$L(p,1)$ ラベリングのための固定パラメータアルゴリズム *

川井一馬 †1, 土中哲秀 ‡2, 小野廣隆 §1

1 名古屋大学
2 中央大学

Abstract

Given a graph, an $L(p,1)$-labeling of the graph is an assignment f from the vertex set to the set of nonnegative integers such that for any pair of vertices (u,v), $|f(u)-f(v)| \geq p$ if u and v are adjacent, and $f(u) \neq f(v)$ if u and v are at distance 2. The $L(p,1)$-LABELING problem is to minimize the span of f (i.e.,$\max_{u \in V}(f(u)) - \min_{u \in V}(f(u)) + 1$). It is known to be NP-hard even for graphs of maximum degree 3 or graphs with tree-width 2, whereas it is fixed-parameter tractable with respect to vertex cover number. Since vertex cover number is a kind of the strongest parameter, there is a large gap between tractability and intractability from the viewpoint of parameterization. To fill up the gap, in this paper, we propose new fixed-parameter algorithms for $L(p,1)$-LABELING by the twin cover number plus the maximum clique size and by the tree-width plus the maximum degree. These algorithms reduce the gap in terms of several combinations of parameters.

1 Introduction

Let G be an undirected graph, and p and q be constant positive integers. An $L(p,q)$-labeling of a graph G is an assignment f from the vertex set $V(G)$ to the set of nonnegative integers such that $|f(x) - f(y)| \geq p$ if x and y are adjacent and $|f(x) - f(y)| \geq q$ if x and y are at distance 2, for all x and y in $V(G)$. We call the former *distance*-1 *condition* and the latter *distance*-2 *condition*. A k-$L(p,q)$-labeling is an $L(p,q)$-labeling $f : V(G) \to \{0, \dots, k\}$, where the labels start from 0 for conventional reasons. The k-$L(p,q)$-LABELING problem determines whether given G has a k-$L(p,q)$-labeling, or not, and the $L(p,q)$-LABELING problem asks the minimum k among all possible assignments. The minimum value k is called the $L(p,q)$-labeling number, and we denote it by $\lambda_{p,q}(G)$, or simply $\lambda_{p,q}$. Notice that we can use $k+1$ different labels when $\lambda_{p,q}(G) = k$.

The original notion of $L(p,q)$-labeling can be seen in the context of frequency assignment. Suppose that vertices in a graph represent wireless devices. The presence/absence of edges indicates the presence/absence of direct communication between the devices. If two devices are very close, that is, they are connected in the graph, they need to use sufficiently different frequencies, that is, their frequencies should be apart at least p. If two devices are not very but still close, that is, they are at distance 2 in the graph, their frequencies should be apart at least q ($\leq p$). Thus, the setting of $q = 1$ as one unit and $p \geq q = 1$ is considered natural and interesting, and the minimization of used range becomes the issue. Note that $L(1,1)$-labeling on G is equivalent to the ordinary coloring on the square of G. From these, $L(p,1)$-LABELING for $p > 1$ is intensively and extensively studied among several possible settings of p. In particular, $L(2,1)$-LABELING is considered the most important. A reason is that it is natural and suitable as a basic step to consider, and another reason is that the computational complexity (e.g., hardness or polynomial-time solvability) tends to be inherited from $L(2,1)$ to $L(p,1)$ of $p > 2$; for example, if $L(2,1)$-LABELING is NP-hard in a setting, the hardness proof could be modified to $L(p,1)$-LABELING in the same setting. Designing a polynomial time algorithm is also. We can find various related results in surveys by Calamoneri [5]. See also [24] for algorithmic results.

The notion of $L(p,q)$-LABELING firstly appeared in [20] and [30]. Griggs and Yeh formally introduced the $L(2,1)$-LABELING problem [19]. They also show that $L(2,1)$-LABELING is NP-hard in general. Furthermore, $L(2,1)$-LABELING is shown to be NP-hard even for pla-

* This work was partially supported by JSPS KAKENHI Grant Numbers JP17K19960, 17H01698, 19K21537. A full version of this paper is available in [22].

