Determining $L(2,1)$-Span in Polynomial Space

Konstanty Junosza-Szaniawski, Paweł Rzążewski
{k.szaniawski, p.razzewski}@mini.pw.edu.pl

Warsaw University of Technology
Faculty of Mathematics and Information Science
Pl. Politechniki 1, 00-661 Warsaw, Poland

Abstract

A $k$-$L(2,1)$-labeling of a graph is a function from its vertex set into the set $\{0, \ldots, k\}$, such that the labels assigned to adjacent vertices differ by at least 2, and labels assigned to vertices of distance 2 are different. It is known that finding the smallest $k$ admitting the existence of a $k$-$L(2,1)$-labeling of any given graph is NP-Complete.

In this paper we present an algorithm for this problem, which works in time $O((9+\epsilon)^n)$ and polynomial memory, where $\epsilon$ is an arbitrarily small positive constant. This is the first exact algorithm for $L(2,1)$-labeling problem with time complexity $O(c^n)$ for some constant $c$ and polynomial space complexity.

1 Introduction

A frequency assignment problem is the problem of assigning channels of frequency (represented by nonnegative integers) to each radio transmitter, so that no transmitters interfere with each other. Hale [12] formulated this problem in terms of so-called $T$-coloring of graphs.

According to [11], Roberts was the first who proposed a modification of this problem, which is called an $L(2,1)$-labeling problem. It asks for such a labeling with nonnegative integer labels, that no vertices in distance 2 in a graph have the same label and labels of adjacent vertices differ by at least 2.

A $k$-$L(2,1)$-labeling problem is to determine if there exists an $L(2,1)$-labeling of a given graph with no label greater than $k$. By $\lambda(G)$ we denote an $L(2,1)$-span of $G$, which is the smallest value of $k$ that guarantees the existence of a $k$-$L(2,1)$-labeling of $G$.

The problem of $L(2,1)$-labeling has been extensively studied (see [3, 7, 10, 20] for some surveys on the problem and its generalizations). A considerable attention has been given to bounding the value of $\lambda(G)$ by some function of $G$.

Griggs and Yeh [11] proved that $\lambda(G) \leq \Delta^2 + \Delta + 1$ and conjectured, that

$^{1}\Delta$ denotes the largest vertex degree in a graph.

1 Introduction

A frequency assignment problem is the problem of assigning channels of frequency (represented by nonnegative integers) to each radio transmitter, so that no transmitters interfere with each other. Hale [12] formulated this problem in terms of so-called $T$-coloring of graphs.

According to [11], Roberts was the first who proposed a modification of this problem, which is called an $L(2,1)$-labeling problem. It asks for such a labeling with nonnegative integer labels, that no vertices in distance 2 in a graph have the same label and labels of adjacent vertices differ by at least 2.

A $k$-$L(2,1)$-labeling problem is to determine if there exists an $L(2,1)$-labeling of a given graph with no label greater than $k$. By $\lambda(G)$ we denote an $L(2,1)$-span of $G$, which is the smallest value of $k$ that guarantees the existence of a $k$-$L(2,1)$-labeling of $G$.

The problem of $L(2,1)$-labeling has been extensively studied (see [3, 7, 10, 20] for some surveys on the problem and its generalizations). A considerable attention has been given to bounding the value of $\lambda(G)$ by some function of $G$.

Griggs and Yeh [11] proved that $\lambda(G) \leq \Delta^2 + \Delta + 1$ and conjectured, that

$^{1}\Delta$ denotes the largest vertex degree in a graph.
\( \lambda(G) \leq \Delta^2 \) for every graph \( G \). There are several results supporting this conjecture, for example Gonçalves \[9\] proved that \( \lambda(G) \leq \Delta^2 + \Delta - 2 \) for graphs with \( \Delta \geq 3 \). Havet et al. \[13\] have settled the conjecture in affirmative for graphs with \( \Delta \geq 10^6 \). For graphs with smaller \( \Delta \), the conjecture still remains open.

It is interesting to note that the Petersen and Hoffmann-Singleton graphs are the only two known graphs with maximum degree greater than 2, for which this bound is tight.

