Journal manuscript No.
(will be inserted by the editor)

Computation of Optimum Distance Constrained Labelings

Veronika Halász · Gerhard Reinelt · Ekaterina Tikhoncheva

Version of 06.07.2015

Abstract There are many problems in graph theory concerned with labeling the vertices of a graph subject to certain constraints and opimization criteria. In this paper we study so-called distance constrained labeling problems where, depending on the distance between two vertices, lower bounds on the difference between their (integer or continuous) labels have to observed. We discuss integer programming approaches for finding labelings with minimum spans and report about computational experiments.

Keywords Graphs · Labeling

1 Introduction

Many problems in graph theory deal with the assignment of colors or labels to the vertices of a given graph satisfying certain constraints and optimizing some objective function. Prominent examples are *coloring problems* like the determination of the minimum number of colors necessary such that two adjacent vertices have different colors or *labeling problems* where labels correspond to integer numbers and where the difference between labels plays a role.

A special type of labeling problems, so-called *distance constrained labeling problems*, have been introduced by Hale in [**?**]. They are motivated by the problem of assigning frequencies to radio channels in order to avoid interference. In Hale's approach channels (frequencies) are represented by nonnegative integers and have to be assigned to transmitters. Close transmitters should receive different frequencies and very close transmitters should receive frequencies further apart. The bandwidth of an assignment is the maximum absolute difference between frequencies and should be as small as possible.

The problem can be modeled as a graph labeling problem. The vertices of the graph correspond to transmitters and the vertex labels to frequencies. Depending on the definition of the distance between two vertices several models are possible. At first we consider the classical version where the distance between two vertices is the smallest number of edges in a path connecting them.

Let $G = (V, E)$ denote the (undirected) graph with vertex set V, $|V| = n$, and edge set E. In the most intensively studied model in this context, nonnegative integer labels have to be assigned to the vertices of G, such that the labels of adjacent vertices must differ by at least 2 and vertices having common neighbors must get distinct labels. This so-called $L(2,1)-$*labeling problem* was first investigated by Griggs and Yeh [**?**]. The smallest possible value of the largest label in such a labeling is denoted by $\lambda_{2,1}(G)$ (for example: see Fig. **??**).

More generally, an $L(j_1, j_2)-$labeling of G is an assignment of nonnegative integer labels to the vertices of G in such a way that the labels of vertices at distance 2 apart differ at least by j_2

Institut für Informatik, Universität Heidelberg
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

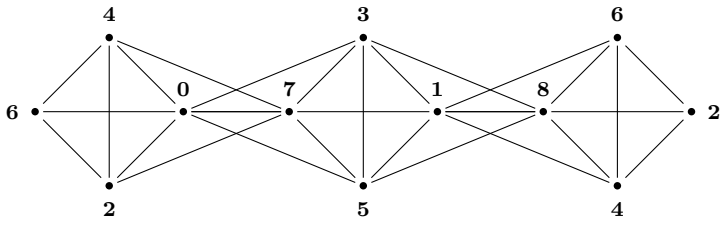


Fig. 1 Example of the $L(2,1)-$labeling problem with $\lambda_{2,1}(G) = 8$.

and labels of adjacent vertices differ by j_1 or more. A comprehensive survey of $L(j_1, j_2)-$labeling and bounds on the minimum $\lambda_{j_1,j_2}(G)$ of the largest label, with nearly 200 references, was given by Calamoneri in [**?**].

As regards vertices at larger distances apart, $L(3,2,1)-$labeling (first studied by Shao and Liu in [**?**]) and more generally $L(j_1, j_2, j_3)-$labeling (introduced by Shao in [**?**] put three conditions, depending on the distances between vertices. Here the parameter j_i describes that the difference between the labels of vertices at distance i apart must be at least j_i. The properties of $L(3,2,1)-$labelings have been analyzed further in [**?**].

The problem can be extended in a natural way to longer distances. Let $j_1, j_2, \ldots, j_s$ be non-negative integers such that $j_1 \geq j_2 \geq \ldots \geq j_s$. An $L(j_1, j_2, \ldots, j_s)-$*labeling* of G is an assignment $\varphi : V \to \{0, 1, 2, \ldots\}$ such that for every pair u and v of vertices the inequality $|\varphi(u) - \varphi(v)| \geq j_i$ holds if the distance between u and v in G is equal to i, for some $i = 1, 2, \ldots, s$. Note that there are no constraints for u and v if their distance is greater than j_s and that with this definition an $L(j_1, j_2)-$labeling can be considered as $L(j_1, j_2, 0, 0, \ldots)-$labeling.

