Fixed-parameter complexity of λ-labelings

Jiří Fiala[1]*, Ton Kloks[2]**, and Jan Kratochvíl[1],

[1] Department of Applied Mathematics and DIMATIA
Charles University
Malostranské nám. 25, 11800 Praha 1
Czech Republic
{honza,fiala}@kam.ms.mff.cuni.cz
[2] Department of mathematics and Computer Science
Vrije Universiteit
De Boelelaan 1081 A, 1081 HV Amsterdam
The Netherlands
kloks@cs.vu.nl

Abstract. A λ-labeling of a graph G is an assignment of labels from the set $\{0, \dots, \lambda\}$ to the vertices of a graph G such that vertices at distance at most two get different labels and adjacent vertices get labels which are at least two apart. We study the minimum value $\lambda = \lambda(G)$ such that G admits a λ-labeling. We show that for every fixed value $k \geq 4$ it is NP-complete to determine whether $\lambda(G) \leq k$. We further investigate this problem for sparse graphs (k-almost trees), extending the already known result for ordinary trees.

In a generalization of this problem we wish to find a labeling such that vertices at distance two are assigned labels that differ by at least q and the labels of adjacent vertices differ by at least p (where p and q are given positive integers). We denote the minimum number of labels by $L(G; p, q)$ (hence $\lambda(G) = L(G; 2, 1)$). We show several hardness results for $L(G; p, q)$ including that for any $p > q \geq 1$ there is a $\lambda = \lambda(p, q)$ such that deciding if $L(G; p, q) \leq \lambda(p, q)$ is NP-complete.

1 Introduction

Radio frequency assignment is a widely studied area of research. The task is to assign radio frequencies to transmitters at different locations without interference. In a classical setting the problem is closely related to graph coloring where the vertices of a graph represent the transmitters and adjacencies indicate possible interference.

In many practical cases one wishes to assign frequencies such that the difference between frequencies assigned to two transmitters depends on the distance between the transmitters. In this paper we assume that the distances between

* Research supported in part by the Czech Research Grants GAUK 194 and GAČR 201/1996/0194.
** Supported by DIMATIA and GAČR 201/99/0242

pairs of transmitters can be modeled by the distance between the corresponding vertices of a graph. In this framework the problem is to find a non-negative real-valued function f on vertices of a graph such that if u and v are vertices at distance i, the values $f(u)$ and $f(v)$ should differ by at least a prescribed integer p_i (see [10]). The smallest possible maximal value of such a function f over all vertices of the graph G is denoted by $L(G; p_1, p_2, ..., p_k)$.

Roberts proposed the problem of assigning frequencies (integers) to transmitters such that transmitters that are close to each other receive different channels and transmitters that are *very* close together receive radio channels that are at least two apart (see [9, 11, 20]). In this paper we study the fixed parameter tractability of this $L(2,1)$-problem and also its generalized $L(p,q)$ version.

In [9] the $L_d(2,1)$-labeling problem was introduced, which is the problem to determine $\lambda(G,d)$, i.e. the smallest number m such that G has an $L_d(2,1)$-labeling f with $\max\{f(x) \mid x \in V\} = m$. An $L_d(2,1)$-labeling of a graph $G = (V,E)$ is a function $f : V \to [0,\infty)$ such that whenever $x, y \in V$ are adjacent $|f(x) - f(y)| \geq 2d$ and whenever the distance between x and y is two, $|f(x) - f(y)| \geq d$. It is shown that it suffices to consider integral-valued labelings. We denote $\lambda(G,1) = L(G; 2, 1)$ by $\lambda(G)$.

The version of the problem where the labels that are used are in a consecutive range, was studied in [18].

As far as special graph classes are concerned, $\lambda(G)$ can be determined efficiently for paths, cycles and wheels [9]. Bounds for cubes were obtained in [11, 9, 19]. It is conjectured that $\lambda(Q_n) = n + 3$ for $n \geq 3$ in [9]. Non-trivial algorithms were also found for cographs [4]. In [11] planar graphs and 3-regular graphs were considered. It was conjectured that the problem is NP-complete for trees in [9], however shown to be polynomial in [4]. In this paper we show that there is a polynomial time algorithm for k-almost trees, for every fixed integer k, thereby extending the result of Chang and Kuo.

In case the parameter is *not* fixed, the most interesting and general result was obtained in [7]. It is shown that $\lambda(G) \leq n - 1$ if and only if $\pi(\overline{G}) = 1$ and $\lambda(G) = n + r - 2$ if and only if $\pi(\overline{G}) = r$, where $n = |V|$, $\pi(G)$ is the path cover number of a graph G and $\overline{G}$ is the complement of G (see also [20]).

In [9, 20] it was shown that determining $\lambda(G)$ is an NP-complete problem even for graphs G with diameter two. In this paper we study the fixed parameter tractability. Answering a question of [20], we show that it is NP-complete to determine if $\lambda(G) \leq k$ for every fixed integer $k \geq 4$. A fortiori, the problem is not in FPT (see, e.g., [5]). Our observations fill in important blank spots in the complexity overview of the λ-labeling problem. It is easy to see that $\lambda(G) \leq 3$ if and only if G is a disjoint union of paths of length at most 3.

Much less is known for the more general $L(p,q)$-problem. Special instances such as the $L(0,1)$-problem and the $L(1,1)$-problem were already considered in [6, 20]. Some results on heuristics for the $L(p,q)$ problem appeared in [2, 6]. We show the first complexity results for this problem.

2 Preliminaries

Definition 1. Let G be a graph and let $d_1, \ldots, d_k$ be a collection of integers. A λ*-labeling* is an assignment of labels from the set $\{0, \ldots, \lambda\}$ to the vertices of G. We say that the labeling satisfies constraints $(d_1, \ldots, d_k)$ if each pair of vertices at distance i in the graph gets labels that differ by at least d_i. In such a case we talk about $\lambda_{(d_1,\ldots,d_k)}$*-labelings*. The minimum value λ for which G admits a $\lambda_{(d_1,\ldots,d_k)}$-labeling is denoted by $L(G; d_1, \ldots, d_k)$. We refer to the problem of deciding the existence a $\lambda_{(d_1,\ldots,d_k)}$-labeling as the $L(d_1, \ldots, d_k)$*-problem*.