† kawai.kazuma@g.mbox.nagoya-u.ac.jp
‡ hanaka.91t@g.chuo-u.ac.jp
§ ono@nagoya-u.jp

nar graphs, bipartite graphs, chordal graphs [3], graphs with diameter of 2 [19] and graphs with tree-width 2 [12]. Moreover, for every $k \geq 4$, k-$L(2,1)$-Labeling, that is the decision version of $L(2,1)$-Labeling is NP-complete for general graphs [14] and even for planar graphs [8]. These results imply that k-$L(2,1)$-Labeling is NP-complete for every $\Delta \geq 3$, where Δ denotes the maximum degree. On the other hand, $L(2,1)$-Labeling can be solved in polynomial time for paths, cycles, wheels [19], but these are rather trivial. For non-trivial graph classes, only a few graph classes (e.g., co-graphs [6]) are known to be solvable in polynomial time. In particular, Griggs and Yeh conjectured that $L(2,1)$-Labeling on trees was NP-hard, which was later disproved (under P≠NP) by the existence of an $O(n^{5.5})$-time algorithm [6]. It is now known that $L(p,1)$-Labeling on trees can be solved in linear time [23].

From these results, we roughly understand the boundary between polynomial-time solvability and NP-hardness concerning graph classes, and studies are going to fixed-parameter (in)tractability. For a problem A with input size n and parameter t, A is called *fixed-parameter tractable* with respect to t if there is an algorithm whose running time is $g(t)n^{O(1)}$, where g is a certain function. Such an algorithm is called a *fixed-parameter* algorithm. If problem A is NP-hard for a constant value of t, there is no fixed-parameter algorithm unless P=NP; we say A is paraNP-hard. Unfortunately, $L(2,1)$-Labeling is already shown to be paraNP-hard for several parameters such as $\lambda_{2,1}$, maximum degree and tree-width as seen above. For positive results, it is fixed-parameter tractable with respect to vertex cover number [13] or neighborhood diversity [11]. Note that vertex cover number is a stronger parameter than tree-width, which means that if the vertex cover number is bounded, the tree-width is also. There is still a gap on fixed-parameter (in)tractability between them. For such a situation, two approaches can be taken. One is to finely classify intermediate parameters and see fixed-parameter (in)tractability for them, and the other is to combine two or more parameters and see fixed-parameter (in)tractability under the combinations. In this paper, we take the latter approach.

1.1 Our contribution

In this paper, we present algorithms with combined parameters. The parameters that we focus on are clique-width ($\mathtt{cw}$), tree-width ($\mathtt{tw}$), maximum clique size (ω), maximum degree (Δ) and twin cover number ($\mathtt{tc}$). These are selected in connection with aforementioned parameters, $\lambda_{p,1}$, maximum degree and tree-width. Maximum clique size and clique-width are well used parameters weaker than tree-width. Maximum degree itself is a considered parameter, which is strongly related to $\lambda_{p,q}(G)$. In fact, it is easy to see that $\lambda_{p,1} \geq \Delta + p - 1$, and $\lambda_{p,1} \leq \Delta^2 + (p-1)\Delta - 2$ [18]. Thus, $\lambda_{p,1}$ and Δ are parameters equivalent in terms of fixed-parameter (in)tractability. Twin cover number is picked up as a parameter that is moderately weaker than vertex cover number but stronger than clique-width and is also incomparable to neighborhood diversity.

