\bar{V}_3$. For every tristar $S \subseteq G_2$ such that $p_2(v) = 2$ for every $v \in V(S)$:
(a) If S has three neighbors u, v, z in G_2 with $p_2(u) = p_2(v) = p_2(z) = 1$, and the lengths of two paths among $(u, \ldots, x, \ldots, v)$, $(u, \ldots, x, \ldots, z)$, and $(v, \ldots, x, \ldots, z)$ are odd, then connect the end-vertices of the paths of odd length (see the upper part of Figure 4).
(b) If S has two neighbors w_1, w_2 in G_2, with $p_2(w_1) = p_2(w_2) = 1$, then connect w_1 and w_2 if the length of the path between them is odd.

Finally, create the graph $\widetilde{G} = (\widetilde{V}_3, \widetilde{E})$ by identifying vertices of $\bar{V}_3$ using the following rule: For every tristar S such that $p_2(v) = 2$ for every $v \in V(S)$, having three neighbors u, v, z with $p_2(u) = p_2(v) = p_2(z) = 1$, do the following. If the lengths of all three paths $(u, \ldots, x, \ldots, v)$, $(u, \ldots, x, \ldots, z)$, and $(v, \ldots, x, \ldots, z)$ are even, then identify the L and DR neighbors (see the lower part of Figure 4).

Color vertices of $\widetilde{G}$ with four colors $\{1, 2, 3, 4\}$. Here we use the fact that $\widetilde{G}$ is a planar graph without loops (see Lemma 8), hence it is 4-colorable [1, 11].

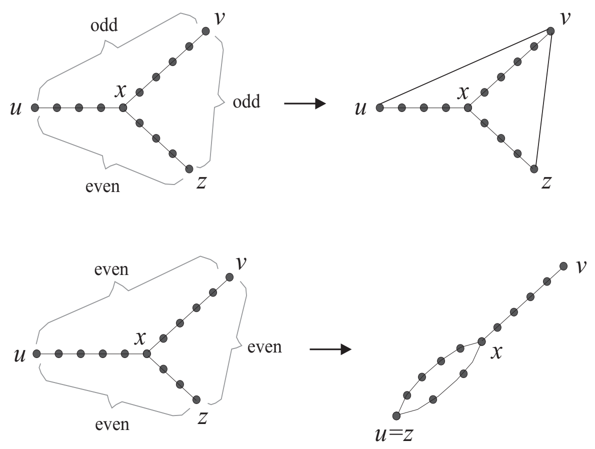

Figure 4: New edges and identifications.

The color assigned to vertex $v \in V(\widetilde{G})$ is denoted by $N(v)$. Note that this is at the same time a partial assignment to vertices of G_2. Hence the new demand is equal to $p_3(v) = p_2(v) - 1 = 0$ for every vertex $v \in \bar{V}_3$ and $p_3(v) = p_2(v) = 2$ for any other vertex v. Let $G_3 = (V, E, p_3)$.

Use the 4-coloring of $\widetilde{G}$ to assign one color from the set $\{1, 2, 3, 4\}$ to the vertices of G_3 in the natural way, that is, identified vertices receive the same color.

Step 4: (Extension of the obtained coloring):
Extend the assigned coloring (of Steps 1 up to 3) to the vertices of graph G that are not completely multicolored (that is, vertices v with $p_3(v) = 2$), in the following way. The only uncolored connected subgraphs induced on vertices v of demand $p_3(v) = 2$ are tristars (see Lemma 7). Using Lemma 9, extend the partial coloring of G_3 to multicolor uncolored tristars, using colors $\{1, 2, 3, 4\}$.

4 Correctness and running time

We prove in Section 4.1 the correctness of the algorithm and we analyze its running time in Section 4.2.

4.1 Correctness proof

We show in this section that the algorithm of Section 3 gives a proper 7-[3]coloring of an arbitrary triangle-free hexagonal graph G. After proving some useful facts, we continue with an outline of the proof following the structure of the algorithm.