For the sake of history (and also for convenience) we write $\lambda(G)$ instead of $L(G; 2, 1)$.

Definition 2. A λ-labeling is called *consecutive* if the labels that are used are consecutive.

Clearly, a consecutive λ-labeling does not always exist. Consecutive labelings were studied in [18]. In general, it can be shown [18] that $G = (V, E)$ has a Hamiltonian path if and only if $\overline{G}$ (the complement of G) has a consecutive $\lambda_{(2,1)}$-labeling with $\lambda \leq |V| - 1$. In this paper we show that the fixed parameter variant with $\lambda \geq 4$ is equally difficult. Most of our NP-hardness results are obtained via graph covers.

Definition 3. Let G be a graph and H be a multigraph. The *H-COVER problem* asks for a local isomorphism $f : G \to H$, i.e., a homomorphism that maps edges incident with every vertex x isomorphically to edges incident with $f(x)$.

The study of the complexity of the H-COVER problem for particular parameter multigraphs H was initiated in [1] and carried on in a series of papers [12–15]. In particular, the K_t-COVER problem is shown NP-complete for every fixed $t \geq 4$ in [12]. Note that a $(t-1)$-regular graph G covers the complete graph K_t if and only if the vertices of G can be assigned t different colors in such a way that closed neighborhood of each vertex is assigned all t colors, i.e., if and only if $L(G, 1, 1) = t - 1$. The immediate consequence for the complexity of the $L(1, 1)$ problem follows:

Proposition 4. *The problem if $L(G; 1, 1) \leq \lambda$ is NP-complete for every fixed $\lambda \geq 3$.*

Note also that this problem asks for the chromatic number of the square G^2 of G. For an alternative proof of NP-completeness we refer to [17] (the chromatic number problem is NP-complete for all powers).

We find convenient to reformulate in terms of graph coloring a class of particular graph covering problems which will be used several times in the hardness proofs:

Definition 5. Let $k \geq 3$ be an integer. The $BW(k)$–problem is the following 'black and white coloring' problem:

Instance: A k-regular graph G.
Question: Is there a coloring of the vertices of G with labels black and white such that every vertex has exactly two neighbors with the same color and $k-2$ neighbors with opposite colors.

(The $BW(k)$–problem is equivalent to the H-COVER problem for H being the multigraph with two vertices of degree k, joined by $k-2$ parallel edges and having a loop at each of the two vertices.) The following theorem was proved in [14].

Proposition 6. *The $BW(k)$–problem is NP-complete for every $k \geq 3$.*

The paper is organized as follows. In Section 3 we present the positive result, namely a polynomial algorithm for sparse graphs (k-almost trees). The algorithm is polynomial both in λ and the size of the input graph, i.e., the maximum label bound λ is regarded as part of the input. In Section 4 we present the hardness results for fixed λ. Section 5 is devoted to the general $L(p,q)$ problem. Here we only state our results and the proofs are postponed to the Appendix.

3 $L(2,1)$ for sparse graphs

For trees T, $\lambda(T)$ can be computed efficiently [4]. We show how this algorithm can be used to solve the problem for a slightly wider class of graphs, namely k-almost trees for fixed values of k. (A *k-almost tree* is a connected graph G with $n+k-1$ edges. Clearly, k-almost trees can be recognized in linear time, cf. e.g., [3, 16]). Let us remark in this connection that the $L(2,1)$ problem seems to be harder than many other graph-theoretical problems, at least as concerns tree-like graphs. Many generally difficult problems can be solved in polynomial time for graphs of bounded tree-width, whereas the complexity of $L(2,1)$ for such graphs is unknown (when λ is part of the input, as assumed in this section).

Theorem 7. *Let G be a k-almost tree with n vertices. Then $L(G;2,1) \leq \lambda$ can be tested in $O(\lambda^{2k+\frac{9}{2}}n)$ time.*

Proof. Let G be a k-almost tree. Choose a spanning tree of G and let $e_1, e_2, ..., e_k$ be the edges of G that do not belong to the spanning tree.

In order to use the tree algorithm of Chang and Kuo, we build a new tree T by modifying the spanning tree. For each edge $e_i = (u_i, v_i)$, add two leaves $e'_i = (u_i, x_i)$ and $e''_i = (v_i, y_i)$ to the spanning tree, so that x_i and y_i are new leaves, and denote the new tree by T. Note that identifying vertices u_i with y_i and v_i with x_i gives the original graph G. Clearly T has a $\lambda_{(2,1)}$-labeling such that the labels of u_i and y_i and of v_i and x_i are equal for all $i = 1, \ldots, k$, if and only if the original graph G has a proper $\lambda_{(2,1)}$-labeling.

To test whether T has a $\lambda_{(2,1)}$-labeling we use the following procedure: Fix a pre-labeling on the endvertices of all the edges of $\{e'_i, e''_i \mid 1 \leq i \leq k\}$ such that u_i and y_i and also v_i and x_i have the same labels and test whether there exists a

$\lambda_{(2,1)}$-labeling of the tree T compatible with this pre-labeling. Repeat this step for all possible pre-labelings. (Note that there are $O(\lambda^{2k})$ different pre-labelings.)

Since the tree-labeling algorithm of [4] does not involve pre-labeled vertices, and also for the sake of completeness, we describe our modification for trees with pre-labeled vertices.

