1-local 4/3-competitive Algorithm for Multicoloring a Subclass of Hexagonal Graphs

Rafał Witkowski*
Adam Mickiewicz University,
Faculty of Mathematics and Computer Science,
Poznań, Poland
rmiw@amu.edu.pl

February 27, 2010

Abstract

In the frequency allocation problem we are given a cellular telephone network whose geographical coverage area is divided into cells where phone calls are serviced by frequencies assigned to them, so that none of the pairs of calls emanating from the same or neighboring cells is assigned the same frequency. The problem is to use the frequencies efficiently, i.e. minimize the span of used frequencies. The frequency allocation problem can be regarded as a multicoloring problem on a weighted hexagonal graph. In this paper we present a 1-local 4/3-competitive distributed algorithm for multicoloring a hexagonal graph without certain forbidden configuration (introduced in [7]). Its extension is also a new version of 2-local 4/3-competitive algorithm for multicoloring a general case hexagonal graphs.

1 Introduction

The basic problem concerning cellular networks concentrates on assigning sets of frequencies (colors) to transmitters (vertices) in order to avoid unacceptable interference (see [1]). In an ordinary cellular model the transmitters are centers of hexagonal cells and the corresponding adjacency graph is a subgraph of the infinite triangular lattice. In our model to each vertex v of a the triangular lattice T we assign a non-negative integer $d(v)$, called the *demand* (or *weight*) of the vertex v. A *proper multicoloring* of G is a mapping φ from $V(G)$ to subsets of integers (colors) $[n] = \{1, 2, \dots, n\}$, such that $|\varphi(v)| = d(v)$ for any vertex $v \in G$ and $\varphi(v) \cap \varphi(u) = \emptyset$ for any pair of adjacent vertices u and v in the graph G. The

*This work was supported by grant N206 017 32/2452 for years 2007-2010

minimal n for which there exists a proper multicoloring of G, denoted by $\chi_m(G)$, is called the *multichromatic number* of G.

In studies concerning cellular networks arise naturally the idea of hexagonal graphs. Formally, following the notation from [3], the vertices of the triangular lattice T can be described as follows: the position of each vertex is an integer linear combination $x\vec{p} + y\vec{q}$ of two vectors $\vec{p} = (1, 0)$ and $\vec{q} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$. Thus vertices of the triangular lattice may be identified 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: $(x-1, y), (x-1, y+1), (x, y+1), (x+1, y), (x+1, y-1), (x, y-1)$. For simplicity we refer to the neighbors as: *left*, *up-left*, *up-right*, *right*, *down-right* and *down-left*. We define a *hexagonal graph* $G = (V, E)$ as an induced subgraph of the triangular lattice (see Figure 1).

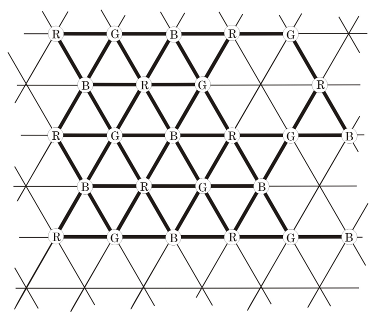

Figure 1: An example of a hexagonal graph

A *triangle-free hexagonal graph* is a subgraph of the triangular lattice which does not contain any 3-clique. A *corner* in a triangle-free hexagonal graph is a vertex which has at least two neighbors which are adjacent to it at angle different than π. A vertex which is not a corner is called a *non-corner* (see Figure 2).

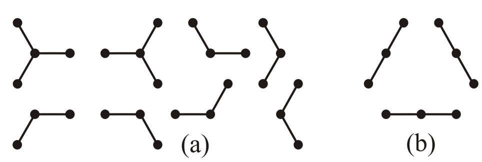

Figure 2: All possibilities for: (a) - corners, (b) - non-corners

The multichromatic number is closely related to the *weighted clique number* $\omega(G)$, which is defined as the maximum over all cliques of G of their weights, where the weight of a clique is the sum of demands on its vertices. Obviously, for any graph, $\chi_m(G) \geq \omega(G)$, while for hexagonal graphs, $\chi_m(G) \leq \left\lceil \frac{4\omega(G)}{3} \right\rceil + O(1)$ (see, for example, [3], [5], [6]). Since all proofs of the upper bound are constructive, therefore it implies the existence of a *4/3-competitive algorithm*, i.e. algorithms which can online serve calls with the approximation ratio equal to 4/3 respectively to the weighted clique number (see [2], [4]). It should be also mentioned, that McDiarmid and Reed showed in [3] that to decide whether $\chi_m(G) = \omega(G)$ is NP-complete.

In distributed graph algorithms a special role plays their "locality" property. An algorithm is *k-local* if the computation at any vertex v uses only the information about the demands of vertices at distance at most k from v. For hexagonal graphs the best known 1-local algorithm for multicoloring is 7/5-competitive, and it has been presented in [9].