The *span* of an $L(j_1, j_2, \ldots, j_s)-$labeling φ is the largest label assigned by φ to the vertices.

In analogy to $\lambda_{2,1}$ we define $\lambda_{j_1,j_2,\ldots,j_s} = \min_\varphi \max_{v \in V} \varphi(v)$ as the smallest possible span taken over all $L(j_1, j_2, \ldots, j_s)-$labelings φ. This general labeling is less studied but there are some interesting results for a special case, namely for the radio-labeling in [**?**,**?**,**?**,**?**]. A brief summary of the basic results with the related references is given in subsection 7.4 of the survey [G].

This paper deals with the computation of optimum distance constrained labelings. In section **??** we discuss models for the classical problem as well as for variants closer to real frequency assignment problems. Section 3 describes the graphs on which the tests were carried out. We present our first computational experiments in section 4 and the improvements in section 5. In the end of the paper we may draw some conclusions about the results.

1.1 Computational complexity

It was already known from [**?**] that the decision version of the simplest labeling $L(2,1)$ (that is, the input consists of a graph G and an integer k, and the question is whether $\lambda_{2,1}(G) \leq k$ holds) on unrestricted input graphs is NP-complete.

Fiala, Golovach and Kratochvil [**?**] proved that the decision version of the $L(j_1, j_2)-$labeling problem is NP-complete on trees whenever j_1 is not a multiple of j_2. Otherwise it is reducible to $L(\frac{j_1}{j_2}, 1)$, which is solvable in polynomial time [**?**] using the modificated algorithm of Chang and Kuo [**?**]. This algorithm works not only on trees but also on the slightly wider class of graphs which can be transformed to a tree by removing at most p edges for some fixed p.

The $L(2,1)-$labeling problem still remains NP-hard for planar graphs, bipartite graphs, split graphs and chordal graphs [**?**], even for graphs of treewidth 2 [**?**]. In the case of planar graphs, determining the existence of a $k - L(2,1)-$labeling is NP-hard for $k \geq 4$, while it can be done in polynomial time for $k \leq 3$ [**?**]. A $k - L(2,1)-$labeling of a graph G is a proper $L(2,1)$-labeling with span k.

2 Models for computing distance constrained labelings

We start with an integer programming model for the classical problem. Let the label of vertex v be represented by the integer variable $c(v)$ and let $\mathrm{dist}(u,v)$ denote the distance between two vertices u and v.

The problem of finding an optimum $L(j_1, j_2, \ldots, j_s)-$labeling can be stated as follows.

$$\begin{aligned}
&\min L \\
L - c(v) &\geq 0 && \forall\, v \in V && (1)\\
|c(v) - c(u)| &\geq j_{\mathrm{dist}(u,v)} && \forall\, u, v \in V \text{ with } \mathrm{dist}(u,v) \leq j_s && (2)\\
c(v) &\geq 0 && \forall\, v \in V && (3)\\
c(v) &\text{ integer} && \forall\, v \in V. && (4)
\end{aligned}$$

Inequalities (**??**) model the min-max objective function and so $\min \lambda_{j_1,j_2,\ldots,j_s}(G)$ is computed. The distance constraints (**??**) for vertex pairs u and v can be linearized in the standard way with additional binary variables z_{uv} as

$$\begin{aligned}
c(v) - c(u) + M \cdot z_{uv} &\geq j_{\mathrm{dist}(u,v)} && \forall\, u, v \in V \text{ with } \mathrm{dist}(u,v) \leq j_s\\
c(u) - c(v) + M \cdot (1 - z_{uv}) &\geq j_{\mathrm{dist}(u,v)} && \forall\, u, v \in V \text{ with } \mathrm{dist}(u,v) \leq j_s\\
z_{uv} &\in \{0,1\} && \forall\, u, v \in V.
\end{aligned}$$

The number M has to be chosen big enough. Obviously $M \geq j_1 + \lambda_{j_1,j_2,\ldots,j_s}(G)$ has to hold. If no good upper bound on $\lambda_{j_1,j_2,\ldots,j_s}(G)$ is known, we just set $M = n \cdot j_1$.

The distance labeling problem introduced so far is an interesting graph theoretical problem. However, it cannot model a practical frequency assignment problem properly because the discrete graph distances do not reflect the real distances between transmitters. Hence it can only serve as a coarse approximation.

A possibility for making the model more practical would be to insert artificial vertices to increase the distance between vertices closer to reality. This way we preserve the discrete nature of the model, but the graph size is increased.