These parameters are ordered in the following two ways: (1) $(\mathtt{vc} \succeq)\{\mathtt{tw}, \mathtt{tc}\} \succeq \mathtt{cw}$ and (2) $(\lambda_{p,1} \simeq)\Delta \succeq \omega$. Here, for graph parameters α and β, $\alpha \succeq \beta$ represents that there is a positive function g such that $g(\alpha(G)) \geq \beta(G)$ holds for any G, and we denote $\alpha \simeq \beta$ if $\alpha \succeq \beta$ and $\beta \succeq \alpha$. For combined parameters of one from (1) and another from (2), we design fixed-parameter algorithms. Note that some combination yields essentially one parameter. For example, $\mathtt{tw} + \omega$ is equivalent to $\mathtt{tw}$, because $\mathtt{tw} \geq \omega - 1$ holds. The obtained results are listed below:

- $L(p,1)$-Labeling is fixed-parameter tractable (FPT, for short) when parameterized by $\mathtt{cw}+\Delta$ for $p \geq 1$. The proof is based on the monadic second order logic (MSO_1) and the Courcelle's theorem, which implies that the exponent part of the time complexity could be quite large.
- $L(p,1)$-Labeling can be solved in time $\Delta^{O(\mathtt{tw}\Delta)}n$ for $p \geq 1$. Note that the FPT result itself follows from the above FPT result with respect to $\mathtt{cw}+\Delta$. We here give an explicit algorithm. This result also implies that $L(p,1)$-Labeling is FPT when parameterized by band-width.
- $L(p,1)$-Labeling is FPT when parameterized by $\mathtt{tc}+\omega$. Since $\mathtt{tc}+\omega \leq \mathtt{vc}+1$ for any graph, it generalizes the fixed-parameter tractability with respect to vertex cover number in [13].
- $L(1,1)$-Labeling is FPT when parameterized by *only* twin cover number. This also yields a fixed-parameter p-approximation algorithm for $L(p,1)$-Labeling with respect to twin cover number.

Figure 1 illustrates the detailed relationship between graph parameters and the parameterized complexity of $L(p,1)$-Labeling.

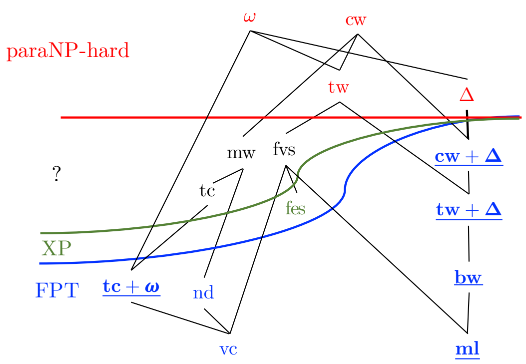

Figure 1. The relationship between graph parameters and the parameterized complexity of $L(p, 1)$-Labeling. Let $\mathtt{cw}, \omega, \Delta, \mathtt{mw}, \mathtt{nd}, \mathtt{tc}, \mathtt{tw}, \mathtt{fvs}, \mathtt{fes}, \mathtt{bw}, \mathtt{ml}$, and $\mathtt{vc}$ denote clique-width, maximum clique size, modular-width, neighborhood diversity, twin cover number, tree-width, feedback vertex set number, feedback edge set number, band-width, max leaf number, and vertex cover number, respectively. Connections between two parameters imply that the upper is bounded by a function of the lower. The underlines for parameters indicate that they are obtained in this paper.

1.2 Related work

As mentioned above, $L(p, 1)$-Labeling is NP-hard even on graphs of tree-width 2 [12]. Using stronger parameters than tree-width, Fiala et al. showed that $L(p, 1)$-Labeling is fixed-parameter tractable when parameterized by vertex cover [13] and neighborhood diversity [10]. Moreover, Fiala, Kloks and Kratochvíl showed that the problem is XP when parameterized by feedback edge set number [14]. For approximation, it is NP-hard to approximate $L(p, 1)$-Labeling within a factor of $n^{0.5-\varepsilon}$ for any $\varepsilon > 0$, whereas it can be approximated within $O(n(\log\log n)^2/\log^3 n)$ [21]. For $L(1, 1)$-Labeling, it can be solved in time $O(\Delta^{2^{8(\mathtt{tw}+1)+1}} n + n^3)$, and hence it is XP by tree-width [32]. This result is tight in the sense of fixed-parameter (in)tractability, because it is W[1]-hard with respect to tree-width [13]. Moreover, it can be solved in time $O(\mathtt{cw}^3 2^{6\mathtt{cw}} n^{2^{4\mathtt{cw}}+2^{2\mathtt{cw}}+1})$ [31].