In this paper we develop a new algorithm with ratio 4/3, which is 1-local for hexagonal graphs in which we exclude two adjacent heavy corners in triangle-free graph induced by heavy vertices (vertex is heavy if its weight is larger than average weights of all clique which contain this vertex) and its coordinates (x_1, y_1), (x_1, y_1) satisfy $x_1 \bmod 2 \neq y_1 \bmod 2$ and $x_2 \bmod 2 \neq y_2 \bmod 2$ (see Figure 3). We call such subgraph *special heavy double-corner.*

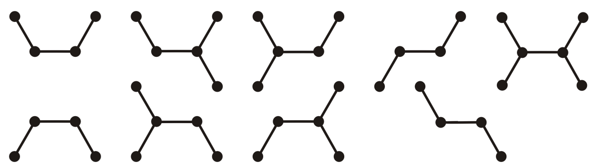

Figure 3: All possibilities for two horizontal adjacent corners (special heavy double-corner)

Triangle-free hexagonal graphs without two adjacent corners (heavy double-corners) were introduced in [7]. Here we consider a wider subclass of hexagonals than in [7] since we exclude two adjacent corners in one specific position only.

Our algorithm can be extended into 2-local model of computation to multicolor general case hexagonal graph with the same 4/3 ratio, as the best known 2-local algorithm proposed by Šparl and Žerovnik (see [6]) (Proof is in Appendix A).

We will prove the following theorem:

Theorem 1.1. *There is a 1-local distributed approximation algorithm for multicoloring hexagonal graphs without a special heavy double-corner which uses at most* $\left\lceil \frac{4}{3}\omega(G) \right\rceil + O(1)$ *colors. Time complexity of the algorithm at each vertex is constant.*

In [2] it was proved that a k-local c-approximate algorithm can be easily converted to a k-local c-competitive algorithm. Hence,

Corollary 1.2. *There is a 1-local 4/3-competitive algorithm for multicoloring hexagonal graphs without a special heavy double-corner.*

In the next Section we formally define some basic terminology while in Section 3 we present algorithm for multicoloring hexagonal graphs without a special heavy double-corner in 1-local model of computation. In this way we prove Theorems 1.1.

2 Basic definition and useful facts

There exists an obvious 3-coloring of the infinite triangular lattice which gives the partition of the vertex set of any hexagonal graph into three independent sets. Let us denote a color of any vertex v in this 3-coloring by $bc(v)$ and call it a *base color* (for simplicity we will use *red*, *green* and *blue* as base colors and their arrangement is given in Figure 1), i.e. $bc(v) \in \{R, G, B\}$.

Notice that in any weighted hexagonal graph G, a subgraph of the triangular lattice T induced by vertices with positive demands $d(v)$, the only cliques are triangles, edges and isolated vertices. Note also that we assume that all vertices of T which are not in G have to have demand $d(v) = 0$. Therefore, the weighted clique number of G can be computed as follows:

$$\omega(G) = \max\{d(u) + d(v) + d(t) : \{u, v, t\} \in \tau(T)\},$$

where $\tau(T)$ is the set of all triangles of T.

For each vertex $v \in G$, define *base function* κ as

$$\kappa(v) = \max\{a(v, u, t) : \{v, u, t\} \in \tau(T)\},$$

where $a(u, v, t) = \lceil (d(u)+d(v)+d(t))/3 \rceil$, is an average weight of the triangle $\{u, v, t\} \in \tau(T)$. It is easy to observe that the following fact holds.

Fact 2.1. *For each $v \in G$,*

$$\kappa(v) \leq \left\lceil \frac{\omega(G)}{3} \right\rceil$$

We call vertex v *heavy* if $d(v) > \kappa(v)$, otherwise we call it *light*. If $d(v) > 2\kappa(v)$ we call vertex *very heavy*.

From [5] we know that for any weighted bipartite graph H, $\chi_m(H) = \omega(H)$, and it can be optimally multicolored by the following procedure:

Procedure 2.2. *Let $H = (V', V'', E, d)$ be a weighted bipartite graph. We get an optimal multicoloring of H if to each vertex $v \in V'$ we assign a set of colors $\{1, 2, \ldots, d(v)\}$, while with each vertex $v \in V''$ we associate a set of colors $\{m(v)+1, m(v)+2, \ldots, m(v)+d(v)\}$, where $m(v) = \max\{d(u) : \{u, v\} \in E\}$.*

To color vertices of G we use colors from appropriate *palette*. For a given base color c, its palette is defined as a set of pairs $\{(c, i)\}_{i\in\mathbb{N}}$, $c \in \{R, G, B\}$. Such palettes are called *base color palette*. We will also use *extra color palette*, $(c = X)$. In our algorithm we will use colors from each palette for $i \in \{1, \ldots, \lceil\omega(G)/3\rceil\}$. Notice that palettes are equal, so we can tranlate them to numbers in the following way:

- red base color palette: $\{4i : i = 1, \ldots, \lceil\omega(G)/3\rceil\}$

- green base color palette: $\{4i-1 : i = 1, \ldots, \lceil \omega(G)/3 \rceil\}$
- blue base color palette: $\{4i-2 : i = 1, \ldots, \lceil \omega(G)/3 \rceil\}$
- extra color palette: $\{4i-3 : i = 1, \ldots, \lceil \omega(G)/3 \rceil\}$

Since we are not use any other colors, our algorithm assign no more then $\frac{4}{3}\omega(G) + 4$.