One could also allow real values as labels (leaving the graph and the distance unchanged). Since proper integer solutions remain feasible, the minimum span is not bigger than in the classical model. Actually, this is the LP relaxation of the model above. We even can prove that the span is the same in the cases of integer and real labels.

Theorem If the required differences $j_1, j_2, \ldots, j_s$ are each integers, then the labeling number is an integer and there exists an optimal labeling with integer labels.

Proof Let an optimal labeling φ be given. Take the floor of each labels. In this way we get the integer labeling φ_i. Obviously, the biggest one of the new labels is not greater than before. In the following we prove that the requirements are met for each vertex pairs also in φ_i.

Let u and v be arbitrary vertices of G, where without loss of generality we may assume that $\varphi(u) \leq \varphi(v)$. Then $\varphi(v) - \varphi(u) \geq d_{dist(u,v)}$ since φ is a proper labeling.

By definition $\varphi_i(v) = \lfloor \varphi(v) \rfloor = \varphi(v) - [\varphi(v)]$, where $[\varphi(v)]$ defines the ceiling function of $\varphi(v)$. Similar holds for u, as well. So, $\varphi_i(v) + [\varphi(v)] - \varphi_i(u) - [\varphi(u)] \geq d_{dist(u,v)}$, where $d_{dist(u,v)} \in \mathbb{N}$. It means that $\varphi_i(v) - \varphi_i(u) + ([\varphi(v)] - [\varphi(u)]) \geq d_{dist(u,v)} \Rightarrow \varphi_i(v) - \varphi_i(u) \geq d_{dist(u,v)} - ([\varphi(v)] - [\varphi(u)])$.

The greatest possible value of $([\varphi(v)]-[\varphi(u)])$ is $(1-\varepsilon)-0 = 1-\varepsilon < 1$, the smallest possible one is $\varepsilon-1 > -1$. Since $\varphi_i(v)-\varphi_i(u)$ is an integer, it is at least as large as $\lceil d_{dist(u,v)} - ([\varphi(v)] - [\varphi(u)]) \rceil$. $d_{dist(u,v)} \in \mathbb{N}$ and $-1 < ([\varphi(v)] - [\varphi(u)]) < 1$, hence $\lceil d_{dist(u,v)} - ([\varphi(v)] - [\varphi(u)]) \rceil = d_{dist(u,v)}$ $\Rightarrow \varphi_i(v) - \varphi_i(u) \geq d_{dist(u,v)}$. The same argument holds for each vertex pair, so φ_i is a proper labeling with span not bigger than that of φ. Because of the optimality of φ, φ_i is an optimal labeling using only integers.

Some results about labeling with real numbers are in a paper by Griggs and Jin [**?**]. It would be most precise to consider the complete graph on the transmitters and to take the Euclidean distance between transmitters as edge weights. Since the graph is finite there is also only a finite number, say t, of occurring distances, and constraints can be formulated as above relative to a sequence $L(j_{dist_1}, j_{dist_2}, \ldots, j_{dist_t})$.

For the *Euclidean model* we can basically adopt the integer model with two little changes. First, the variables $c(v)$ are now continuous. Second, the Euclidean distances have to be transformed to suitable right hand side values for the inequalities (**??**). This will be accomplished by a function $\ell(\mathrm{dist}(u,v))$. The principle formulation of the Euclidean approach is the following.

$$
\begin{aligned}
\min L & & \\
L - c(v) &\geq 0 & \forall\, v \in V \\
c(v) - c(u) + M \cdot z_{uv} &\geq \ell(\mathrm{dist}(u,v)) & \forall\, u, v \in V \text{ with } \mathrm{dist}(u,v) \leq j_s \\
c(u) - c(v) + M \cdot (1 - z_{uv}) &\geq \ell(\mathrm{dist}(u,v)) & \forall\, u, v \in V \text{ with } \mathrm{dist}(u,v) \leq j_s \\
c(v) &\geq 0 & \forall\, v \in V \\
z_{uv} &\in \{0,1\} & \forall\, u, v \in V.
\end{aligned}
$$

3 Test instances

We are not aware of any research on solving distance constrained labeling problems to optimality according to the Euclidean distances of the transmitters, so we generated own benchmark problems. The goal was to test the graph model and the Euclidean model on instances which somehow resemble the practical situation for real frequency assigment problems.

In radio and mobile networks large areas are usually covered by polygons which together form lattices. In practice three lattices play a prominent role: hexagonal, triangular and square lattices (see Fig.**??**).