Apart from $L(p, 1)$-Labeling, twin cover number is a relatively new graph parameter, which is introduced in [15] as a stronger parameter than vertex cover number. In the same paper, many problems are shown to be FPT when parameterized by twin cover number, and it is getting to be a standard parameter (e.g., [1, 9, 16, 25, 26]). Recently, for Imbalance, which is one of graph layout problems, a parameterized algorithm is presented [28]. It is interesting that they also adopt twin cover number plus maximum clique size as the parameters.

2 Preliminaries

In this paper, we use the standard graph notations. Suppose that $G = (V, E)$ is a simple and connected graph with the vertex set V and the edge set E. We sometimes use $V(G)$ or $E(G)$ instead of V or E respecively, to specify graph G. For $G = (V, E)$, we denote the numbers of vertices and edges by $n = |V|$ and $m = |E|$, respectively. For $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V'. For two vertices u and v, the *distance* $\mathrm{dist}_G(u, v)$ is defined by the length of a shortest path between u and v where the length of a path is the number of edges of it. We denote the closed neighbourhood and the open neighbourhood of a vertex v by $N_G[v]$ and $N_G(v)$, respectively. For a set $S \subseteq V$, let $N_G(S) = \bigcup_{v\in S} N_G(v)$ and $N_G[S] = \bigcup_{v\in S} N_G[v]$. The degree of v is denoted by $d_G(v) = |N_G(v)|$. The maximum degree of G is denoted by $\Delta(G)$. For simplicity, we sometimes omit the subscript G.

The k-th power $G^k = (V, E^k)$ of a graph $G = (V, E)$ is a graph such that the set of vertices is V and there is an edge (u, v) in E^k if and only if there is a path of length at most k between u and v in G [4]. In particular, G^2 is called the *square* of G.

In the following, we introduce several graph parameters.

Definition 1 (Tree Decomposition). *A* tree decomposition *of a graph $G = (V, E)$ is defined as a pair $\langle \mathcal{X}, T\rangle$, where T is a tree with node set $I(T)$ and $\mathcal{X} = \{X_i \mid i \in I(T)\}$ is a collection of subsets, called* bags*, of V such that:*

1. (vertex condition) $\bigcup_{i\in I(T)} X_i = V$;
2. (edge condition) *For every $\{u, v\} \in E$, there exists an $i \in I(T)$ such that $\{u, v\} \subseteq X_i$;*
3. (coherence property) *For every $u \in V$, $I_u = \{i \in I(T) \mid u \in X_i\}$ induces a connected subtree of T.*

The width *of a tree decomposition is defined as $\max_{i\in I} |X_i| - 1$ and the* tree-width *of G, denoted by $\mathtt{tw}(G)$, is defined as the minimum width among all possible tree decompositions of G.*

Two vertices u, v are called *twins* if both u and v have the same neighbors. Moreover, if twins u, v have edge $\{u, v\}$, they are called *true twins* and the edge is called a *twin edge*. Then a *twin cover* of G is defined as follows.

Definition 2 (Twin Cover, [15]). *A set of vertices X is a* twin cover *of G if every edge $\{u, v\} \in E$ satisfies either (1) $u \in X$ or $v \in X$, or (2) u, v are true twins. The* twin cover number *of G, denoted by $\mathtt{tc}(G)$,* *is defined as the minimum size of twin covers in G.*

An important observation is that the complement $V \setminus X$ of a twin cover X induces disjoint cliques. Moreover, for each clique Z of $G[V \setminus X]$, $N(u) \cap X = N(v) \cap X$ for every $u, v \in Z$ [15].