In our model of computations we assume that each vertex knows its coordinates as well as its own demand (weight) and demands of all it neighbors in distance k (k-local model). In the next Section we present 1-local algorithm for multicoloring hexagonal graphs without a special heavy double-corner.

3 1-local algorithm for subclass of hexagonals

Our algorithm consists of three main phases. In the first phase vertices take $\kappa(v)$ colors from its base color palette, so use no more than $\omega(G)$ colors. After this phase all light vertices are fully colored while the remaining vertices create a triangle-free hexagonal graph with weighted clique number not exceeding $\lceil \omega(G)/3 \rceil$ (after technical removing very heavy vertices). In the second phase we construct bipartite subgraph of the remaining graph, which is induced by all vertices except some corners. We use Procedure 2.2 and color such graph optimally by using colors from extra color palette. In the third phase we color isolated vertices in remaining graph by using free colors from base color palettes. Since we assumed G is free from a special heavy double-corner, after this phase whole graph is fully properly multicolored.

More precisely, our algorithm consists of the following steps:

Algorithm

Step 0 For each vertex $v = (x, y) \in V$ compute its base color $bc(v)$

$$bc(v) = \begin{cases} R & \text{if} \quad x + 2y \bmod 3 = 0 \\ G & \text{if} \quad x + 2y \bmod 3 = 1 \\ B & \text{if} \quad x + 2y \bmod 3 = 2 \end{cases},$$

and its base function value

$$\kappa(v) = \max\left\{ \left\lceil \frac{d(u) + d(v) + d(t)}{3} \right\rceil : \{v, u, t\} \in \tau(T) \right\}.$$

Step 1 For each vertex $v \in V$ assign to v first $\min\{\kappa(v), d(v)\}$ colors from its base color palette. Construct a new weighted triangle-free hexagonal graph $G_1 = (V_1, E_1, d_1)$ where $d_1(v) = \max\{d(v) - \kappa(v), 0\}$, $V_1 \subseteq V$ is the set of vertices with $d_1(v) > 0$

(heavy vertices) and $E_1 \subseteq E$ is the set of all edges in G with both endpoints from V_1 (E_1 is induced by V_1).

Step 2 For each vertex $v \in V_1$ with $d_1(v) > \kappa(v)$ (very heavy vertices) assign first $\kappa(v)$ free colors from extra color palette and free colors from the base color palettes of its right neighbors in T. Construct a new graph $G_2 = (V_2, E_2, d_2)$ where d_2 is the difference between $d_1(v)$ and the number of assigned colors in this Step, $V_2 \subseteq V_1$ is the set of vertices with $d_2(v) > 0$ and $E_2 \subseteq E_1$ is the set of all edges in G_1 with both endpoints from V_2 (E_2 is induced by V_2).

Step 3 Determine the value of the following function p on vertices of G_2:

- if $v = (x, y)$ is a non-corner:
  - if v has up-left or down-right neighbors in G_2 then $p(v) = x \bmod 2$
  - if v has up-right or down-left neighbors in G_2 then $p(v) = y \bmod 2$
  - if v has left or right neighbors in G_2 then $p(v) = x \bmod 2$
- if $v = (x, y)$ is a corner:
  - if $x \bmod 2 = y \bmod 2$ then $p(v) = y \bmod 2$
  - if $x \bmod 2 \neq y \bmod 2$ then $p(v) = 2$

Step 4 Apply Procedure 2.2 to bipartite graph induced by all vertices from G_2 with $p(v) \in \{0, 1\}$ to satisfy all demands in G_2 by colors from extra color palette. Construct a new graph $G_3 = (V_3, E_3, d_3)$, collection of isolated vertices, where $d_3 = d_2$ is the same as in G_2, $V_3 \subseteq V_2$ is the set of vertices with $p(v) = 2$ and E_3 is an empty set since E_3 is induced by V_3.

Step 5 Color all isolated vertices in G_3 using free colors from the base color palettes of its neighbors in T.

Correctness proof

At the very beginning of the algorithm there is a 1-local communication when each vertex finds out about the demands of all its neighbors. From now on, no more communication will be needed. Recall that each vertex knows its position (x, y) on the triangular lattice T.

In Step 0 there is nothing to prove.