Choose a leaf as a root of T. Use dynamic programming from the leaves up to the root of the tree. For each edge (x, y) of T, with x closer to the root and every pair of labels a and b, determine if there is a $\lambda_{(2,1)}$-labeling of the subtree induced by x, y and the descendants of y, with vertex x labeled a and y labeled b. Assuming that we know all admissible labelings on the edges yz, for z children of y, we can use a matching algorithm to find if there is an admissible labeling. (For each z, let M_z be the set of labels different from a that are compatible with y labeled b. The matching algorithm tests if the set system M_z, z child of y, has a system of distinct representatives. Such representatives would be labels for the children of y. If such a system of distinct representatives does not exists, this pair of labels (a, b) is not feasible for the edge (x, y).) The tree T allows a $\lambda_{(2,1)}$-labeling if and only if the edge incident to the root allows at least one feasible labeling. The matching algorithm takes at most $O(\lambda^{\frac{5}{2}})$ time since it needs to be executed only for vertices y of degree at most $\lambda - 1$ (see, e.g., [16]). If no endvertex of (x, y) is pre-labeled, the matching problem has to be solved $O(\lambda^2)$ times. (Clearly, the algorithm does not consider other labels for an already pre-labeled vertex.)

It follows that this algorithm can be implemented to run in $O(n\lambda^{2k+\frac{9}{2}})$ time. □

4 Fixed parameter λ-labeling is NP-complete

Griggs and Yeh showed in [9] that it is NP-complete to determine whether $\lambda(G) \leq |V(G)|$ for a general graph G. In this section we focus on the fixed parameter case.

Theorem 8. *For each $\lambda \geq 4$, it is NP-complete to decide if the input graph allows a $\lambda_{(2,1)}$-labeling.*

Proof. We prove our result by induction on λ. The base cases are $\lambda = 4$ and $\lambda = 5$. For these cases we use a reduction from the H-cover problem for multigraphs H_1 and H_2 respectively (see Fig. 2 in the Appendix), which are equivalent to the $BW(3)$ and $BW(4)$ coloring problems and are both NP-complete (Prop. 6).

$\underline{\lambda = 4}$: We use a reduction from $BW(3)$. Let G be a 3-regular graph for which we want to find a black and white coloring such that every white vertex has exactly two white and one black neighbor and every black vertex has one white and two black neighbors.

We claim that the graph G' obtained by subdividing each edge of G by two new vertices of degree 2, has a $4_{(2,1)}$-labeling if and only if G covers H_1 (cf. an illustrative picture in Fig. 3 in the Appendix).

First assume that G' has a $4_{(2,1)}$-labeling. The original vertices of G have degree 3, so they must be labeled either by number 0 or by 4. An easy case analysis shows that the only possible labelings of the paths of length 3 replacing the edges of G are $(0,2,4,0)$, $(0,3,1,4)$, $(4,0,2,4)$ and the reverse 4-tuples.

Consider a vertex u of G in G'. Say u is labeled 0. Its neighbors (in G') have to be assigned different labels, hence each of the labels $2,3,4$ is used on exactly one of the neighbors. Therefore, the three paths leaving u are labeled $(0,2,4,0)$, $(0,3,1,4)$ and $(0,4,2,0)$, and two of them lead to vertices labeled 0 and the third one to a vertex labeled 4. Similarly, if u is labeled 4 then two of the paths starting in u lead to vertices labeled 4 and one to a vertex labeled 0. Thus coloring every vertex labeled 0 with white color (in G), and coloring every vertex labeled 4 with black color translates the given 4-labeling into a black and white coloring of the original graph, satisfying $BW(3)$.

Now assume that a black and white coloring of G is given (satisfying $BW(3)$). We can partition the edges of G into three sets: edges with two white end-vertices, edges with two black end-vertices and "black-white edges". In G', we assign label 0 to every white vertex of G, and label 4 to every black vertex of G. It remains to label the intermediate vertices on the paths replacing edges of G. Each path corresponding to a black-white edge in G, will be labeled by $(0,3,1,4)$. The paths corresponding to white edges in G induce a disjoint union of cycles in G' (every white vertex has two white neighbors in G). On each of these cycles, we use consecutively the scheme $(0,2,4,0)$. Thus every vertex labeled 0 has exactly one neighbor labeled 2, one neighbor labeled 3 and one neighbor labeled 4, i.e., common neighbors of a vertex are assigned different labels. Similarly, the paths corresponding to the black edges of G induce a disjoint union of cycles, and as such will be consecutively labeled $(4,0,2,4)$. Since adjacent vertices (in G') are always assigned labels which are at least 2 apart, this gives a $4_{(2,1)}$-labeling of G'.

$\lambda = 5$: We reduce from the $BW(4)$–coloring problem.

Let $G = (V,E)$ be a 4-regular graph. Replace each edge in E by the graph depicted in Figure 1 a). An easy case analysis shows that this auxiliary graph allows 6 combinations depicted in Figure 1 b), subject to the condition that vertices u and v are labeled 0 or 5 (this must be the case in any $5_(2,1)$-labeling of G' since these vertices have degree 4).

Call the new graph G'. We claim that G' has a $5_{(2,1)}$-labeling if and only if G has an H_2-cover.

Consider a $5_{(2,1)}$-labeling of G'. The four neighbors of a vertex labeled 0 must be labeled $2,3,4,5$, and so the paths starting in such a vertex are labeled $(0,2,5,0)$, $(0,3,1,5)$, $(0,4,x,5)$ and $(0,5,2,0)$ (here x is either 1 or 2). Hence two of these paths lead to vertices labeled 0 and the other two lead to vertices labeled 5. Regarding label 0 as white color and label 5 as black color, we see that the restriction of the labeling to the original vertices of G solves the $BW(4)$–problem for G in the affirmative way.