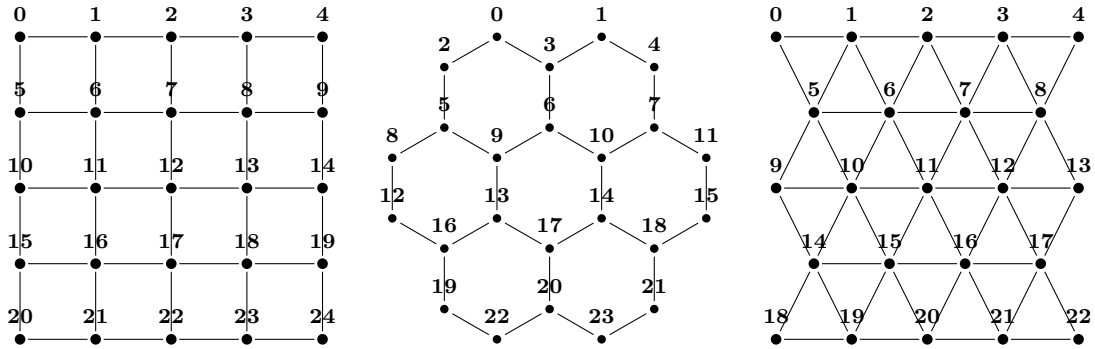


Fig. 2 Examples of lattice graphs

Maybe the hexagonal covering is the most common one. In this case the transmitters are assumed to at the centers of the hexagons and two transmitters are adjacent in the graph if and only if the corresponding hexagons share a common edge. The graph constructed this way is a triangular lattice. Some research has been done for the classical frequency assignment problem on these graphs, see [**?**,**?**,**?**,**?**,**?**].

In our experiments we performed computations with the Euclidean model and with the graph model for three problem types arising from lattice graphs. The three lattice types shown in Figure **??** were considered. For the graph version the classical graph distance was taken, (e.g., the

distance between 0 and 21 in the triangular lattice is 5). For the Euclidean version coordinates were given to the vertices such that every edge shown in Fig. **??** has length 1. So the coordinates for the vertices 0, 1 and 2 are $(-\frac{\sqrt{3}}{2}, +\frac{3}{2})$, $(+\frac{\sqrt{3}}{2}, +\frac{3}{2})$, $(-\sqrt{3}, +1)$ in the hexagonal lattice, $(-2, +2)$, $(-1, +2)$, $(0, +2)$ in the square lattice and $(-2, +2)$, $(-1, +2)$, $(0, +2)$ in the triangular lattice. E.g., the distance between 0 and 1 in the hexagonal lattice is $\sqrt{3}$.

The difference of the distance between two vertices in the Euclidean and in the graph model can be small or zero, but also relatively high. Two vertices with graph distance 4 could, for example, have Euclidean distance 4, $\sqrt{10}$ or $\sqrt{8}$.

With the help of the transformation function ℓ we tried to have a similar range of spans for the graph and the corresponding Euclidean model. Of course, ℓ has to be monotonically decreasing. We experimented with two variants for ℓ which in addition have the property that the values for integer distances are preserved. i.e. $\ell(\text{dist}(u, v)) = j_{\text{dist}(u,v)}$ for $\text{dist}(u, v)$ integer. E.g., this is satisfied by defining $\ell(\text{dist}(u, v)) = j + 1 - \text{dist}(u, v)$ for $L(j, j-1, \ldots, 1)-$labelings. Since the graph distances are not smaller than the Euclidean distances, the spans for the graph problems will be higher in general.

4 First computational experiments

For our computations we consider the square lattice on 25 vertices, the hexagonal lattice on 24 vertices and the triangular lattice on 23 vertices (as depicted in Fig. **??**). The minimum spans have been computed with ILOG CPLEX Version 12.4.1 [**?**].

All computations were done on a computer with 4 processors Intel Core $i7-2600$, 3.40 GHz, and 16GB RAM. The implementation is in $C++$. We should mention here, that our implementation is not necessary efficient and an optimized version could perform better.

4.1 Classical model

As a first step we want to test, problems of what size is it possible to solve in reasonable time using the classical model (**??**)-(**??**). For this, we tested the model on the three types of lattices with different number of nodes. The running times (in min:sec) and obtained optimal solutions in the different cases are presented in Table **??**.

One can see, that already for the triangular lattice with 30 nodes computation of an optimal $L(3, 2, 1)-$labelling took hours. The computation of an optimal $L(4, 3, 2, 1)-$labeling exhausted all computer resources and was cancelled by the processor.