A *vertex cover* X is the set of vertices such that for every edge, at least one endpoint is in X. The *vertex cover number* of G, denoted by $\mathtt{vc}(G)$, is defined as the minimum size of vertex covers in G. Since every vertex cover of G is also a twin cover of G, $\mathtt{tc}(G) \leq \mathtt{vc}(G)$ holds. Also, for any graph G, we have $\mathtt{tc}(G) + \omega(G) \leq \mathtt{vc}(G) + 1$.

3 Parameterization by $\mathtt{cw} + \Delta$

As $L(p,1)$-Labeling is paraNP-hard for tree-width, so is for clique-width. In this section, as a complement, we show that $L(p,1)$-Labeling (actually, $L(p,q)$-Labeling for any constant p and q) is fixed-parameter tractable when parameterized by $\mathtt{cw} + \Delta$.

To show this, we give a one-sorted monadic-second order logic (MSO_1) representation of k-$L(p,q)$-Labeling. We first define the following formula $\mathsf{dist}_{=2}(u,w)$, which is true if the distance between u and w is exactly 2:

$$\begin{aligned}\mathsf{dist}_{=2}(u,w) :=& (u \neq w) \wedge \neg\mathsf{adj}(u,w)\\ & \wedge (\exists v \in V : (u \neq v)\\ & \wedge (v \neq w) \wedge (\mathsf{adj}(u,v) \wedge \mathsf{adj}(v,w)).\end{aligned}$$

Then the formula φ_k such that $G \models \phi_k$ if and only if (G,k) is a yes instance of $L(p,q)$-Labeling is defined as follows:

$$\begin{aligned}\varphi_k :=& \exists V_0, \ldots, V_k : \left(\forall v : \bigvee_{0 \leq i \leq k} (v \in V_i \wedge \bigwedge_{0 \leq j \neq i \leq k} v \notin V_j)\right)\\ & \wedge (\forall u, v : \mathsf{adj}(u,v) \Rightarrow \bigvee_{0 \leq i \leq k} (u \in V_i \wedge (\bigwedge_{i-p+1 \leq j \leq i+p-1} v \notin V_j)))\\ & \wedge \left(\forall u, v : (\mathsf{dist}_{=2}(u,v)) \Rightarrow \bigvee_{0 \leq i \leq k} (u \in V_i \wedge (\bigwedge_{i-q+1 \leq j \leq i+q-1} v \notin V_j))\right).\end{aligned}$$

For a graph G of clique-width at most $\mathtt{cw}$ and for an MSO_1 formula ψ, it can be checked whether $G \models \psi$ in time $O(g(|\psi|, \mathtt{cw}) \cdot n^3)$, where g is some computable function [7, 29]. Because the length of MSO_1 formula φ_k depends on k, p, and q, k-$L(p,q)$-Labeling is fixed-parameter tractable when parameterized by $k + \mathtt{cw}$. Thus, we obtain the following theorem.

Theorem 1. *For any fixed p, q, $L(p,q)$-Labeling is fixed-parameter tractable when parameterized by $\lambda_{p,q} + \mathtt{cw}$.*

As for the labeling number, since the degree of G^2 is Δ^2, $\lambda_{1,1}(G) \leq \Delta(G)^2$ holds. This and $\lambda_{cp,cq} = c\lambda_{p,q}$ ([17]) imply that $\lambda_{p,q} \leq \max\{p,q\}\Delta^2$ holds. For $q = 1$, a better bound $\lambda_{p,1} \leq \Delta^2 + (p-1)\Delta - 2$ is known [18]. Thus we have the following corollary.