The second main branch of research in \( L(2,1) \)-labeling was pointed to analyzing the problem from the complexity point of view. For \( k \geq 4 \), the \( k-L(2,1) \)-labeling problem was proven to be NP-complete by Fiala et al. \[6\] (for \( k \leq 3 \) the problem is polynomial). It remains NP-complete even for regular graphs (see Fiala and Kratochvíl \[8\]), planar graphs (see Eggeman et al. \[4\]) or series-parallel graphs (see Fiala et al. \[5\]).

An exact algorithm for the so called Channel Assignment Problem, presented by Král’ \[19\], implies an \( O^*(4^n) \) \(^2\) algorithm for the \( L(2,1) \)-labeling problem. Havet et al. \[14\] presented an algorithm for computing \( L(2,1)(G) \), which works in time \( O^*(15^\omega) = O^*(3.8730^n) \). This algorithm has been improved \[17\] \[18\], achieving a complexity bound \( O^*(3.2361^n) \). Recently, a new algorithm for \( L(2,1) \)-labeling with a complexity bound \( O^*(2.6488^n) \) has been presented \[16\].

All algorithms mentioned above are based on dynamic programming approach and use exponential memory. Havet et al. \[14\] presented a branching algorithm for \( k-L(2,1) \)-labeling problem with a time complexity \( O^*((k-2.5)^n) \) and polynomial space complexity. Until now, no algorithm for \( L(2,1) \)-labeling with time complexity \( O(c^n) \) for some constant \( c \) and polynomial space complexity has been presented. However, there are such algorithms for a related problem of classical graph coloring. The first one, with time complexity \( O(5.283^n) \), was shown by Bodleander and Kratsch \[2\]. The best currently known algorithm for graph coloring with polynomial space complexity is by Björklund et al. \[1\], using the inclusion-exclusion principle. Its time complexity is \( O(2.246^n) \).

In this paper we present the first exact algorithm for the \( L(2,1) \)-labeling problem with polynomially bounded space complexity. The algorithm works in time \( O((9+\epsilon)n) \) (where \( \epsilon \) is an arbitrarily small positive constant) and is based on a divide and conquer approach.

## 2 Preliminaries

Throughout the paper we consider finite undirected graphs without multiple edges or loops. The vertex set (edge set) of a graph \( G \) is denoted by \( V(G) \) (\( E(G) \), respectively).

Let \( \text{dist}_G(x, y) \) be the distance between vertices \( x \) and \( y \) in a graph \( G \), which is the length of a shortest path joining \( x \) and \( y \).

A set \( X \subseteq V(G) \) is a 2-packing in \( G \) if and only if all its vertices are in distance at least 3 from each other (\( \forall x, y \in X \text{ dist}_G(x, y) > 2 \)).

\(^2\)In the \( O^* \) notation we omit polynomially bounded terms.
Let $N(v) = \{u \in V(G): (u, v) \in E(G)\}$ denote the set of neighbors (the neighborhood) of a vertex $v$. The set $N[v] = N(v) \cup \{v\}$ denotes the closed neighborhood of $v$. The neighborhood of a set $X$ of vertices in $G$ is denoted by $N(X) = \bigcup_{v \in X} N(v)$ and its closed neighborhood is denoted by $N[X] = N(X) \cup X$.

For a subset $X \subseteq V(G)$, we denote the subgraph of $G$ induced by the vertices in $X$ by $G[X]$. A square of a graph $G = (V, E)$ is the graph $G^2 = (V, \{uv \in V^2: \text{dist}_G(u, v) \leq 2\})$.

**Definition 1.** For a graph $G$ and sets $Y, Z, M \subseteq V(G)$, a $(k - 1)$-$L^M_Z(Y)$-labeling of a graph $G$ is a function $c: Y \to \{0, 1, \ldots, k - 1\}$, such that $c^{-1}(0) \cap Z = c^{-1}(k - 1) \cap M = \emptyset$, and for every $u, v \in Y$:

$$|c(v) - c(u)| \geq 2 \text{ if } \text{dist}_G(u, v) = 1$$

$$|c(v) - c(u)| \geq 1 \text{ if } \text{dist}_G(u, v) = 2.$$ 

A function $c: Y \to \mathbb{N}$ is an $L^M_Z(Y)$-labeling of $G$ if there exists $k \in \mathbb{N}$ such that $c$ is a $(k - 1)$-$L^M_Z(Y)$-labeling of $G$.