Now assume G covers H_2-cover, i.e., we assume a black-white coloring of G which satisfies $BW(4)$. The edges of G can be partitioned into edges with two

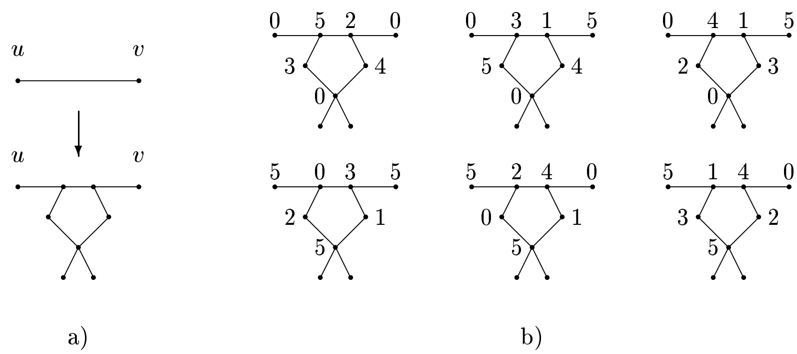

Fig. 1. Replacement for $\lambda = 5$ case and its 5-labeling

white end-vertices, edges with two black end-vertices, and black-white edges. Similarly as in the case $\lambda = 4$, the white-white edges induce a disjoint union of cycles and the corresponding paths can be labeled consecutively $(0, 5, 2, 0)$. The paths corresponding to black-black edges will be labeled $(5, 0, 3, 5)$. The black-white edges induce a 2-regular bipartite graph, whose edges can be partitioned into two perfect matchings. Label the paths corresponding to edges of one of the matchings by $(0, 3, 1, 5)$, and the paths corresponding to the other matching by $(0, 4, 2, 5)$. (Note that the two rightmost labelings of the Figure 1 are not used.) In this way we obtain a $5_{(2,1)}$-labeling for G'.

Induction step: Assume that $G = (V, E)$ is a graph whose $\lambda_{(2,1)}$-labeling is the question. We construct a graph G' which can be $(\lambda + 2)_{(2,1)}$-labeled if and only if G can be $\lambda_{(2,1)}$-labeled.

First construct a complete binary tree T with at least $|V(G)|$ leaves, all of them in the bottom layer. Call the layers, from the bottom to the top, $L_1, \ldots, L_k$.

Add a layer L_0 with $|L_1|$ vertices, and connect these vertices by a perfect matching to the vertices of L_1. Subdivide each edge of T (including the edges between L_0 and L_1) by a new extra vertex. Connect each vertex of G to a unique vertex in L_0. Finally, add leaves to vertices in layers L_i so that their degrees add up to $\lambda + 1$. This completes the construction of G'.

Assume that G' is $(\lambda + 2)_{(2,1)}$-labeled. Vertices from layers L_i have degree $\lambda+1$ and therefore each of them is labeled either by 0 or by $\lambda+2$. Furthermore all vertices in the same layer have the same label. Assume without loss of generality that the vertices of L_0 are labeled with label $\lambda + 2$. Then all vertices of G have labels in $\{0, \ldots, \lambda\}$ and we obtain a $\lambda_{(2,1)}$-labeling of G.

Now assume that G can be $\lambda_{(2,1)}$-labeled. Color the vertices from layers L_i with an even index i with $\lambda + 2$ and those from layer with an odd index with 0. Color the vertices between layers L_i and L_{i+1}, $i \geq 1$ arbitrary (but feasible). A vertex between layers L_0 and L_1 has at most 6 forbidden labels: $0, 1, \lambda+1, \lambda+2$, and two more labels of vertices at distance two: of the vertex in G and of the vertex between L_1 and L_2 (see Fig. 4 in appendix). Since $\lambda \geq 4$ at least one label is available. Finally, label the leaves at the vertices from layers L_i. Since such a vertex is labeled with 0 or with $\lambda + 2$ and since such a vertex has degree $\lambda + 1$, there is always an available label for every leaf. This gives a $(\lambda + 2)$-labeling of G'. □

As we mentioned in Section 2, sometimes it is important to require that the labels actually used in a labeling are consecutive. An easy argument shows that consecutive $\lambda_{2,1}$-labelings are not easier.

Corollary 9. *The problem whether there exists a consecutive $\lambda_{(2,1)}$-labeling for a given graph G is NP-complete.*

Proof. Let H_λ be the tree containing two vertices of degree $\lambda - 1$ and with $2(\lambda - 2)$ leaves.

For a graph G, consider the disjoint union $G' = G + H_\lambda$. Obviously G' admits a consecutive $\lambda_{(2,1)}$-labeling if and only if G admits a $\lambda_{(2,1)}$-labeling. Hence, the existence of consecutive $\lambda_{(2,1)}$-labeling is also NP-complete for every fixed $\lambda \geq 4$. □

5 Complexity of the generalized distance two labeling

Recall that the $L(p, q)$ problem asks for a λ-labeling such that adjacent vertices receive labels which are at least p apart, and vertices at distance 2 receive labels which differ by at least q. This is already the general frequency assignment problem restricted to relevant distance 2 (in the $(p_1, p_2, \ldots, p_k)$ condition). Nobody would expect the $L(p, q)$-labeling problem for $p > q \geq 1$ to be easier than $L(2, 1)$, however, the actual NP-hardness proofs seem tedious and not easily achievable in full generality. We conjecture:

Conjecture 10. *For any $p \geq q \geq 1$ there is a $\lambda(p, q)$ such that deciding if $L(G; p, q) \leq \lambda$ is NP-complete for every fixed $\lambda \geq \lambda(p, q)$.*

In this section we gather results which support our conjecture, the proofs are mostly technical and left to the Appendix. We also discuss the complexity of $L(p, q)$ for trees and almost trees.

We first aim at restricting the range of parameters p, q. The following easy generalization of the special case $p = 2, q = 1$ proven in [9] shows that we may restrict our attention to the case of p, q being relatively prime.

Observation 11. *Let p, q, and k be integers. Then $L(G; kp, kq) = kL(G; p, q)$. Hence $L(p, q)$ and $L(kp, kq)$ are polynomially equivalent.*

The case $p = q$ is thus fully understood (NP-complete for $\lambda \geq 3p$ and polynomial for $\lambda < 3p$, cf. Proposition 4 and Observation 11). So is now the case $p = 2q$ (NP-complete for $\lambda \geq 4q$ and polynomial for $\lambda < 4q$).