In Step 1 each heavy vertex v assigns $\kappa(v)$ colors from its base color palette, while each light vertex u assigns $d(u)$ colors from its base color palette. Note that G_1 consists only of heavy vertices, therefore G_1 is a triangle-free hexagonal graph. For any $\{v, u, t\} \in \tau(G)$, since $3 \min\{\kappa(v), \kappa(u), \kappa(t)\} \geq d(v)+d(u)+d(t)$ and $\min\{\kappa(v), \kappa(u), \kappa(t)\} \geq \min\{d(v), d(u), d(t)\}$, at most two of $d_1(v), d_1(u), d_1(t)$ are strictly positive and at least one of the vertices u, v

and t has all its required colors totally assigned in Step 1. Therefore, the graph G_1 does not contain 3-clique, i.e. it is a triangle-free hexagonal graph. The remaining weight of each vertex $v \in G_1$ is

$$d_1(v) = d(v) - \kappa(v).$$

In Step 2 only vertices with $d_1(v) > \kappa(v)$ (very heavy vertices) are colored. If vertex v is very heavy in G then it is isolated in G_1 (all its neighbors are light in G). Otherwise, for some $\{v, u, t\} \in \tau(T)$ we would have

$$d(v) + d(u) > 2\kappa(v) + \kappa(u) \geq 3a(v, u, t) \geq d(v) + d(u),$$

a contradiction. Without loss of generality we may assume that $bc(v) = R$ and its right neighbor is blue. Denote by

$$D_B(v) = \min\{\kappa(v) - d(u) : \{u, v\} \in T, bc(u) = B\},$$

the number of free colors from blue base color palette. Obviously, $D_B(v) > 0$ for very heavy vertices $v \in G_1$. Since in Step 1 each light vertex t uses exactly $d(t)$ colors from its base color palette, we have at least $D_B(v)$ free colors from the blue base color palette, so that vertex v can assign those colors to itself. After that it can take all $\kappa(v)$ colors from extra color palette. Then, we would have G_2 with $\omega(G_2) \leq \lceil \omega(G)/3 \rceil$. To prove it, we will need the following lemma:

Lemma 3.1. *In G_1 for every edge $\{v, u\} \in E_1$ holds:*

$$d_1(v) + d_1(u) \leq \kappa(v), \quad d_1(u) + d_1(v) \leq \kappa(u).$$

Proof. Assume that v and u are adjacent heavy vertices in G and $d_1(v) + d_1(u) > \kappa(v)$. Then for some $\{v, u, t\} \in \tau(T)$ we have:

$$d(v) + d(u) = d_1(v) + \kappa(v) + d_1(u) + \kappa(u) > 2\kappa(v) + \kappa(u) \geq 3a(u, v, t) \geq d(u) + d(v),$$

a contradiction. □

Fact 3.2.

$$\omega(G_2) \leq \lceil \omega(G)/3 \rceil.$$

Proof. Recall that in hexagonal graph the only cliques are triangles, edges and isolated vertices. Since G_1 is a triangle-free hexagonal graph, G_2 also does not contain any triangle, so we have only edges and isolated vertices to check.

For each edge $\{v, u\} \in E_2$ from Lemma 3.1 and Fact 2.1 we have:

$$d_2(v) + d_2(u) \leq d_1(v) + d_1(u) \leq \kappa(v) \leq \lceil \omega(G)/3 \rceil.$$

For each isolated vertex $v \in G_2$ we also should have $d_2(v) \leq \lceil \omega(G)/3 \rceil$. Indeed, if $d_2(v) \leq \kappa(v)$, then it holds by Fact 2.1. If $d_2(v) > \kappa(v)$, then $d_1(v) > \kappa(v)$, so in Step 2 vertex v has to assign $\kappa(v)$ colors from extra color palette and take colors from its right neighbor's base color palette. Then, for $bc(v) = R$, and its right neighbors with base color equal to blue:

$$d_2(v) = d_1(v) - \kappa(v) - D_B(v) \le d(v) - \kappa(v) - \kappa(v) - \kappa(v) + d(u) \le d(v) + d(u) - 3\kappa(v) \le 0$$

for some $\{v, u\} \in T$, $bc(u) = B$. Hence, $d_2(v) \le \lceil \omega(G)/3 \rceil$, and so $\omega(G_2) \le \lceil \omega(G)/3 \rceil$. □

Notice that in proof of Lemma 3.2 we showed that in G_2 we don't have very heavy vertices at all.

In Step 3 each vertex v has to decide whether it is a corner or not, and where are located heavy neighbors of v. In 1-local model v does not know which of his neighbors are heavy (and still exist in G_2) and which are light. Vertex v knows only where its neighbors with $d(u) \le \max\{a(v, u, t) : \{v, u, t\} \in \tau(T)\}$ are located. We call those vertices *slight neighbors* of v. Slight neighbors of v must be light and, so, they are fully colored in Step 1. Thus, v knows where it cannot have neighbors in G_2 and presumes that all its neighbors which are not slight, still exist in G_2. Based on that knowledge, it can decide whether it is a corner or not. In each triangle in $\tau(T)$ containing v at least one neighbor of v is slight, so v has at least three such neighbors. If vertex v has more than four slight neighbors, then it is a non-corner. If vertex v has four slight neighbors, then the remaining two are not slight. In this case if an angle between those two is π, then v is non-corner, otherwise it is a corner. If vertex v has exactly three slight neighbors, then it is a corner.