Lemma 7 *Let G_4 be the graph induced on vertices of demand 2 in G_3. The connected components of G_4 are tristars.*

Proof. Since every right center was assigned two colors in Steps 1 and 2, there is no right center in G_4. Therefore, the connected components of the graph G_4 are tristars. Recall that paths and isolated vertices are special cases of tristars. ■

Lemma 8 *Graph $\widetilde{G}$ obtained in Step 3 of the algorithm is a planar graph without loops.*

Proof. The original graph G is an induced subgraph of a triangular lattice and thus planar. The identifications of vertices and the addition of new edges described in Step 3 cannot ruin the planarity of $\widetilde{G}$. Indeed, each of the new edges can be drawn following some induced paths without crossing existing edges (see the upper part of Figure 4), and no other edge is added following the same paths. On the other hand, each pair of identified vertices is linked by an induced path (see the lower part of Figure 4), so identifying the end-vertices of such a path does not create an edge crossing.

We shall now see that there are no loops in $\widetilde{G}$. A loop would appear if there were a pair of adjacent vertices in a set of vertices which are to be identified. But this is not possible because all identifications involve the end-vertex of the left ray and the end-vertex of the down-right ray of a tristar. Note that when end-vertices of a tristar are identified, they are never connected with edges in G_3. More precisely, recall that all identifications are done following the same rule that identifies an L and a DR neighbor of a tristar. Hence any set of identified vertices of G that was merged to a vertex in $\widetilde{G}$ must be a result of a chain of identifications that correspond to a path which visits L and DR neighbors of tristars, implying that this must be a path formed from R and UL rays of the tristars. Furthermore, we know that any such R-UL-R-UL-...-R-UL path is of even length and is a shortest path. As the edges of $\widetilde{G}$ can only connect vertices that are neighbors of a tristar at odd distance, no edge connecting two identified vertices is possible, and therefore there is no loop in $\widetilde{G}$. ■

Lemma 9 *Let $\hat{S} = S \cup \{u, v, z\}$ be an extended tristar of a tristar S in G_2 composed of three rays: left ray $(x, u_1, \ldots, u_k, u)$, up-right ray $(x, v_1, v_2, \ldots, v_\ell, v)$, and down-right ray $(x, z_1, z_2, \ldots, z_m, z)$ such that $p_2(u) = p_2(v) = p_2(z) = 1$ and $p_2(s) = 2$ for every vertex $s \in V(S)$. There exists a proper 4-[2]coloring with colors $\{1, 2, 3, 4\}$ of S that coincides with the coloring of vertices u, v, and z in Step 3.*

Proof. We will use the fact that a tristar is bipartite. Let V_1 and V_2 be the sets of the bipartition of the extended tristar $\hat{S}$. We distinguish two cases.

(1) First, if all three rays of $\hat{S}$ have lengths of the same parity, then all end-vertices are in the same set of the bipartition of the tristar, assume w.l.o.g. that $u, v, z \in V_1$. As the L and DR end-vertices were identified, the 4-coloring of Step 3 used at most 2 colors for the end-vertices. Hence the vertices of S that are in the set V_1 receive the set $\mathcal{S}$ of two colors, such that $N(u), N(v) \in \mathcal{S}$, and the vertices of the set V_2 are assigned the remaining two colors. If the tristar has only two rays, the reasoning is similar and simpler.

(2) In the second case, one or two rays of $\hat{S}$ are of even length, and the other lengths are odd. One of the end-vertices, which is at odd distance to the other two (w.l.o.g., say u), was connected to the other two end-vertices in Step 3. Hence, v and z are in one set (w.l.o.g., say V_1), and u is in the other set (V_2) of the bipartition of the tristar. Thus, the vertices of S that are in the set V_1 receive the set $\mathcal{S}$ of two colors, such that $N(v), N(z) \in \mathcal{S}$, and the vertices in the set V_2 are assigned the remaining two colors. If the tristar has only two rays, the reasoning is even easier. ■

We now proceed with the proof of the correctness of the algorithm.

Step 1: The **rbg**-coloring gives a proper 1-coloring of the graph G and reduces the demands of all vertices by 1, so there is nothing to prove.

Step 2: Assigning color c to a suitable c-free vertex is not in conflict with the previously assigned colors, since suitable vertices have no neighbors with base color c and two suitable vertices cannot be adjacent in the graph G.

Step 3: By Lemma 8, the graph $\widetilde{G}$ is planar without loops, and therefore it is properly 4-[1]colorable with colors $\{1, 2, 3, 4\}$. Since only the base colors $\{\mathbf{r}, \mathbf{b}, \mathbf{g}\}$ were used in previous steps, no conflicts can occur.