**Definition 2.** For $Y, Z, M \subseteq V(G)$ let $\Lambda^M_Z(Y, G)$ denote the smallest value of $k$ admitting the existence of $(k - 1)$-$L^M_Z(Y)$-labeling of $G$. We define $\Lambda^M_Z(\emptyset, G) \overset{\text{def.}}{=} 0$ for all graphs $G$ and sets $Z, M \subseteq V(G)$.

Any $(k - 1)$-$L^M_Z(Y)$-labeling of $G$ with $k = \Lambda^M_Z(Y, G)$ is called optimal.

We observe that even if $c$ is an optimal $L^M_Z(Y)$-labeling of $G$, then any of the sets $c^{-1}(0)$ and $c^{-1}(\Lambda^M_Z(Y, G) - 1)$ may be empty. In the extremal case, if $Z = M = Y$, then $c^{-1}(0) = c^{-1}(k - 1) = \emptyset$ for all $k$ and feasible $(k - 1)$-$L^M_Z(Y)$-labelings $c$ of $G$.

Notice that $\Lambda^0_Z(V(G), G) = \lambda(G) + 1$ for every graph $G$.

**Definition 3.** For a graph $G$, a $G$-correct partition of a set $Y \subseteq V(G)$ is a triple $(A, X, B)$, such that:

1. The sets $A, X, B \subseteq Y$ form a partition of $Y$
2. $X$ is a nonempty 2-packing in $G$
3. $|A| \leq \frac{|Y|}{2}$ and $|B| \leq \frac{|Y|}{2}$

### 3 Algorithm

In this section we present a recursive algorithm for computing $\Lambda^M_Z(Y, G)$ for any graph $G$ and sets $Y, Z, M \subseteq V(G)$. It is then used to find an $L(2, 1)$-span a graph $G$.

The algorithm is based on the divide and conquer approach. First, the algorithm exhaustively check if $\Lambda^M_Z(Y, G) \leq 3$. If not, the set $Y$ is partitioned into three sets $A, X, B$, which form a $G$-correct partition of $Y$. The sets $A$ and $B$ are then labeled recursively.
The labeling of the whole \( Y \) is constructed from the labelings found in the recursive calls. The sets of labels used on the sets \( A \) and \( B \) are separated from each other by the label used for the 2-packing \( X \). This allows to solve the subproblems for \( A \) and \( B \) independently from each other.

Iterating over all \( G \)-correct partitions of \( Y \), the algorithm computes the minimum \( k \) admitting the existence of a \((k-1)\)-\( L^M_Z(Y) \)-labeling of \( G \), which is by definition \( \Lambda^M_Z(Y,G) \).

**Algorithm 1: Find-Lambda**

1. **Input**: Graph \( G \), Sets \( Y, Z, M \subseteq V(G) \)
2. if \( Y = \emptyset \) then return 0
3. foreach \( c : Y \rightarrow \{0, 1, 2\} \) do
4.     for \( k \leftarrow 1 \) to 3 do
5.         if \( c \) is a \((k-1)\)-\( L^M_Z(Y) \)-labeling of \( G \) then return \( k \)
6.     \( k \leftarrow \infty \)
7. foreach \( G \)-correct partition \((A, X, B)\) of \( Y \) do
8.     if \( A \neq \emptyset \) and \( B \neq \emptyset \) then \( k_X \leftarrow 1 \)
9.     if \( A = \emptyset \) and \( X \cap Z = \emptyset \) then \( k_X \leftarrow 1 \)
10.    if \( A = \emptyset \) and \( X \cap Z \neq \emptyset \) then \( k_X \leftarrow 2 \)
11.    if \( B = \emptyset \) and \( X \cap M = \emptyset \) then \( k_X \leftarrow 1 \)
12.    if \( B = \emptyset \) and \( X \cap M \neq \emptyset \) then \( k_X \leftarrow 2 \)
13.    \( k_A \leftarrow \text{Find-Lambda}(G, A, Z, N(X)) \)
14.    \( k_B \leftarrow \text{Find-Lambda}(G, B, N(X), M) \)
15.    \( k \leftarrow \min(k, k_A + k_X + k_B) \)
16. return \( k \)

**Lemma 1.** For a graph \( G \) and sets \( Y, Z, M \subseteq V(G) \), if \( Y \) is a 2-packing in \( G \), then \( \Lambda^M_Z(Y,G) \leq 3 \).