In Step 4 we would like to apply Procedure 2.2 to graph induced on G_2 by all vertices with $p(v) \in \{0, 1\}$. It is easy to see that such graph is bipartite. For non-corners we use coordinates to find out if vertex is "odd" or "even" on some "line", while for corners we take only those with the same parity in all directions. The only problem is that, under 1-locality assumption, vertices cannot calculate value of d_2 of the neighbors, which is needed in Procedure 2.2 to calculate value $m(v) = \max\{\lceil d_2(u) \rceil : \{u, v\} \in E_2\}$. However, we can replace $d_2(u)$ by $d_2^v(u)$, which is the number of expected remaining demands on vertex u in vertex v after Step 2, and take $m'(v) = \max\{\lceil d_2^v(u) \rceil : \{u, v\} \in E_2\}$. More precisely,

$$d_2^v(u) = d(u) - \max\{a(u, v, t) : \{u, v, t\} \in \tau(T)\}$$

Note that $d_2^v(u) \ge d_2(u)$ for any $\{u, v\} \in E_2$. However, for every $\{v, u\} \in E_2$ we have

$$d_2(v) + d_2^v(u) \le \kappa(v).$$

Assume that this inequality does not hold. Denote by

$$b(u, v) = \max\{a(u, v, t) : \{u, v, t\} \in \tau(T)\}.$$

Then, for some $\{t, v, u\} \in \tau(T)$, we have:

$$d(v) + d(u) = d_2(v) + \kappa(v) + d_2^v(u) + b(u, v) > 2\kappa(v) + b(u, v) \ge 3a(u, v, t) \ge d(u) + d(v),$$

a contradiction. Hence, if we use d_2^v instead of d_2 in each vertex from the second set of our bipartition, Procedure 2.2 works and uses at most $\lceil \omega(G)/3 \rceil$ colors, which is not greater than size of extra color palette. We don't create conflicts with colors from extra color palette assigned in previous Steps. Only isolated vertices in G_1 in Step 2 put some colors from extra color palette, but they cannot be adjacent to any vertices in G_2 since they are isolated.

After coloring vertices with $p(v) \in \{0, 1\}$, only corners from G_2 with $p(v) = 2$ remain and they induce a new graph G_3. If a corner in G_2 is surrounded by non-corners then it is isolated vertex in G_3. If a corner is adjacent to some other corner in G_2 then they must form one of the three situation given in Figure 3.

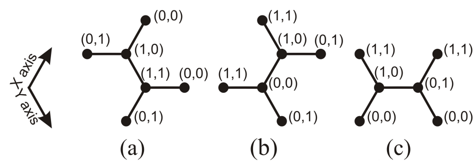

Figure 4: All possibilities for adjacent corners in G_2 (with coordinates mod 2)

Notice that in situation (a) and (b) from Figure 4, corner v with $p(v) = 2$ has always its corner neighbor u with $p(u) \in \{0, 1\}$. Indeed, while $v = (x, y)$, $x \bmod 2 \neq y \bmod 2$ and in u we change one coordinate by 1, then $p(u) \neq 2$. In situation (c) we change both coordinates by 1, so in this case $p(u) = 2$. Hence, after coloring bipartite graph only the configuration from Figure 4 (c) can survive in G_3, but this is a special heavy double-corner which is forbidden in graph G. Hence G_3 is a collection of isolated vertices only.

In Step 5 we consider an isolated vertex $v \in G_3$ and return to base color palettes. Vertex v has three slight neighbors with the same base color. Without loss of generality, assume that $bc(v) = R$ and its slight neighbors' base color is blue. From definition of function D_B we have $D_B(v)$ free colors from blue base color palette. We have to show that $d_3(v) \leq D_B(v)$. Let $\Delta = \{u, v, t\} \in \tau(T)$ be a triangle such that t is a green vertex which is heavy neighbor of v, and u is a blue vertex which is a light neighbor of v. Denote by $s_\Delta(t) = d(t) - a(u, v, t)$. Then we have

$$0 \geq d(v) - a(u,v,t) + d(t) - a(u,v,t) + d(u) - a(u,v,t) \geq d(v) - \kappa(v) + s_\Delta(t) + d(u) - \kappa(v) \geq$$

$$\geq d_1(v) + s_\Delta(t) - D_B(v) \geq d_2(v) + s_\Delta(t) - D_B(v) = d_3(v) + s_\Delta(t) - D_B(v)$$

Since t is a heavy neighbor of v, therefore $d_3(v) < D_B(v)$. Hence, vertex v has as much as $d_3(v)$ free colors from the blue base color palette at his disposal.

After Step 5, all demands are satisfied in proper way and we use only colors from our palettes. Hence we arrived at the thesis of Theorem 1.1.

4 Conclusion and Open Problem

We have given a 1-local 4/3-approximation algorithm for multicoloring hexagonal graphs without a special heavy double-corner, subclass of hexagonals with condition introduced in [7].

Notice that in 1-local model of computation for general case hexagonal graphs we are unable to predict if graph contains a special heavy double-corner. If vertex want to check while it is a part of a special heavy double-corner or not, it has to obtain which neighbors of its non slight neighbors are slight, so it is forced to use 2-local communication. Therefore in general case vertices in our 1-local algorithm cannot decide itself if they need more communication to color properly. The open problem is to find such 1-local 4/3-competitive algorithm for some subclass of hexagonal graphs which would know in 1-local model if it can or cannot obtain proper multicoloring for whole graph.