Lattice	$L(2,1)$		$L(3,2,1)$		$L(4,3,2,1)$	
	Span	CPU	Span	CPU	Span	CPU
Hexagonal (24 nodes)	5	0.2	9	1.0	19	47.1
Hexagonal (30 nodes)	5	0.3	9	1.9	20	8:11.4
Triangular (23 nodes)	8	0.9	18	9:33.5	32	15575:50
Triangular (30 nodes)	8	1.7	18	163:50.7	-	-
Square (25 nodes)	6	0.6	11	1.72	25	143:39.05
Square (30 nodes)	6	1.1	11	2.3	25	335:24.8

Table 1 Spans and running time using the classical model (**??**)-(**??**)

We also wanted to know, if the different values of j_1, j_2 in the $L(j_1, j_2)-$labeling have an influence on the running time. It was the case comparing $L(3, 2)-$labeling with $L(2, 1)-$labeling: the computation of $\lambda_{3,2}$ took longer for almost all considered graphs (see Table **??**).

Lattice	$L(2,1)$		$L(3,2)$	
	Span	CPU	Span	CPU
Hexagonal (24 nodes)	5	0.2	9	0.42
Hexagonal (30 nodes)	5	0.3	9	1.04
Triangular (23 nodes)	8	0.9	16	18
Triangular (30 nodes)	8	2.6	16	19.23
Square (25 nodes)	6	0.6	11	2.24
Square (30 nodes)	6	1.1	11	1.3

Table 2 $L(3,2)$ vs $L(2,1)$

4.2 Linear transformation

We first considered $L(2,1)$- and $L(3,2,1)$-labelings for the classical model (**??**)-(**??**) and used functions $\ell_2(\mathrm{dist}(u,v)) = 3-\mathrm{dist}(u,v)$ and $\ell_3(\mathrm{dist}(u,v)) = 4-\mathrm{dist}(u,v)$ in the respective Euclidean model. The distance of adjacent vertices is 1 in both cases, the distance beween non-adjacent vertices can be the same as their graph distance, but also considerably smaller. Because of this, the constraints for most vertex pairs are stronger in the Euclidean model.
Table **??** gives the spans and running times as obtained with Cplex. The spans between the models differ as expected. Surprisingly, the running times for the Euclidean model are considerably higher.

Lattice	$L(2,1)$		Euclidean		$L(3,2,1)$		Euclidean	
	Span	CPU	Span	CPU	Span	CPU	Span	CPU
Hexagonal	5	0.2	6.42	4.6	9	0.9	16.62	92:57.7
Triangular	8	0.9	9.78	38.6	18	9:33.5	21.91	6149:18.0
Square	6	0.6	8.64	15.3	11	1.7	19.91	2559:47

Table 3 Spans and running times for the linear function

4.3 Stepwise transformation

With this type of function we want to map the Euclidean distances to integer values, again preserving the value if the distance is integer already.

Table **??** shows the possible Euclidean distances for the three lattices and the corresponding graph distances.

Hexagonal		Triangular		Square	
Euclidean	Graph	Euclidean	Graph	Euclidean	Graph
1	1	1	1	1	1
$\sqrt{3}$	2	$2, \sqrt{3}$	2	$2, \sqrt{2}$	2
$2, \sqrt{7}$	3	$3, \sqrt{7}$	3	$3, \sqrt{5}$	3
$3, 2\sqrt{3}$	4	$4, \sqrt{13}, 2\sqrt{3}$	4	$4, \sqrt{10}, 2\sqrt{2}$	4
$4, \sqrt{13}, \sqrt{19}$	5	$\sqrt{19}, \sqrt{21}$	5	$\sqrt{13}, \sqrt{17}$	5
$\sqrt{21}, 3\sqrt{3}$	6	$2\sqrt{7}$	6	$2\sqrt{5}, 3\sqrt{2}$	6
$5, 2\sqrt{7}$	7			5	7
				$4\sqrt{2}$	8

Table 4 Euclidean and graph distances for the three lattices

For the hexagonal lattice here are some vertex pairs with Euclidean distance $3\sqrt{3}$ and graph distance 6, and some with Euclidean distance 5 and graph distance 7. For these pairs the graph distance is longer although the Euclidean is shorter. For the square lattice there are for example pairs with Euclidean distance $2\sqrt{2}$ and graph distance 4, and some with Euclidean distance 3 and graph distance 3. For the triangular lattice there are no such exceptions.

Let $f(x)$ denote the graph distance corresponding to the Euclidean distance (in the respective lattice). For $L(2,1)$- and $L(3,2,1)$-labelings only the graph distances 1, 2 and 3 have to be taken into account. For the respective pairs we set $\ell_2(\mathrm{dist}(u,v)) = 3 - f(\mathrm{dist}(u,v)))$ for $L(2,1)$-labelings and $\ell_3(\mathrm{dist}(u,v)) = 4 - f(\mathrm{dist}(u,v)))$ for $L(3,2,1)$-labelings.