**Proof.** The labeling \( c : Y \rightarrow \{0, 1, 2\} \) such that \( c(v) = 1 \) for every \( v \in Y \) is a 2-\( L^M_Z(Y) \)-labeling of \( G \).

**Theorem 1.** For any graph \( G \) and sets \( Y, Z, M \subseteq V(G) \), the algorithm call \( \text{Find-Lambda}(G, Y, Z, M) \) returns \( \Lambda^M_Z(Y,G) \).

**Proof.** If \( Y = \emptyset \), the correct result is given in the line \( \mathbb{I} \) (by the definition of \( \Lambda^M_Z(\emptyset, G) \)). If \( \Lambda^M_Z(Y,G) \leq 3 \), the result is found by the exhaustive search in the line \( \mathbb{II} \). Notice that if \( |Y| \leq 1 \), then by Lemma \( \mathbb{I} \) \( \Lambda^M_Z(Y,G) \leq 3 \).

Assume that the statement is true for all graphs \( G' \) and all sets \( Y', Z', M' \subseteq V(G') \), such that \( |Y'| < n \), where \( n \geq 1 \).

Let \( G \) be a graph and \( Y, Z, M \) be subsets of \( V(G) \) such that \( |Y| = n \). We may assume that \( \Lambda^M_Z(Y,G) > 3 \). Let \( k \) be the value returned by the algorithm call \( \text{Find-Lambda}(G, Y, Z, M) \).

First we prove that \( k \geq \Lambda^M_Z(Y,G) \), i.e. there exists a \((k-1)\)-\( L^M_Z(Y) \)-labeling of \( G \). Let us consider the \( G \)-correct partition \((A, X, B)\) of \( Y \), for which the value of \( k \) was set in the line \( \mathbb{III} \). Since each of the sets \( A \) and \( B \) has less than \( n \)
vertices, by the inductive assumption there exists a $(k_A - 1) - L_{Z}^{N(X)}(A)$-labeling $c_A$ of $G$ and a $(k_B - 1) - L_{N(X)}^{M}(B)$-labeling $c_B$ of $G$.

One of the following cases occurs:

1. If $A \neq \emptyset$ and $B \neq \emptyset$, then in the line 7 the value of $k_X$ is set to 1 and thus $k = k_A + k_B + 1$. The labeling $c$ of $Y$, defined as follows:

$$c(v) = \begin{cases} 
  c_A(v) & \text{if } v \in A \\
  k_A & \text{if } v \in X \\
  k_A + 1 + c_B(v) & \text{if } v \in B 
\end{cases}$$

is a $(k - 1) - L_{Y}^{M}(Y)$-labeling of $G$.

2. If $A = \emptyset$ and $X \cap Z = \emptyset$, then in the line 8 the value of $k_X$ is set to 1 and thus $k = k_B + 1$. The labeling $c$ of $Y$, defined as follows:

$$c(v) = \begin{cases} 
  0 & \text{if } v \in X \\
  c_B(v) + 1 & \text{if } v \in B 
\end{cases}$$

is a $(k - 1) - L_{Y}^{M}(Y)$-labeling of $G$.

3. If $A = \emptyset$ and $X \cap Z \neq \emptyset$, then in the line 9 the value of $k_X$ is set to 2 and thus $k = k_B + 2$. The labeling $c$ of $Y$, defined as follows:

$$c(v) = \begin{cases} 
  1 & \text{if } v \in X \\
  c_B(v) + 2 & \text{if } v \in B 
\end{cases}$$

is a $(k - 1) - L_{Y}^{M}(Y)$-labeling of $G$.

4. If $B = \emptyset$ and $X \cap M = \emptyset$, then in the line 10 the value of $k_X$ is set to 1 and thus $k = k_A + 1$. The labeling $c$ of $Y$, defined as follows:

$$c(v) = \begin{cases} 
  c_A(v) & \text{if } v \in A \\
  k_A & \text{if } v \in X 
\end{cases}$$

is a $(k - 1) - L_{Y}^{M}(Y)$-labeling of $G$.