References

[1] Hale, W.K.: Frequency assignment: theory and applications, Proceedings of the IEEE, vol 68(12), pp 1497-1514 (1980)

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

[3] McDiarmid, C., Reed, B.: Channel assignment and weighted coloring, Networks, vol. 36(2), pp. 114-117 (2000)

[4] Narayanan, L.: Channel assignment and graph multicoloring, Handbook of wireless networks and mobile computing, pp 71-94, Wiley, New York, (2002)

[5] Narayanan, L., Shende, S.M.: Static frequency assignment in cellular networks, Algorithmica, vol. 29(3), pp 396-409 (2001)

[6] Šparl, P., Žerovnik, J.: 2-local 4/3-competitive Algorithm for Multicoloring Hexagonal Graphs, Journal of Algorithms, vol. 55(1), pp 29-41 (2005)

[7] Šparl, P., Žerovnik, J.: 2-local 7/6-competitive algorithm for multicoloring a sub-class of hexagonal graphs, International Journal of Computer Mathematics, to appear

[8] Witkowski, R. *A 1-local 17/12-competitive Algorithm for Multicoloring Hexagonal Graphs*, Lecture Notes of Computer Science, vol. 5699/2009, pp 346-356 (2009)

[9] Witkowski, R., Žerovnik, J. *1-local 7/5-competitive Algorithm for Multicoloring Hexagonal Graphs*, International Symposium on Combinatorial Optimization, 2010, accepted.

A 2-local version of algorithm for general case hexagonals

In this Appendix we will prove in different way than in [6] that:

Theorem A.1. *There is a 2-local distributed approximation algorithm for multicoloring hexagonal graphs which uses at most* $\lceil \frac{4}{3}\omega(G) \rceil + O(1)$ *colors. Time complexity of the algorithm at each vertex is constant.*

Since in [2] it was proved that a k-local c-approximate algorithm can be easily converted to a k-local c-competitive algorithm, we have,

Corollary A.2. *There is a 2-local 4/3-competitive algorithm for multicoloring general case hexagonal graphs.*

Assume that we deal with general case hexagonal graphs. Then adding two additional steps to the algorithm from Section 3, we get a 2-local 4/3-competitive algorithm for multicoloring general case hexagonal graphs. Those two steps are given below:

Step 6 For each isolated edge $\{v, u\} \in G_3$ (v is left neighbor of u), to vertices v and u assign colors from its slight neighbors base color palette in the following way:

- to v assign colors from interval $\{\kappa(v) - d_3(v) + 1, \ldots, \kappa(v)\}$,
- to u assign colors from interval $\{a(u, v, r) - d_3(v) - d_3(u) + 1, \ldots, a(u, v, r) - d_3(v)\}$

where r is the common slight neighbor of v and u with larger value of demand in G.

Step 7 For u, v, r as in Step 6, let s be the right neighbor of u. Denote by t right neighbor of s. If $d(s) > d(r)$ then:

- **Case (7.1)** if t is light or very heavy vertex in G or non corner in G_2 then recolor s using colors $\{1, \ldots, d(r)\} \cup \{\kappa(s) - (d(s) - d(r)) + 1, \ldots, \kappa(s)\}$ from base color palette of s.
- **Case (7.2)** if t is a corner in G_2 and s has exactly two additional neighbors $w, w' \in G_2$ at angle π ($p(w) = p(w')$) then choose the value of function $p(s) = (1 + p(w)) \bmod 2$ and color s in the way as vertices during Procedure 2.2 in Step 4.
- **Case (7.3)** if t is a corner in G_2 and s has exactly two additional neighbors $w, w \in G_2$ at angle $\pi/3$ ($p(w) \neq p(w')$) then recolor s using colors from extra color palette.
- **Case (7.4)** if t is a corner in G_2 and s has more than two additional neighbors in G_2 then recolor s using last $d(s)$ free colors from extra color palette. If necessary, recolor neighbors of s having value of p equal to 1.

If edge $\{v,u\} \in G_3$ is isolated, then $x_v \bmod 2 \neq y_v \bmod 2, x_u \bmod 2 \neq y_u \bmod 2$ and it has to be situated as on Figure 4 (c). Denote by o,q,r,s slight neighbors of vertices u and v. Notice that they all have the same base color. Recall that each subset of colors from base color palette can be identify with a subset of natural numbers. We can assume that light vertices o,q,r,s have colors from its base color palette, for i in $\{1,\ldots,d(o)\}$, $\{1,\ldots,d(q)\}$, $\{1,\ldots,d(r)\}$, $\{1,\ldots,d(s)\}$, respectively.