Corollary 1. *For any positive constant p and q, $L(p,q)$-Labeling is fixed-parameter tractable when parameterized by $\Delta + \mathtt{cw}$.*

4 Parameterization by $\mathtt{tw} + \Delta$

In previous section, we show that $L(p,1)$-Labeling is fixed-parameter tractable when parameterized by $\mathtt{cw} + \Delta$. However, it is shown by using the MSO_1 representation and the Courcelle's theorem. Thus, the exponent part of the running time of the algorithm might be quite large. In this section, we give an explicit fixed-parameter algorithm for $L(p,1)$-Labeling parameterized by $\mathtt{tw} + \Delta$. The running time is $\Delta^{O(\mathtt{tw}\Delta)}n$.

In the algorithm, we first construct the square G^2 of G and then compute $L(p,1)$-Labeling of G by dynamic programming on a nice tree decomposition $\langle \mathcal{X}', T' \rangle$ of G^2. Actually, the algorithm runs for $L(p,q)$-Labeling though the running time depends on λ. One can obtain the square of G^2 in time $O(m\Delta(G)) = O(\Delta(G)^2 n)$. We then prove the following lemma.

Lemma 1. *Given a tree decomposition of a graph G of width t with ℓ bags, one can construct a tree decomposition of G^2 of width at most $(t+1)\Delta(G) + t$ with ℓ bags in time $O(t\Delta(G)\ell)$.*

Proof. We are given a tree decomposition $\langle \mathcal{X}, T \rangle$ of G of width t. Let $X'_i = X_i \cup N(X_i)$ and $\mathcal{X}' = \{X'_i \mid i \in I(T)\}$ be the set of bags. We here define $\langle \mathcal{X}', T' \rangle$ as a tree decomposition of G^2, where T' and T are identical; T and T' has the same node set and the same structure, where each $i \in I(T')$ corresponds to $i \in I(T)$. In the following, we denote $\langle \mathcal{X}', T \rangle$ instead of $\langle \mathcal{X}', T' \rangle$.

We can see that $\langle \mathcal{X}', T \rangle$ is really a tree decomposition of G^2 with width $(t+1)\Delta(G) + t$. It satisfies the properties of tree decomposition indeed: Since $\bigcup_{i \in I} X'_i = \bigcup_{i \in I}(X_i \cup N(X_i)) = V(G) = V(G^2)$, the vertex condition is satisfied. We next see edge condition. For each $e \in E$, there is X_i containing e, so $e \in X'_i$. For each $\{u,v\} \in E^2 \setminus E$, there is a vertex $v'(\neq u, v)$ such that $\{u,v'\} \in E$ and $\{v',v\} \in E$. Thus there is X_i satisfying $\{u,v'\} \subseteq X_i$, which implies $\{u,v\} \subseteq X_i \cup \{v\} \subseteq X_i \cup N(\{v'\}) \subseteq X'_i$. These show that the edge condition is satisfied.

Finally, we check coherent property: we show that for every $u \in V$, $I'_u = \{i \in I(T) \mid u \in X'_i\}$ induces a connected subtree of T. Note that

$$I'_u = \{i \in I(T) \mid u \in X'_i\} = \{i \in I(T) \mid u \in X_i\} \cup \bigcup_{v \in N(u)} \{i \in I(T) \mid v \in X_i\}.$$

Here, the subgraph T_v of T induced by $\{i \in I(T) \mid u \in X_i\}$ is connected by the coherent property of $\langle \mathcal{X}, T\rangle$. Also for each $v \in N(u)$, the subgraph T_v of T induced by $\{i \in I(T) \mid v \in X_i\}$ is connected. By $\{u, v\} \in E$, the edge condition of $\langle \mathcal{X}, T\rangle$ implies that there exists a bag X_j containing both u and v. Since T_u and T_v has a common node j, the subgraph of T induced by $\{i \in I(T) \mid u \in X_i\} \cup \{i \in I(T) \mid v \in X_i\}$ is also connected, which leads that the subgraph of T induced by I'_u is also connected.