For general p, q, we can prove that there is at least one NP-complete fixed parameter instance:

Theorem 12. *For all fixed $p > q \geq 1$, it is NP-complete to decide whether $L(G; p, q) \leq p + q\lceil \frac{p}{q} \rceil$.*

And under slightly less general conditions we can prove that there are infinitely many hard instances:

Theorem 13. *The problem whether $L(G; p, q) \leq p + kq$ is NP-complete for any fixed $k \geq \frac{p}{q}$ and $p > 2q$.*

It follows that for $q = 1$ (and more generally p divisible by q), there are only finitely many polynomial instances (unless of course P=NP), namely if $p > 2$ then the $L(p, 1)$-problem is NP-complete for every fixed $\lambda \geq 2p$. In this case we are able to prove a little more:

Theorem 14. *For every $p > 2$, the $L(p, 1)$-problem is NP-complete for $\lambda \geq p+5$ and polynomially solvable for $\lambda \leq p + 2$.*

For $p \geq 5$, this result leaves the cases $\lambda = p+3$ and $\lambda = p+4$ as the last open cases for the fixed parameter complexity of $L(p, 1)$ (for $p \leq 4$ the bound $\lambda \geq 2p$ is better than $\lambda \geq p + 5$, and $\lambda = 7$ is the only open case for $p = 4$, while $L(3, 1)$ is fully understood being polynomial for $\lambda \leq 5$ and NP-complete for $\lambda \geq 6$).

Finally we comment on the complexity of the $L(p, q)$-problem on trees. Obviously, for fixed λ, the problem is solvable in linear time. When λ is part of the input and $q = 1$, then the same algorithm as we described in Section 3 can be applied to trees and k-almost trees (in the recursive step we only consider labelings (a, b) of the currently processed edge xy such that $|a - b| \geq p$). Hence we get:

Corollary 15. *The $L(p, 1)$-problem is polynomially solvable for k-almost trees for every fixed k.*

However, this algorithm is not of immediate help in case of $q > 1$. In this case we would need to be able to decide the existence of a system of *distant* (rather than distinct) representatives (in the language of the proof of Theorem 7, we would need to decide if the set system M_z, z children of y, allows representatives $a(z) \in M_z$ such that $|a(z) - a(z')| \geq q$ for any two distinct children z, z' of y). Unfortunately, this variant of bipartite matching is NP-complete (easy reduction from SAT). Therefore the complexity of $L(p, q)$ for trees remains (for $q > 1$) open.

References

1. Abello J., M.R. Fellows and J.C. Stillwell, On the complexity and combinatorics of covering finite complexes, *Australasian Journal of Combinatorics* **4** (1991), 103-112
2. Battiti, Roberto, Alan A. Bertossi, and Maurizio A. Bonuccelli, Assigning codes in wireless networks: Bounds and scaling properties. Manuscript 1998.
3. Brandstädt, Andreas, Special graph classes – A survey, Schriftenreihe des Fachsbereichs Mathematik, SM-DU-199 (1991), Universität Duisburg Gesamthochschule.
4. Chang, G. J. and D. Kuo, The $L(2,1)$-labeling problem on graphs, *SIAM J. Disc. Math.* **9**, (1996), pp. 309–316.
5. Downey, R. G. and M. R. Fellows, *Parameterized Complexity*, (eds. D. Gries and F. Schneider), Monographs in Computer Science, Springer–Verlag New York, 1998.
6. Fotakis, Dimitris, Grammati Pantziou, George Pentaris, and Paul Spirakis, Frequency assignment in mobile and radio networks. Manuscript 1998.
7. Georges, J. P. and D. W. Mauro, On the size of graphs labeled with a condition at distance two, *Journal of Graph Theory* **22**, (1996), pp. 47–57.
8. Georges, J. P., D. W. Mauro and M. A. Whittlesey, Relating path coverings to vertex labellings with a condition at distance two, *Discrete Mathematics* **135**, (1994), pp. 103–111.
9. Krzysztof Giaro, The complexity of consecutive Δ-labeling of bipartite graphs: 4 is easy, 5 is hard, *Ars Combinatorica* **47**, (1997), pp. 287–298.
10. Griggs, J. R. and R. K. Yeh, Labelling graphs with a condition at distance 2, *SIAM J. Disc. Math.* **5**, (1992), pp. 586–595.
11. Heuvel, J. van den, R. A. Leese, and M. A. Shepherd, Graph labeling and radio channel assignment, `http:// www.maths.ox.ac.uk / users / gowerr / preprints.html`, 1996.
12. Jonas, Kimball, *Graph colorings analogues with a condition at distance two: $L(2,1)$-labelings and list λ-labelings*, Ph.D. thesis, University of South Carolina, 1993.
13. Kratochvíl, Jan, Regular codes in regular graphs are difficult, *Discrete Math.* **133** (1994), 191-205
14. Kratochvíl, Jan, Andrzej Proskurowski, and Jan Arne Telle, Covering regular graphs *Journal of Combin. Theory Ser. B* 71 (1997), 1-16
15. Kratochvíl, Jan, Andrzej Proskurowski, and Jan Arne Telle, Covering directed multigraphs I. Colored directed multigraphs, In: *Graph-Theoretical Concepts in Computer Science, Proceedings 23rd WG '97, Berlin, Lecture Notes in Computer Science 1335*, Springer Verlag, (1997), pp. 242-257.
16. Kratochvíl, Jan, Andrzej Proskurowski, and Jan Arne Telle, Complexity of graph covering problems *Nordic Journal of Computing* **5** (1998), 173-195
17. Leeuwen, Jan van, Graph algorithms. In: J. van Leeuwen (ed.) *Handbook of Theoretical Computer Science, A: Algorithms and Complexity*, Elsevier Science Publ., Amsterdam 1990, pp. 527–631.
18. Lin, Yaw-Ling and Steven S. Skiena, Algorithms for square roots of graphs, *SIAM J. Discrete Math.* **8**, (1995), pp. 99–118.
19. Liu, D. D.-F. and R. K. Yeh, On distance two labellings of graphs, *ARS Combinatorica* **47**, (1997), pp. 13–22.
20. Whittlesey, M. A., J. P. Georgess and D. W. Mauro, On the lambda-coloring of Q_n and related graphs, *SIAM J. Disc. Math.* **8**, (1995), pp. 499–506.
21. Yeh, Kwan-Ching, Labeling graphs with a condition at distance two, Ph.D. Thesis, University of South Carolina, 1990.