5. If $B = \emptyset$ and $X \cap M \neq \emptyset$, then in line 11 the value of $k_X$ is set to 2 and thus $k = k_A + 2$. The labeling $c$ of $Y$, defined as follows:

$$c(v) = \begin{cases} 
  c_A(v) & \text{if } v \in A \\
  k_A & \text{if } v \in X 
\end{cases}$$

is a $(k - 1) - L_{Y}^{M}(Y)$-labeling of $G$ (the label $k_A + 1$ is counted as used, but no vertex is labeled with it).

The case when $X = \emptyset$ is not possible, since the partition $(A, X, B)$ is $G$-correct. The case when $A = B = \emptyset$ is not possible, since then $Y = X$ is a 2-packing in $G$ and by the Lemma 1 $A_{Y}^{M}(Y, G) \leq 3$, so the algorithm would finish in the line 4.
Now let us show that $k \leq \Lambda_M^Y(Y, G)$. Let $c$ be an optimal $L_M^Y(Y)$-labeling of $G$. Let $l$ be the smallest number, such that $|c^{-1}(0)| \cup c^{-1}(l) \cup \cdots \cup c^{-1}(|l|) \geq \frac{|Y|}{2}$.

Let $A = c^{-1}(0) \cup \cdots \cup c^{-1}(l-1)$, $X = c^{-1}(l)$ and $B = c^{-1}(l+1) \cup \cdots \cup c^{-1}(\Lambda_M^Y(Y, G) - 1)$. Notice that $X$ is a 2-packing and $X \neq \emptyset$ by the choice of $l$. Hence we observe that the partition $(A, X, B)$ is $G$-correct, so the algorithm considers it in one of the iterations of the main loop.

Let $c_A: A \to \mathbb{N}$ be a function such that $c_A(v) = c(v)$ for every $v \in A$ and $c_B: B \to \mathbb{N}$ be a function such that $c_B(v) = c(v) - (l + 1)$ for every $v \in B$. Notice that $c_A$ is an optimal $L_N^X(A)$-labeling of $G$ and $c_B$ is an optimal $L_N^X(B)$-labeling of $G$, because otherwise $c$ would not be an optimal.

Hence by the inductive assumption the call in the line 12 returns the number $k_A \leq \Lambda_N^X(A, G)$ and the call in the line 13 returns the number $k_B \leq \Lambda_N^X(B, G)$.

Let $k'$ be the value of $k_A + k_X + k_B$ in the iteration of the main loop when partition $(A, X, B)$ is considered.

Let us consider the following cases:

1. $A, B \neq \emptyset$. In such a case the algorithm **Find-Lambda** sets $k_X = 1$ in the line 4 and

   \[ \Lambda_M^Y(Y, G) = \Lambda_N^X(A, G) + \sum_{c^{-1}(l) = X} + \Lambda_N^X(B, G) \geq k_A + k_X + k_B = k'. \]

2. $A = \emptyset$ and $l = 0$. In such a case $k_A = 0$ and $X \cap Z = \emptyset$ and the algorithm **Find-Lambda** sets $k_X = 1$ in the line 8 and

   \[ \Lambda_M^Y(Y, G) = \Lambda_N^X(A, G) + \sum_{c^{-1}(0) = X} + \Lambda_N^X(B, G) \geq k_A + k_X + k_B = k'. \]

3. $A = \emptyset$ and $l = 1$. In such a case $k_A = 0$ and $X \cap Z \neq \emptyset$. Otherwise $c'$ defined by $c'(v) = c(v) - 1$ for every $v \in Y$ would be a $L_M^Y(Y)$-labeling of $G$ using less labels than the optimal $L_M^Y(Y)$-labeling $c$ of $G$ – contradiction.