Table **??** shows spans and running times for these step functions. Although the constraints are the same for each vertex pair in this case, the calculations are still much slower in the Euclidean model. However, compared with the previous section (see table **??**) they have improved a lot.

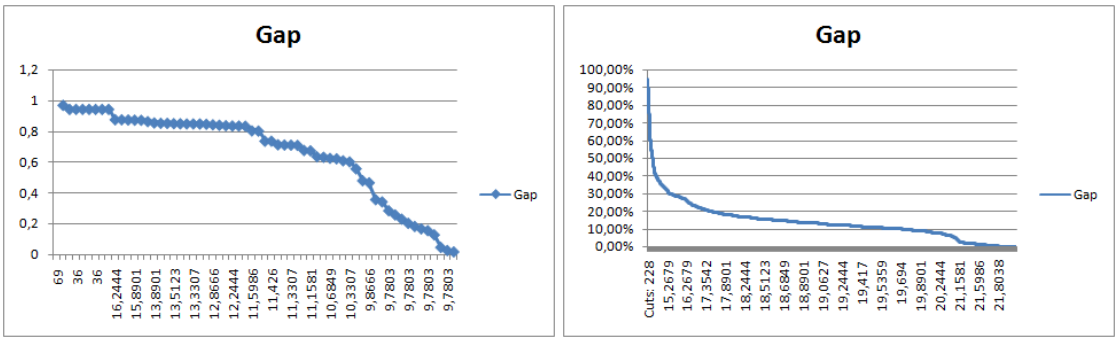
Lattice	$L(2,1)$		Euclidean		$L(3,2,1)$		Euclidean	
	Span	CPU	Span	CPU	Span	CPU	Span	CPU
Hexagonal	5	0.2	5	2.2	9	0.9	9	3.8
Triangular	8	0.9	8	27.3	18	9:33.5	18	76:39.1
Square	6	0.	6	23.0	11	1.7	11	10.2

Table 5 Spans and running times for the step function

5 Model improvements

As one can see from the previous section, computing an optimal $L(2,1)-$ and an $L(3,2,1)-$labeling for considered lattice graphs take much time. One of the reasons for not satisfactory performance could lie on the bad evolutions of the lower bounds in the optimization process (see fig.**??**). In this section we discus our attempts to improve the performance.

The computation times are not really satisfactory, so we examined several possibilities for improving the model. They will be discussed in this section. All computations have been performed for the linear transformation function.



(a) $L(2,1)$ Labeling, optimal value 8.78029 (b) $L(3,2,1)$ Labeling, optimal value 21.9068

Fig. 3 Evolution of the LB in case of triangular lattice graph and real number labeling

5.1 Reducing the value of M

It is well-known that whenever a linear model contains a so-called *big M* it is usually advantageous to find the smallest possible value for M. A speed-up of the computations can be expected. However, this is not guaranteed and the effect can also be converse.

Since the optimum values from the classical model are available, it is easy to get good M values for the Euclidean model. In the case $\ell_2(dist) = 3 - dist$ we set $M_2^* = 2 \cdot \lambda_{L(2,1)}$, while $M_3^* = 3 \cdot \lambda_{L(3,2,1)}$ in the case $\ell_3(dist) = 4 - dist$.

Table **??** gives the running times with those of Table **??**. The new CPU times as well as the percentage of these times compared to those of Table **??** are given.

Lattice	$L(2,1)$		$L(3,2,1)$	
	CPU	%	CPU	%
Hexagonal	5.2	113	104:43.5	112.7
Triangular	30.3	**78.5**	3519:4.4	**57.2**
Square	23.7	155.0	1416:30.2	**55.3**

Table 6 Effect of new setting of M

One can see that the modification works fairly well for the triangular lattice, but rather poorly for the hexagonal lattice. For the square lattice we have mixed results.

5.2 Strengthening the model

The inequalities in the above models are straightforward and we are interested in strengthening them. Let G' be a node induced subgraph of G. If we consider feasible labelings for G' then any constraint on the labels for G' is valid for all subgraphs of G isomorphic to G'. In the following we have chosen the smallest possible sum of the labels for G' as constraint, i.e., if $s(G)$ is the smallest sum of feasible labels for G' then the inequality $\sum_{v \in V(H)} c(v) \geq s(G)$ is valid for all subgraphs H of G isomorphic to G'. We have experimented with several types of subgraphs.