Hence, $\langle \mathcal{X}', T\rangle$ is a tree decomposition of G^2. Since the size of bag X'_i is $|X'_i| = |X_i \cup N(X_i)| = |\bigcup_{u \in X_i} N[u]| \leq (t+1)(\Delta(G)+1)$, the width is at most $(t+1)(\Delta(G)+1)-1 = (t+1)\Delta(G)+t$. The construction of $\langle \mathcal{X}', T\rangle$ is done by preparing each X'_i, which takes $O(t\Delta(G))$ steps for each i. Thus it can be done in time $O(t\Delta(G)\ell)$ in total. □

Corollary 2. $\mathtt{tw}(G^2) \leq (\mathtt{tw}(G)+1)\Delta(G) + \mathtt{tw}(G)$ *holds.*

By the above lemma, the tree-width of G^2 is bounded if $\mathtt{tw}(G)$ and $\Delta(G)$ are bounded. Thus we can design a dynamic programming algorithm on a nice tree decomposition of G^2, although we omit the detail.

Lemma 2. *Given a nice tree decomposition of G^2 of width at most t, one can compute k-$L(p,q)$-Labeling on G in time $O((k+1)^{t+1}t^2 n)$.*

Here, one can construct a tree decomposition $\langle \mathcal{X}, T\rangle$ of G of width $5\mathtt{tw}(G)+4$ with $O(n)$ bags in time $2^{O(\mathtt{tw}(G))}n$ [2]. By Lemma 1, we can obtain a tree decomposition $\langle \mathcal{X}', T\rangle$ of G^2 of width $(5\mathtt{tw}(G)+4+1)\Delta(G) + 5\mathtt{tw}(G)+4 = O(\mathtt{tw}(G)\Delta(G))$ from $\langle \mathcal{X}, T\rangle$ in time $O(\mathtt{tw}(G)\Delta(G)n)$. By Lemma 2 and $\lambda_{p,q} \leq \max\{p,q\}\Delta^2$, we have the following theorem .

Theorem 2. *For any positive constant p and q, $L(p,q)$-Labeling can be solved in time $\Delta^{O(\mathtt{tw}\Delta)}n$.*

Since $\mathtt{tw}(G) \leq \mathtt{bw}(G)$ and $\Delta(G) \leq 2\mathtt{bw}(G)$, we have the following corollary.

Corollary 3. *For any positive constant p and q, $L(p,q)$-Labeling is fixed-parameter tractable when parameterized by band-width.*

5 Parameterization by twin cover number

5.1 $L(p,1)$-Labeling parameterized by $\mathtt{tc}+\omega$

We design a fixed-parameter algorithm for $L(p,1)$-Labeling with respect to $\mathtt{tc}+\omega$. Notice that for a twin cover X of $G=(V,E)$, each of the connected components of $G[V \setminus X]$ forms a clique. We categorize vertices in $V \setminus X$ with respect to the neighbors in X. Let $T_1, T_2, \ldots, T_s$ be the sets of vertices having common neighbors in X, called *types* of vertices in $V \setminus X$, where s is the number of types. Moreover, we say that a clique $C \subseteq V \setminus X$ is of type T_i if $C \subseteq T_i$. Note that $V \setminus X = \bigcup_{i=1}^{s} T_i$. Let $n_i = |T_i|$ and ω_i be the maximum clique size in T_i.

We first see a general property about cliques with a common neighbor: Suppose that a graph G consists of only cliques and common neighbors Y of all the vertices in the cliques. That is, all the vertices are within distance 2. Also suppose that vertices in Y has some labels $a_1, a_2, \ldots, a_{|Y|}$ and L is a set of labels that are at least p apart from $a_1, a_2, \ldots, a_{|Y|}$. Then the following lemma holds.