Step 4: By Lemma 7, the only connected components that are not completely colored are tristars. Let S be a tristar in G_4 and let V_1 and V_2 be the sets of the bipartition of S. Three cases may occur:
(a) S has three neighbors u, v, z in G_2, such that $p_2(u) = p_2(v) = p_2(z) = 1$. By Lemma 9, the partial coloring of u, v, and z can be properly extended to S.
(b) S has two neighbors u, v such that $p_2(u) = p_2(v) = 1$. If the length of the path between u and v is even, then u and v are in the same set of the bipartition of S, assume w.l.o.g. that they are in V_1. Hence, the vertices of S that are in the set V_1 receive the set $\mathcal{S}$ of two colors, such that $N(u), N(v) \in \mathcal{S}$, and the vertices in the set V_2 are assigned the remaining two colors. If the length between u and v is odd, then $N(u) \neq N(v)$ (because of Step 3(b)) and u and v are in different sets of the bipartition of S, assume w.l.o.g. that $u \in V_1$ and $v \in V_2$. In this case the vertices of S that are in the set V_1 receive the set of colors $\mathcal{S}$, which contains color $N(u)$ and one of the colors in $\{1, 2, 3, 4\} \backslash \{N(u), N(v)\}$. The vertices in V_2 receive the remaining two colors.
(c) S has at most one neighbor $v \in V(G)$ such that $p_2(v) = 1$. In this case the extension of the coloring is straightforward.

Note that since we use only the bipartition of a tristar S and an extended tristar $\hat{S}$, respectively, the same arguments of cases (b) and (c) hold also if the star S has at most two rays.

Accordingly, every vertex $v \in V(G)$ is assigned three different colors among $\{\mathbf{r}, \mathbf{b}, \mathbf{g}, 1, 2, 3, 4\}$, and adjacent vertices get disjoint sets of colors, as needed. Therefore, Theorem 1 follows.

4.2 Running time

Concerning the running time, one can check that all steps of our algorithm run in time linear in $|V(G)|$, except for the 4-coloring of the planar graph $\widetilde{G}$, which takes a priori quadratic time [11]. It has been recently proved that triangle-free planar graphs can be 3-colored in linear time [3]. Thus, if one could prove that $\widetilde{G}$ is triangle-free (or modify the construction in such a way that the constructed graph is triangle-free), the overall running time of the algorithm would be linear.

5 Concluding remarks

In this article we provided an algorithm for 7-[3]coloring triangle-free hexagonal graphs. The described 7-[3]coloring can be extended to a proper [p]coloring with at most $\lceil (7/6)\omega_p(G) \rceil + C$ colors of any weighted triangle-free hexagonal graph, for some absolute constant C. The main idea (used for example in [6, 7, 9, 16]) is to divide the set of colors into 7 palettes and to use the algorithm for 7-[3]coloring to define the order of color palettes from which vertices will take colors from. We omit the details here.

The *odd girth* $og(G)$ of a graph G is the length of a shortest cycle of odd length. It is easy to see (see for instance [8]) that for a triangle-free hexagonal graph G, $og(G) \geq 9$. The approximation ratio of our algorithm can be expressed more precisely with respect to $og(G)$ for some values of p. Indeed, by assigning demand $\frac{og(G)-1}{2}$ to all the vertices belonging to a cycle of length $og(G)$, one can check that $og(G)$ colors are needed to color the vertices of that cycle. Therefore, if $p(v) \leq \frac{og(G)-1}{2}$ for all $v \in V(G)$, $\frac{og(G)}{og(G)-1} \cdot \omega_p(G)$ is a lower bound on the number of needed colors. For instance, if $og(G) = 9$ and $p(v) \leq 4$ for all $v \in V(G)$, the approximation ratio of our algorithm becomes $\frac{(7/6)\omega_p(G)}{(9/8)\omega_p(G)} = \frac{28}{27}$, which improves over $7/6$.

It is worth mentioning that both the 5-[2]coloring of triangle-free hexagonal graphs [16] and the 4/3-competitive algorithm for multicoloring hexagonal graphs [15] are fully distributed. The 7-[3]coloring algorithm presented in Section 3 is not distributed, since it uses a 4-coloring of a planar graph. Nevertheless, most steps of the algorithm can be easily performed locally. Therefore, it is an interesting question whether one can find a coloring of our auxiliary graph $\widetilde{G}$ in a distributed way, using its structural properties.