A *star* is a graph $G = (V, E)$ such that $E = \{vw \mid w \in V \setminus \{v\}\}$ for some $v \in V$ (center vertex of the star) and suppose $|V| = m$. The sum of labels for a star is a lower bound for the sum of labels of every subgraph containing a star. Consider an $L(j_1, j_2, \ldots)$-labeling. For the star m labels have to be chosen such that the gap between the label of the central vertex and the label of any other vertex is at least j_1, and the gap between the labels of any two non-central vertices is at least j_2. We distinguish three cases.

1. The label of the center is the smallest one.
   Then the labels are 0, j_1, $j_1 + j_2$, $j_1 + 2j_2$, $j_1 + 3j_2, \ldots$, $j_1 + (m-2)j_2$ summing up to $(m-1) \cdot j_1 + \sum_{i=1}^{m-2} i \cdot j_2$.
2. The label of the center is the greatest one.
   Then the labels are 0, j_2, $2j_2, \ldots$, $(m-2)j_2$, $(m-2)j_2 + j_1$ with sum $(\sum_{i=1}^{m-2} i \cdot j_2) + (m-2)j_2 + j_1$.
3. The central label is the $(k+1)$st greatest label for some $k \geq 1$.
   In this case the labels 0, j_2, $2j_2, \ldots$, $(k-1)j_2$, $(k-1)j_2 + j_1$, $(k-1)j_2 + 2j_1$, $k \cdot j_2 + 2j_1, \ldots$, $(m-3)j_2 + 2j_1$ and their sum is $(\sum_{i=1}^{m-3} i \cdot j_2) + 2(k-1)j_2 + (2(m-k-1)+1) \cdot j_1$.

Since $j_1 \geq j_2$, an easy calculation shows that the smallest sum is obtained in case 2 and is equal to $\frac{1}{2}(m+1)(m-2) \cdot j_2 + j_1$. So for every vertex $u \in V$ we can add its associated *star inequality*

$$c(u) + \sum_{i \in N(u)} c(i) \geq \frac{(\deg(u)+2)(\deg(u)-1)}{2} \cdot j_2 + j_1$$

to the models (where $N(u)$ denotes the set of neighbors of u and $\deg(u)$ is the degree of u.

A second possibility is to associate inequalities with small sublattices of the lattices we considered in our computational experiments. We considered the triangles with side length 1, 2 and 3, the hexagons with side length 1 and 2, trapezes as half of a hexagon, squares with side length 1, 2 and 3, and small square lattices with 4 and 9 vertices. Table **??** gives the smallest sums of labels for these subgraphs for the labelings $L(2,1)$ and $L(3,2,1)$.

Sublattice	L(2,1)	L(3,2,1)
Hexagon with side length 1	16.61	31.61
Hexagon with side length 2	3.00	10.82
Trapeze with side length 1	7.0718	13.07
Triangle with side length 1	6.00	9.00
Triangle with side length 2	3.00	6.00
Triangle with side length 3	0.00	3.00
Square 1×1 (4 vertices)	10.34	16.34
Square 2×2 (4 vertices)	2.69	8.69
Square 3×3 (4 vertices)	0.00	2.00
Square lattice 2×2 (9 vertices)	29.19	59.40
Square lattice 1×2 (6 vertices)	16.23	30.28

Table 7 Minimum label sums for sublattices

We examined the effect of the addition of these small subgraph inequalities. The results are presented in Table **??** for the hexagonal, in Table **??** for the triangular, and in Table **??** for the square lattice. For the square lattice we only added non-overlapping squares which is only a small subset of possible squares.

Hexagonal lattice	L(2,1)	L(3,2,1)
Without any subgraph-inequality	4.6	92:57.7
Hexagons with side length 1	4.3	265:52.0
Hexagons with side length 1 and 2	4.4	148:0
Trapezes	6.8	118:24.6

Table 8 Effect of subgraph inequalities for the hexagonal lattice

Triangular lattice	L(2,1)	L(3,2,1)
Without any subgraph-inequality	38.6	6149:18.0
Triangles with side length 1	21.0	4914:18.8
Triangles with side length 1,2 and 3	37.2	9059:15.6
Hexagons with side length 1	36.6	2519:20.3

Table 9 Effect of subgraph inequalities for the triangular lattice

For the hexagonal lattice we observe a running time improvement of about 10-20%. In the triangular case, time reduces to about 50% in two cases for the $L(3,2,1)$-labeling. Also in one case for the square lattice a considerable improvement is achieved.