A Appendix

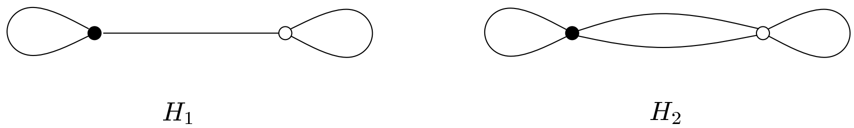

Fig. 2. Multigraphs H_1 and H_2.

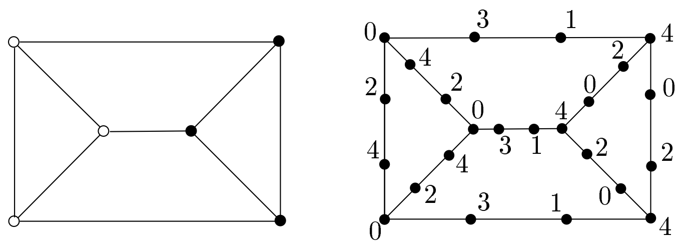

Fig. 3. $BW(3)$-coloring of the prism and labeling of its prime-graph.

Proof of Theorem 12

Theorem: *It is NP-complete to decide whether* $L(G;p,q) \leq p + q\lceil \frac{p}{q} \rceil$ *for all* $p > q \geq 1$.

Proof. We use a reduction from the $BW(k+2)$–problem, i.e., that of finding a black and white coloring of the vertices of a $(k+2)$-regular graph, such that every vertex has two neighbors with the same color and k neighbors with opposite color.

Let $k = \lceil \frac{p}{q} \rceil - 1$ and let G be a $(k+2)$-regular graph for which we want to find a black and white coloring. Replace each edge of G with a P_4. Call this new graph G'. Assume we have a $\lambda_{(p,q)}$-labeling for G' with $\lambda = p + (k+1)q$. Notice that $\lambda < 2p + q$. The original vertices must be labeled by 0 or λ and their neighbors have labels $p, p+q, \ldots, p+kq, \lambda$ or $0, q, 2q, \ldots, \lambda - p$.

We show that a P_4 starting at a vertex with label 0 and ending in vertex labeled by 0 or by λ can be labeled only by one of the following schemes $(0, p, \lambda, 0)$, $(0, p+iq, jq, \lambda)$ for $1 \leq i \leq k$ and $1 \leq j \leq i$, and $(0, \lambda, p, 0)$.

Suppose the path has vertices w, x, y, z and suppose $f(w) = 0$, where f denotes the labeling function. Clearly $f(x) \geq p$ and $f(y) \geq q$.

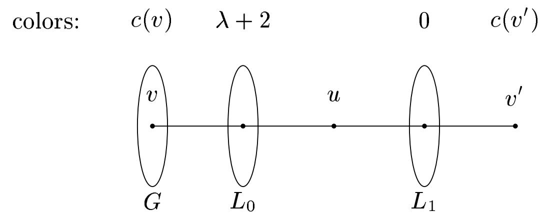

Fig. 4. Induction step of the $L(G, 2, 1)$-problem. Labeling the vertex u, forbidden labels are $\{0, 1, c(v), c(v'), \lambda+1, \lambda+2\}$

First assume $f(x) = p$. Then $f(y) \geq 2p$. If $f(z) = \lambda$ then $f(y) \leq \lambda - p < p+q < 2p$ which is a contradiction. Hence $f(z) = 0$. Now $f(y) \in \{p, p+q, \ldots, p+kq, \lambda\}$ and $f(y) \geq 2p$ imply $f(y) = \lambda$.

Now assume that $f(x) = p + iq$ for some $1 \leq i \leq k$. Then $q \leq f(y) \leq iq$. Now, $f(z) = 0$ would imply $f(y) = p + jq$ for some $0 \leq j \leq k+1$. But then $|f(x) - f(y)| = |i - j|\, q \leq kq < p$ which is a contradiction. Hence $f(z) = \lambda$ and $f(y) = jq$ for some $1 \leq j \leq i$.

Because the degree of an original vertex is $k+2$, two of the above mentioned paths lead to a vertex with the same label and the other paths to vertices with complementary label.

Now assume a black and white coloring of G is given. This coloring can be easily extended to an $\lambda_{(p,q)}$-labeling of G'. The subgraph induced by white vertices is a set of cycles and the corresponding P_4's can be colored by $(0, p, \lambda, 0)$ consecutively along the cycles. Similarly for the subgraph induced by black vertices. Edges joining black and white vertices form a bipartite k-regular graph. Hence this graph is k-edge colorable. Fix a coloring by colors $1, 2, \ldots, k$ and use the sequence $(0, p+iq, iq, \lambda)$ on the path corresponding to an edge labeled by i. □

Proof of Theorem 13

Theorem: *The problem whether* $L(G; p, q) \leq p + kq$ *is NP-complete for all* $p > 2q$ *and any integer* $k \geq \frac{p}{q}$.

Proof. We consider two cases.

Case $2q < p \leq 3q$ We use reduction from the $BW(3)$–problem.

Let G be an input graph for the black and white coloring problem. Replace each edge by the graph H in Figure 5 and add leaves to the original vertices to add up their degree to $k+1$.