To prove that Step 6 approach proper coloring we have to check if pairs of set of adjacent vertices are disjoint. Vertices u and v with r and q do not create conflicts since

$$a(u,v,r) - d_3(u) - d_3(v) + 1 = a(u,v,r) - d(u) + \kappa(u) - d(v) + \kappa(v) + 1 \geq$$

$$\geq 3a(u,v,r) - d(u) - d(v) + 1 \geq d(r) + 1 > d(r) \geq d(q)$$

Vertices v with u do not create conflicts since $\kappa(v) - d_2(v) + 1 > a(u,v,r) - d_2(v)$. Denote by s right neighbor of u. For vertices u and s we need $a(u,v,r) - d_3(v) - d_3(u) + 1 > d(s)$. Assume that this inequality does not hold. Then we have:

$$d(s) > a(u,v,r) - d_3(u) - d_3(v) = a(u,v,r) - d(u) + \kappa(u) - d(v) + \kappa(v) \geq$$

$$\geq 3a(u,v,r) - d(u) - d(v) = d(r)$$

Hence we don't have conflict only if $d(s) \leq d(r)$. Otherwise we have a conflict and must recolor s in Step 7.

In Case 7.1 we assign to s last $d(s) - d(r)$ colors from its base color palette. Vertices s and v do not create conflicts since $\kappa(s) - d(s) + d(r) \geq a(u,v,r) - d_3(v)$ and $a(u,v,r) - d_3(v) - d_3(u) + 1 > d(r)$. Assume that first inequality does not hold and recall that $d_2(u) + d(s) \leq \kappa(s)$ and $d_2(u) \leq a(u,v,r) - d_2(v)$. Then,

$$0 > \kappa(s) - d(s) + d(r) - a(u,v,r) + d_2(v) \geq \kappa(s) - d(s) - d_2(u) \geq 0,$$

a contradiction. Now assume that second inequality does not hold, then

$$d(r) > a(u,v,r) - d_3(u) - d_3(v) \geq a(u,v,r) - d(u) + \kappa(u) - d(v) + \kappa(v) \geq$$

$$\geq 3a(u,v,r) - d(u) - d(v) \geq d(r),$$

again a contradiction. Hence s and v are not conflicted. If vertex t is light or non-corner then we cannot have any cascading conflicts. Also if t is very heavy we cannot have conflict since in Step 2 vertex t took colors from its right neighbor base color palette, not from left neighbor.

In Case 7.2 vertex s has four neighbors in G_2: left, right – corners in G_2, and also two more (w and w') at angle π. Notice that if $w = (x,y)$ then $w' = (x \pm 2, y \pm 2)$, so $p(w) = p(w')$. In this case in s we can put $p(s) = (1 + p(w)) \bmod 2$ and assign to s colors in the same way as we did it during Step 4. We do not create conflicts since in neighborhood of s

only w and w' had assigned extra colors in previous steps, but $d(s) + d_2(w) \leq \kappa(s)$ and $d(s) + d_2(w') \leq \kappa(s)$.

In Case 7.3 vertex s has four neighbors in G_2: left, right – corners in G_2, and also two more (w and w') at angle $\pi/3$. Notice that if $w = (x, y)$ then $w' = (x \pm 1, y \pm 1)$ so $p(w) \neq p(w')$. Since w and w' in Step 4 took $d_2(w) + d_2(w')$ free colors from extra color palette, we have

$$\kappa(s) - d_2(w) - d_2(w') = \kappa(s) - d(w) + \kappa(w) - d(w') + \kappa(w') \geq 3a(s, w, w') - d(w) - d(w') \geq d(s)$$

free colors in extra color palette, and s can use them to color itself.

Notice that if w and w' are the only neighbors of s except u and t, then it cannot be at angle $2\pi/3$. If so, they are both neighbors of s (or t), but then t (respectively s) has at most one neighbor in G_2 and is not a corner – a contradiction.

In Case 7.4 vertex s has more than four neighbors in G_2. Denote by b and c adjacent neighbors of s, different then t and u. Assume without loss of generality that $p(c) = 0$ and $p(b) = 1$ (c is in the first and b is in the second set of bipartition during Procedure 2.2 in Step 4). Let e be the third neighbor of b with the same base color as c, and f, f' be its slight neighbors with base color as in s (see Figure 5).

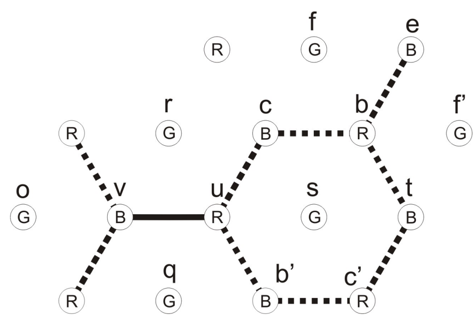

Figure 5: Configuration in Case 7.5

If e is light vertex in G or is a corner in G_2 or $d_2^b(c) \geq d_2^b(e)$ then the largest color from extra color palette assigned to b or c is

$$l(b, c) = d_2^b(c) + d_2(b) = d(c) - \max\{a(c, b, s), a(c, b, f)\} + d(b) - \kappa(b).$$