Square lattice	L(2,1)	L(3,2,1)
Without any subgraph-inequality	15.3	2559:47
Squares with side length 1	24.3	3046:6.4
Squares with side length 1,2 and 3	24.6	4424:14.7
Rectangle with side length 1×2	33.2	*killed*

Table 10 Effect of subgraph inequalities for the square lattice

6 Conclusions

We tried some further heuristics like numbering the vertices randomly in order to see if this makes a difference to CPLEX. But no consistent improvement could be observed.

References

1. CPLEX Optimizer, `www.cplex.com`
2. K. Aardal, S.P.M. van Hoesel, A.M.C.A. Koster, C. Mannino, A. Sassano, Models and solution techniques for frequency assignment problems, Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 1 (4) (2003), pp. 261317
3. H.L. Bodlaender, T. Kloks, R.B. Tan, J. van Leeuwen. Approximations for $\lambda-$coloring of graphs. The Computer Journal 47, 193-204 (2004)
4. R. Borndoerfer, et al. "Frequency assignment in cellular phone networks." Annals of Operations Research 76 (1998): 73-93.
5. T. Calamoneri: The $L(h,k)-$labelling problem: An updated survey and annotated bibliography. Comput. J. 54 (2011), 1344-1371. (A later version is available online at http://wwwusers.di.uniroma1.it/~calamo/PDF-FILES/survey.pdf)
6. G.J. Chang, W.T. Ke, D. Kuo, D.D.F. Liu, R.K. Yeh. On L(d,1)-labelings of graphs. Discrete Math., 220 (2000), pp. 57-66
7. G.J. Chang, D. Kuo. The L(2,1)-labeling on graphs. SIAM J. Discrete Math., 9 (1996), pp. 309-316
8. G. Chartrand, D. Erwin, F. Harary and P. Zhang, Radio Labelings of graphs, Bull. Inst. Combin. Appl. 33 (2001), 77-85.
9. M.L. Chia, D. Kuo, H.Y. Liao, C.H. Yang and R.K. Yeh, $L(3,2,1)-$labeling of graphs, Taiwanese J. Math. 15 (2011), 2439-2457.
10. F.Y.L. Chin, Y. Zhang, H. Zhu, A 1-local 13/9-competitive algorithm for multicoloring hexagonal graphs, in: Proc. 13th Annual International Computing and Combinatorics Conf. COCOON), 2007, pp. 526536
11. N. Eggemann, F. Havet, S.D. Noble. k-L(2,1)-labelling for planar graphs is NP-complete for k$\geq$4. Disc. Appl. Math., 158 (16), 1777-1788 (2010)
12. J. Fiala, P.A. Golovach and J. Kratochvl. Computational complexity of the distance constrained labeling problem for trees, Proc. 35th ICALP , Part I, 294–305 (2008).
13. J. Fiala, P.A. Golovach, J. Kratochvl. Distance constrained labelings of graphs of bounded treewidth. Proc. 32th ICALP, 360-372 (2005)
14. J. A. Gallian, A dynamic survey of graph labeling, Electronic Journal of Combinatorics, (2014) #DS6, 308 pages.
15. J. R. Griggs, X.T. Jin. Real Number Graph Labellings with Distance Conditions. SIAM J. Discrete Math., 20 (2006) p.p. 302-327
16. J.R. Griggs, R.K. Yeh. Labeling graphs with a condition at distance two. SIAM J. Discrete Math., 5 (1992), pp. 586–595
17. V. Halasz, Zs. Tuza: Distance-constrained labeling of complete trees, Discrete Mathematics 338 (2015), pp. 1398-1406
18. W.K. Hale, Frequency assignment: Theory and application, Proc. IEEE, 68 (1980), 1497–1504.
19. X. Li, V. Mak and S. Zhou, Optimal radio labellingsof complete m- ary trees, Discrete Applied Math. 158 (2010), 507-515.
20. D. D.F. Liu, Radio number for trees, Discrete Math. 308 (2008), 1153-1164.
21. L. Narayanan, S.M. Shende, Static frequency assignment in cellular networks, Algorithmica, 29 (3) (2001), pp. 396409
22. S. Rajasekaran, K. Naik and D. Wei. "On Frequency Assignment in Cellular Networks." DIMACS Series in Discrete Mathematics and Theoretical Computer Science 52 (2000): 293-302.
23. Z. Shao, The $L(d_1,d_2,d_3)-$labeling on graphs, J. Nanjing Univ. Math. Biquart. 21 (2004), 234-238.
24. Z.D. Shao and J.Z. Liu, The $L(3,2,1)-$labeling problem on graphs, Math. Appl. 17 (2004), 596-602.