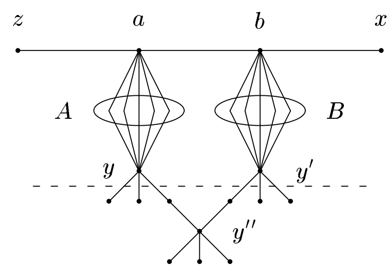

Fig. 5. Edge gadget for the $L(p,q)$-labeling problem

The case analysis shows that on the path (z,a,b,x) only the following sequences of colors can appear: $(0,p,\lambda,0)$, $(0,\lambda,p,0)$, $(0,\lambda-q,q,\lambda)$, $(\lambda,0,\lambda-p,\lambda)$ and $(\lambda,\lambda-p,0,\lambda)$.

Similarly as in the proof of Theorem 8, it follows that a proper black and white coloring exists if and only if the modified graph allows a $\lambda_{(p,q)}$-labeling.

Case $p > 3q$ In this case we use a reduction from $BW(4)$.

Let G be a 4-regular graph. Use the same edge replacement as in the above case, also add leaves to the original vertices to degrees $k+1$.

In this case a case analysis gives more possible label schemes of (z,a,b,x), namely $(0,p,\lambda,0)$, $(0,p+iq,q,\lambda)$, $(0,\lambda-q,iq,\lambda)$, $(0,\lambda,p,0)$ and their reversals with vertex z labeled by λ. In the above schemes i belongs to the interval $[1,\lfloor\frac{p}{q}\rfloor-1]$.

Suppose that a $\lambda_{(p,q)}$-labeling of G' exists. Let the original vertices labeled by 0 have white color and those with label λ be colored black.

W.l.o.g. we suppose that the white vertices form a partition of bigger or equal size, compared to set of black vertices. Each white vertex has at most two white neighbors, so it has at least two black neighbors, at most one edge to the black neighbor corresponds to the path labeled $(0,\lambda-q,iq,\lambda)$ and the remaining edges (at least one) correspond to the scheme $(0,p+iq,q,\lambda)$. But the number of paths labeled $(*,*,q,\lambda)$, cannot be greater then number of black vertices, so the number of white vertices is exactly same as the number of black ones. Hence, the given $\lambda_{(p,q)}$-labeling of G' gives a proper $BW(4)$-coloring.

On the other hand $BW(4)$-coloring divides the edges of the graph G into the sets of white cycles, black cycles and cycles with alternating black and white label. By using patterns $(0,p,\lambda,0)$, $(\lambda,0,\lambda-p,\lambda)$ and $(0,\lambda-2q,q,\lambda)(\lambda,2q,\lambda-q,0)$ consecutively on these cycles followed by completing colors of added leaves we obtain a proper $\lambda_{(p,q)}$-labeling of the modified graph G'. □

The case analysis for the labeling of Fig. 5

Here we describe the case study for the theorem 13. We show that under assumption $2q < p$ only schemes $(0, p, \lambda, 0)$, $(0, p+iq, q, \lambda)$, $(0, \lambda-q, iq, \lambda)$, $(0, \lambda, p, 0)$ and their mirrors are applicable to the path (z, a, b, x) on the graph H.

Let us investigate properties of the graph H in Figure. 5.

Sets A and B have the size $|A| = |B| = k - \lceil \frac{p}{q} \rceil$.

If $k = 0$ we obtain exactly result of the theorem 12 so no case analysis is needed.

The vertices z, x, y, y' and y'' have full degree, so they can be labeled by 0 or λ. W.l.o.g we suppose that z is labeled by 0. It implies that $c(a) \in \{p, p+q, p+2q, ..., \lambda - q, \lambda\}$.

If the vertices y and y' have different labels there is no way to label the vertex y'', so $c(y) = c(y')$. If the part of H over the dashed line is pre-labeled with respect to above mentioned condition, the labeling is always extendible to the bottom part of H.

We divide the proof into several cases.

Case A: $c(z) = 0, c(a) = p$ Then labels of the set $A \cup \{b\}$ could be chosen from the interval $[2p, \lambda]$ which contains at most $|A| + 1$ numbers at distance q. Maximum label of A is greater than $\lambda - 2q > \lambda - p \Rightarrow c(y) = c(y') = 0$. If the label of b belongs to the interval $[2p, \lambda - q]$ then the size of the set of possible labels of B is $\lfloor \frac{\lambda - 2p - q - 1}{q} \rfloor + 1 < |B|$. So $c(b) \in [\lambda - q + 1, \lambda] \Rightarrow c(x) = 0 \Rightarrow c(b) = \lambda$.

Case B: $c(z) = 0, c(a) = p + iq, 1 \le i < k, c(y) = 0$ If $1 < i < k - 1$ then in the intervals $[p, iq] \cup [2p + iq, \lambda]$ there are not enough numbers at distance q to label the set A.

Subcase Ba: $i = 1 \Rightarrow c(b) = q \Rightarrow c(x) = \lambda$. The set B can be easily labeled from the interval $[p + 2q, \lambda - q]$.

Subcase Bb: $i = k - 1$ then all possible labels at distance q from the interval $[p, (k-1)q]$ are used in the set A. It gives that $c(b) < p \Rightarrow c(x) = \lambda$ and $c(b)$ is multiple of q. The labels of the set B are chosen from the interval $[c(b)+p, \lambda-2q]$ which has sufficient size if $c(b) \le (\lfloor \frac{p}{q} \rfloor - 1)q$.

Case C: $c(z) = 0, c(a) = p + iq, 1 \le i < k, c(y) = \lambda$ For b we have $c(b) \ge q$ and labels of B belong to the interval $[q + p, \lambda - p]$. In the case that $c(b) \ge 2q$ there are not enough points to label the set B. Thus $c(b) < 2q < p \Rightarrow c(x) = \lambda \Rightarrow c(b) = q$. The rest of this case should be reduced to the mirror of the case B. (By interchanging the roles of z, a, A and x, b, B.)