Hence s can assign colors $\{l(b, c) + 1, \ldots, l(b, c) + d(s)\}$ from extra color palette and no conflicts are being created. We have to check, if s do not exceeded a capacity of extra palette. But

$$\omega(G)/3 - l(b, c) = \omega(G)/3 - d(c) + \max\{a(c, b, s), a(c, b, f)\} - d(b) + \kappa(b) \geq$$

$$\geq 3 \max\{a(c, b, s), a(c, b, f)\} - d(c) - d(b) \geq \max\{d(s), d(f)\} \geq d(s)$$

Hence the largest color assigned to s is not greater then $\omega(G)/3$. If e is a non corner in G_2 and $d_2^b(c) < d_2^b(e)$ then:

$$l(b,c) = d_2^b(e) + d_2(b) = d(e) - \max\{a(e,b,f), a(e,b,f')\} + d(b) - \kappa(b).$$

and:

$$\omega(G)/3 - l(b,c) = \omega(G)/3 - d(e) + \max\{a(e,b,f), a(e,b,f')\} - d(b) + \kappa(b) \geq$$

$$\geq 3\max\{a(e,b,f), a(e,b,f')\} - d(e) - d(b) \geq \max\{d(f), d(f')\}$$

If $\max\{d(f), d(f')\} \geq d(s)$ then we can assign to s colors $\{l(b,c)+1, \ldots, l(b,c)+d(s)\}$ from extra color palette and no conflict occurs and s also do not fall off its base color palette. But if $\max\{d(f), d(f')\} < d(s)$ we have to make an Extra Step to recolor vertex b and avoid conflicts.

Extra Step Let b be a corner in G_2, $p(b) = 1$.

Case (i) If its down-right (or up-right) neighbor (t) is a corner in G_2, its up-right (down-right) neighbor (e) is not a corner in G_2, denote by c left neighbor of b, by r left neighbor of c, by s down-left (up-left) neighbor of b, by u left neighbor of s and by f, f' remaining slight neighbors of b. If u is a heavy vertex, $d(r) < d(s)$, $d_2^b(c) < d_2^b(e)$ and $\max\{d(f), d(f')\} < d(s)$ then recolor last $d(s) - \max\{d(f), d(f')\}$ colors from extra color palette to $\{\max\{d(f), d(f')\} + 1, \ldots, d(s)\}$ colors from base color palette of s.

Case (ii) If its up-left (or down-left) neighbor (u) is a corner in G_2, its down-left (up-left) neighbor (e) is not a corner in G_2, denote by c right neighbor of b, by v left neighbor of u, by r up-left neighbor of u, by q down-left neighbor of u, by s up-right (down-right) neighbor of b, by t right neighbor of s and by f remaining slight neighbor of b. If t is a heavy vertex, $\max\{d(r), d(q), d(f)\} < d(s)$ and $d_2^b(c) < d_2^b(e)$ then recolor last $d(s) - \max\{d(r), d(q)\}$ colors from extra color palette to $\{\max\{a(u,v,r), a(u,v,q)\} - d_3(v) + 1, \ldots, \max\{a(u,v,r), a(u,v,q)\} - d_3(v) + d(s) - \max\{d(r), d(q)\}\}$ colors from base color palette of s.

Notice that conditions in Extra Step correspond to the situation concerned by Case 7.4 (see Figure 5). As the result of this Step we avoid conflicts between s and its neighbors by freeing largest colors in extra color palette. Finally we have to check if there are no conflicts with b. Indeed, if b is the neighbor of t then it takes $\{\max\{d(f), d(f')\}+1, \ldots, d(s)\}$ colors from base color palette of s. No conflict occurs since e is not a corner, $\{\max\{d(f), d(f')\} + 1 > \max\{d(f), d(f')\}$ and the smallest color used in t from base color palette of s is not smaller than $d(s) + 1$. On the other hand, if b is the neighbor of u then it takes colors $\{\max\{a(u,v,r), a(u,v,q)\} - d_3(v) + 1, \ldots, \max\{a(u,v,r), a(u,v,q)\} - d_3(v) + d(s) - \max\{d(r), d(q)\}\}$ from base color palette of s. There is also no conflicts here since e is not a corner and $\max\{a(u,v,r), a(u,v,q)\} - d_3(v) + 1 > \max\{a(u,v,r), a(u,v,q)\} - d_3(v) - d_3(u) + 1 > d(s) > \max\{d(q), d(r)\}$, as proved before. We also do not exceed our palette since $\max\{a(u,v,r), a(u,v,q)\} - d_3(v) + d(s) - \max\{d(q), d(r)\} \leq \omega(G)/3$ (it holds because $\kappa(s) - d(s) + \max\{d(q), d(r)\} \geq \max\{a(u,v,r), a(u,v,q)\} - d_2(v)$, proved before).

Hence we arrived at the thesis of Theorem A.1.

Therefore, we presented additional steps in algorithm from Section 3 which allow multicolor all hexagonal graph in 2-local model. Hence we have shown different than previously known 2-local 4/3-approximation algorithm for multicoloring hexagonal graphs – the algorithm with the best known ratio in this model of computation.