   The algorithm **Find-Lambda** sets $k_X = 2$ in the line 9 and

   \[ \Lambda_M^Y(Y, G) = \Lambda_N^X(A, G) + \sum_{c^{-1}(0) = \emptyset} + \sum_{c^{-1}(1) = X} + \Lambda_N^X(B, G) \geq k_A + k_X + k_B = k'. \]

4. $B = \emptyset$ and $l = \Lambda_M^Y(Y, G) - 1$. In such a case $k_B = 0$ and $X \cap M = \emptyset$, and the algorithm **Find-Lambda** sets $k_X = 1$ in the line 10 and

   \[ \Lambda_M^Y(Y, G) = \Lambda_N^X(A, G) + \sum_{c^{-1}(\Lambda_M^Y(Y, G) - 1) = X} + \Lambda_N^X(B, G) \geq k_A + k_X + k_B = k'. \]

5. $B = \emptyset$ and $l = \Lambda_M^Y(Y, G) - 2$. In such a case $k_B = 0$ and $X \cap M \neq \emptyset$ and the algorithm **Find-Lambda** sets $k_X = 2$ in the line 11 and

   \[ \Lambda_M^Y(Y, G) = \Lambda_N^X(A, G) + \sum_{c^{-1}(\Lambda_M^Y(Y, G) - 2) = X} + \sum_{c^{-1}(\Lambda_M^Y(Y, G)) = \emptyset} + \Lambda_N^X(B, G) \geq k_A + k_X + k_B = k'. \]
are positive constants):

Since those are all possible cases and \( k \) is the minimum over values of \( k' \) for all correct partitions, clearly \( k \leq \Lambda^M_Z(Y, G) \).

Observation 1. By the definition of \( \Lambda^M_Z(Y, G) \), the algorithm call \textbf{Find-Lambda}(G, V(G), \emptyset, \emptyset) returns \( \lambda(G) + 1 \).

Lemma 2. Let \( G \) be a graph on \( n \) vertices, \( Y, Z, M \subseteq V(G) \) and let \( y = |Y| \). If \( G^2 \) is computed in advance, the algorithm \textbf{Find-Lambda} finds \( \Lambda^M_Z(Y, G) \) in the time \( O(C^{\log y} y^{3 \log y} y^9y) \) and polynomial space, where \( C \) is a positive constant.

Proof. Having \( G^2 \) computed, checking if any two vertices in \( V(G) \) are in distance at most 2 from each other in \( G \) takes a constant time. Hence verifying if a given \( X \subseteq Y \) is a 2-packing in \( G \) can be performed in the time \( O(y^2) \). Moreover, we can check if a given function \( c: Y \to \mathbb{N} \) is an \( L^M_Z(Y) \) labeling of \( G \) in the time \( O(y^2) \).

Let \( y = |Y| \) be the measure of the size of the problem. Let \( T(y) \) denote the running time of the algorithm \textbf{Find-Lambda} applied to a graph \( G \) and \( Y, Z, M \subseteq V(G) \).

The algorithm \textbf{Find-Lambda} first checks in constant time if \( Y = \emptyset \). Then it exhaustively checks if there exists a \( (k-1)-L^M_Z(Y) \) labeling of \( G \) for \( k \in \{1, 2, 3\} \). There are \( 3^y \) functions \( c: Y \to \{0, 1, 2\} \), so this step is performed in the time \( O(y^2 \cdot 3^y) \).

Then for every \( G \)-correct partition of \( Y \) the algorithm is called recursively for two sets of size at most \( \frac{y}{2} \). Notice that there are at most \( 3^y \) considered partitions. Checking if a partition of \( Y \) is \( G \)-correct can be performed in time \( O(y^2) \). Hence we obtain the following inequality for the complexity (\( C_1 \) and \( C_2 \) are positive constants):

\[
T(y) \leq C_1 y^2 4^y + C_2 y^3 3^y 2 \cdot T \left( \frac{y}{2} \right)
\]

Let \( C = \max(C_1, 2C_2) \), then

\[
T(y) \leq C y^2 4^y + C y^3 3^y \cdot T \left( \frac{y}{2} \right)
\]

It is not difficult to verify that \( T(y) \leq D \cdot C^{\log y} y^{3 \log y} y^9y = O(C^{\log y} y^{3 \log y} y^9y) \), where \( D \) is a positive constant.

The space complexity of the algorithm is clearly polynomial.

Theorem 2. For a graph \( G \) on \( n \) vertices \( \lambda(G) \) can be found in the time \( O((9 + \epsilon)^n) \) and polynomial space, where \( \epsilon \) is an arbitrarily small positive constant.