Case D: $c(z) = 0, c(a) = \lambda \Rightarrow c(x) = 0$ The only solution is described in the case A.

Proof of Theorem 14

Theorem: *The $L(G; p, 1) \leq \lambda$ is an NP-complete problem for $p + 5 \leq \lambda < 2p$ and polynomially solvable for $\lambda \leq p + 2$.*

Proof. We prove the polynomial and NP-complete parts separately.

The NP-complete cases We are proving that $L(G; p, q) \leq \lambda$ is NP complete for $p + 5 \leq \lambda < 2p$. Note that in these cases the input graph is bipartite and (assuming the input graph is connected) labels in one class of the bipartition are less than or equal to $\lambda - p$, and labels in the other class greater than or equal to p.

We reduce the 3-colorability of a $(\lambda - p - 1)$-regular $(\lambda - p - 1)$-edge colored graph to our $L(p, 1) \leq \lambda$-problem and then prove that this special coloring problem is also NP-complete.

Let G be the graph which is to be 3-colored. Subdivide each edge by one new extra vertex and ask for a $\lambda_{(p,1)}$-labeling of this graph G'.

The graph G' is bipartite and the newly introduced vertices form one class of the bipartition. W.l.o.g we assume that labels from the interval $[p, \lambda]$ in this class are used. We claim that the original vertex has label at most 2, since all its neighbors have distinct labels and one of them should be less than or equal to $p + 2$. Using these labels as colors for the original graph we obtain a proper coloring, since adjacent vertices in G are at distance two in G' and thus have different labels.

In the opposite way a vertex 3-coloring of G together with an edge $(\lambda - p - 1)$-coloring gives us a proper $\lambda_{(p,1)}$-labeling of G' by using the same label from the interval $[p + 2, \lambda]$ for the vertices added to edges with the same color.

The NP-completeness of the special 3-colorability We reduce the 3-SAT problem where each variable has at most two positive and at most two negative occurences.

We give a modification of the classical reduction from 3-SAT. Let F be the given formula with n variables and m clauses. Construct the graph G as follows: For each variable x_i put into G vertices x_i and $\bar{x}_i$ and join them by an edge. Construct a gadget as on the Fig. 6 a) with $2n + 1$ vertices u and join vertices u_{2i-1} and u_{2i} to vertices x_i and $\bar{x}_i$. (This step replaces the vertex of high degree in the original proof.)

For each clause $(l \wedge l' \wedge l'')$ of F add into G the "clause" gadget depicted in the Fig. 6 b) where edges e_0, e_1 and e_2 lead to the vertices corresponding to the literals l, l' and l''.

Construct a gadget with $m + 1$ vertices u' and identify m of them with the lowest vertices of "clause" gadgets. Finally connect vertices u_{2n+1} and u'_{m+1} by an edge.

Assume that G is 3-colored. All vertices u have the same color hence on vertices x_i and $\bar{x}_i$ both of two remaining colors are used. All vertices u' have also the same color different to the color of vertices u. Let this color be viewed

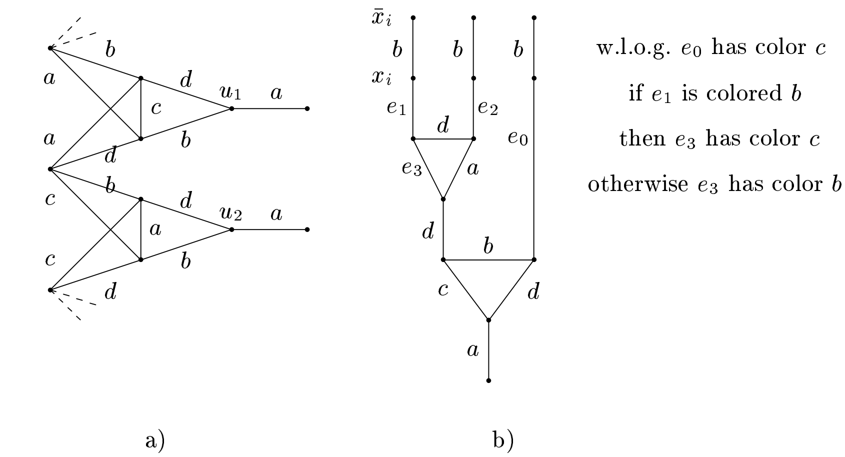

Fig. 6. 4-edge-colorability for 3-SAT

as the "false" assignment. The "clause" gadgets guarantee that in each clause at least one literal has "truth" assignment, hence if a 3-coloring exists, the formula F has an satisfying assignment. In vice versa any satisfying assignment can be extended to a 3-coloring of G.

As depicted on the Figures 6 a) and b) the graph G is 4-edge colorable and contains vertices of degree 3 or 4. To have the four-regular graph use two copies of G and join corresponding vertices of degree 3 by a matching. Now the graph is four-regular, 4-edge-colorable and any coloring of one copy can be extended to a coloring of whole graph by cyclic exchange of colors on the other copy.

Finally we prove by induction the NP-completeness of the 3-coloring of a k-regular k-edge-colorable graph for $k \geq 5$. Two copies of a $(k-1)$-regular $(k-1)$-edge-colorable graph joined by the matching gives us the hardness result similarly as in the above case.

The polynomially solvable cases In the case that a $\lambda_{(p,1)}$-labeling of G exists and $\lambda < p$ then the input graph G has no edges. Similarly concerning a $p_{(p,1)}$-labeling and a $(p+1)_{(p,1)}$-labeling respectively the input graph contains the disjoint set of paths of length at most 3.

Only subgraphs of a (infinite) path $(v_0, v_1, \ldots)$ with leaves added to vertices v_{4k} and v_{4k+1} allows a $(p+2)_{(p,1)}$-labeling. Such a graph is recognizable in the linear time. □