Finally, it is a natural question to ask whether there exists an algorithm (distributed or not) for multicoloring an *arbitrary* hexagonal graph with approximation ratio $7/6$.

References

[1] K. Appel and W. Haken, *Every Planar Map is Four Colorable*, Illinois Journal of Mathematics **21**:439-567, 1977.

[2] F.Y.L. Chin, Y. Zhang, and H. Zhu, *A 1-local 13/9-competitive Algorithm for Multicoloring Hexagonal Graphs*, Proc. of the 13th Annual International Computing and Combinatorics Conference (COCOON), volume 4598 of LNCS, pp. 526-536, 2007.

[3] Z. Dvořák, K. Kawarabayashi, and R. Thomas, *Three-coloring triangle-free planar graphs in linear time*, Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1176-1182, 2009.

[4] W. K. Hale, *Frequency Assignment: Theory and Applications*, Proc. of the IEEE **68**(12):1497-1514, 1980.

[5] F. Havet, *Channel Assignment and Multicoloring of the Induced Subgraphs of the Triangular Lattice*, Discrete Mathematics **233**(1-3):219-231, 2001.

[6] F. Havet and J. Žerovnik, *Finding a Five Bicolouring of a Triangle-free Subgraph of the Triangular Lattice*, Discrete Mathematics **244**(1-3):103-108, 2002.

[7] J. Janssen, D. Krizanc, L. Narayanan, and S. Shende, *Distributed Online Frequency Assignment in Cellular Networks*, Journal of Algorithms **36**(2):119-151, 2000.

[8] C. McDiarmid and B. Reed, *Channel Assignment and Weighted Colouring*, Networks **36**(2):114-117, 2000.

[9] L. Narayanan and S. Shende, *Static Frequency Assignment in Cellular Networks*, Algorithmica **29**(3):396-410, 2001.

[10] L. Narayanan and S. Shende, *Corrigendum to Static Frequency Assignment in Cellular Networks*, Algorithmica **32**(4):697, 2002.

[11] N. Robertson, D. P. Sanders, P. D. Seymour, and R. Thomas, *The four-colour theorem*, Journal of Combinatorial Theory Series B **70**(1):2-44, 1997.

[12] D. H. Smith, S. Hurley and S. M. Allen, *A new lower bound for the channel assignment problem*, IEEE Transactions on Vehicular Technology **49**(4):1265-1272, 2000.

[13] K. S. Sudeep and S. Vishwanathan, *A technique for multicoloring triangle-free hexagonal graphs*, Discrete Mathematics **300**(1-3):256-259, 2005.

[14] P. Šparl, S. Ubeda, and J. Žerovnik, *Upper bounds for the span of frequency plans in cellular networks*, International Journal of Applied Mathematics **9**(2):115-139, 2002.

[15] P. Šparl and J. Žerovnik, *2-Local 4/3-Competitive Algorithm For Multicoloring Hexagonal Graphs*, Journal of Algorithms **55**(1):29-41, 2005.

[16] P. Šparl and J. Žerovnik, 2-local *5/4-competitive algorithm for multicoloring triangle-free hexagonal graphs*, Information Processing Letters **90**(5):239-246, 2004.

[17] P. Šparl and J. Žerovnik, *2-local 7/6-competitive algorithm for multicoloring a sub-class of hexagonal graphs*, International Journal of Computer Mathematics **87**(9):2003-2013, 2010.

[18] R. Witkowski, *A 1-local 17/12-competitive Algorithm for Multicoloring Hexagonal Graphs*, Proc. of Fundamentals of Computation Theory, volume 5699 of LNCS, pp. 346–356, 2009.

[19] R. Witkowski and J. Žerovnik, *A 1-local 7/5-competitive Algorithm for Multicoloring Hexagonal Graphs*, Electronic Notes in Discrete Mathematics **36**: 375-382, 2010.

[20] R. Witkowski and J. Žerovnik, *1-Local 33/24-Competitive Algorithm for Multicoloring Hexagonal Graphs*, Lecture Notes in Computer Science **6732**: 74-84, 2011.