Proof. The square of a graph \( G \) can be found in the time \( O(n^3) \). By the Observation and Lemma the algorithm \textbf{Find-Lambda} applied to \( G, Y = V(G) \) and \( Z = M = \emptyset \) finds \( \Lambda^0_0(V(G), G) = \lambda(G) - 1 \) in the time \( O(C^{\log n} n^{3 \log n} n^9n) = O((9 + \epsilon)^n) \) and polynomial space.
Remark

We have just learned that results similar to those included in this paper were independently obtained (but not published) by Havet, Klazar, Kratochvíl, Kratsch and Liedloff [15].

References

[1] Björklund, A., Husfeldt, T., Koivisto, M.: Set Partitioning via Inclusion-Exclusion. SIAM J. Comput. 39 (2009), pp. 546–563
[2] Bodlaender, H.L., Kratsch D: An exact algorithm for graph coloring with polynomial memory. UU-CS 2006-015 (2006)
[3] Calamoneri, T.: The L(h,k)-Labelling Problem: A Survey and Annotated Bibliography. Computer Journal 49 (2006), pp. 585–608
[4] Eggeman, N., Havet, F., Noble, S.: k-L(2,1)-Labelling for Planar Graphs is NP-Complete for k ≥ 4. Discrete Applied Mathematics 158 (2010), pp. 1777–1788.
[5] Fiala, J., Golovach, P., Kratochvíl, J.: Distance Constrained Labelings of Graphs of Bounded Treewidth. Proceedings of ICALP 2005, LNCS 3580 (2005), pp. 360–372.
[6] Fiala, J., Kloks, T., Kratochvíl, J.: Fixed-parameter complexity of λ-labelings. Discrete Applied Mathematics 113 (2001), pp. 59–72.
[7] Fiala, J., Kratochvíl, J.: Locally constrained graph homomorphisms - structure, complexity, and applications. Computer Science Review 2 (2008), pp. 97–111.
[8] Fiala, J., Kratochvíl, J.: On the Computational Complexity of the L(2,1)-Labeling Problem for Regular Graphs Proceedings of ICTCS 2005, LNCS 3703 (2005), pp. 228–236
[9] Gonçalves, D.: On the L(p; 1)-labelling of graphs. Discrete Mathematics 308 (2008), pp. 1405–1414
[10] Griggs, J. R., Král, D.: Graph labellings with variable weights, a survey. Discrete Applied Mathematics 157 (2009), pp. 2646–2658.
[11] Griggs, J. R., Yeh, R. K.: Labelling graphs with a condition at distance 2. SIAM Journal of Discrete Mathematics 5 (1992), pp. 586–595.
[12] Hale, W.K.: Frequency assignemnt: Theory and applications. Proc. IEEE 68 (1980), pp. 1497–1514
[13] Havet, F., Reed, B., Sereni, J.-S.: L(2,1)-labellings of graphs. Proceedings of SODA 2008 (2008), pp. 621–630.
[14] Havet, F., Klazar, M., Kratochvíl, J., Kratsch, D., Liedloff, M.: Exact algorithms for $L(2,1)$-labeling of graphs. *Algorithmica* 59 (2011), pp. 169–194.

[15] Havet, F., Klazar, M., Kratochvíl, J., Kratsch, D., Liedloff, M.: private communication

[16] Junosza-Szaniawski K., Kratochvíl J., Liedloff M., Rossmanith P., Rzążewski P.: Fast Exact Algorithm for $L(2,1)$-Labeling of Graphs. *Proceedings of TAMC 2011*, LNCS 6648 (to appear)

[17] Junosza-Szaniawski K., Rzążewski P.: On Improved Exact Algorithms for $L(2,1)$-Labeling of Graphs. *Proceedings of IWOCA 2010*, LNCS 6460 (2011), pp. 34–37

[18] Junosza-Szaniawski K., Rzążewski P.: On the Complexity of Exact Algorithm for $L(2,1)$-labeling of Graphs. *Information Processing Letters* (to appear). Preliminary version in KAM-DIMATIA Preprint Series 2010-992

[19] D. Král’: Channel assignment problem with variable weights. *SIAM Journal on Discrete Mathematics* 20 (2006), pp. 690–704.

[20] Yeh, R.: A survey on labeling graphs with a condition at distance two. *Discrete Mathematics* 306(2006), pp. 1217–1231.