Lemma 3. *Suppose that a graph G and a label set L are defined as above, and let $C_1, C_2, \ldots, C_h$ be the set of the cliques, in the descending order of the size. If $|L| \geq \sum_j |C_j|$ and $\sum_j |C_j| \geq p|C_1|$ hold, there exists an $L(p,1)$-labeling of $C_1, \ldots, C_h$ using only labels in L.*

Proof. Let $n' = \sum_j |C_j|$ and $\omega = |C_1|$. The statement of the lemma is rewritten as "if $|L| \geq n'$ and $n' \geq p\omega$, all the cliques can be properly labeled with L". Let us assume $L = \{l_1, l_2, \ldots, l_{n'}\}$. Since we can use distinct labels for vertices in $C_1, C_2, \ldots, C_h$, only the distance-1 condition inside of a same clique matters. If $n' \equiv 1 \pmod p$, we label the vertices in $C_1, C_2, \ldots, C_{n'}$ in this order by using labels in order of $l_1, l_{p+1}, l_{2p+1}, \ldots, l_{n'}, l_2, l_{p+2}, l_{2p+2} \ldots, l_{n'-p+2}, l_3 \ldots, l_p, l_{2p}, \ldots,$
$l_{n'-1}$. Note that the vertices in C_1 are labeled by $l_1, l_{p+1}, \ldots, l_{p(\omega-1)+1}$ (note that $p\omega \leq n'$). Since the difference between $l_{\alpha p+i}$ and $l_{(\alpha+1)p+i}$ for each i and α is at least p, the labeling for cliques does not violate the distance-1 condition. We can choose similar orderings for the other residuals. □

Now we go back to the algorithm parameterized by $\mathtt{tc}+\omega$. Given a twin cover X, we say that a k-$L(p,1)$-labeling is *good* for X if it uses only labels in $\{0, 1, \ldots, (2p-1)|X|-p\} \cup \{k-(2p-1)|X|+p, \ldots, k\}$

for X. The following lemma is also important. It can be shown by repeatedly applying Lemma 3 though we omit the detail.

Lemma 4. *Let X be a twin cover in G such that each T_i satisfies $\omega_i \le n_i/p$. If G has a k-$L(p,1)$-labeling, then G also has a good k-$L(p,1)$-labeling for X.*

Thus, we consider to find a good $L(p,1)$-labeling. Using the lemma, we show that $L(p,1)$-LABELING is fixed-parameter tractable with respect to $\mathtt{tc}+\omega$.

Theorem 3. *$L(p,1)$-LABELING is fixed-parameter tractable when parameterized by $\mathtt{tc}+\omega$.*

Proof. We present an algorithm to solve k-$L(p,1)$-LABELING instead of $L(p,1)$-LABELING. We first compute a minimum twin cover X in time $O(1.2738^{\mathtt{tc}} + \mathtt{tc}n + m)$ [15]. For twin cover X, we define T_i's. Then, we define another twin cover of $X' = X \cup \bigcup_{i:\omega_i > n_i/p} T_i$. Since X is a twin cover, X' is also. The size of X' is at most $\mathtt{tc} + 2^{\mathtt{tc}} \cdot p \cdot \omega$, because the number of types is at most $2^{\mathtt{tc}}$ and the size of T_i joining X is at most $p \cdot \omega$. Let $\mathtt{tc}' = |X'|$.

We are now ready to present the core of the algorithm. We classify an instance into two cases. If k is small enough, we can apply a brute-force type algorithm. Otherwise, we try to find a good k-$L(p,1)$-labeling.

(**Case:** $k < 8p\mathtt{tc}'$) For each type T_i, the distance between two vertices in T_i is at most 2. Thus, the labels of vertices in T_i must be different each other. Due to $k < 8p\mathtt{tc}'$, if $|T_i| \ge 8p\mathtt{tc}'$, we conclude that the input is a no-instance. Otherwise, $n = |X'| + \sum |T_i| \le \mathtt{tc}' + 8p\mathtt{tc}'2^{\mathtt{tc}}$ holds, because the number of T_i's is at most $2^{\mathtt{tc}}$. Thus we check all the possible labelings in time $O((8p\mathtt{tc}')^{8p\mathtt{tc}'2^{\mathtt{tc}}})$.

(**Case:** $k \ge 8p\mathtt{tc}'$) Let $\mathcal{C}_0, \mathcal{C}_1, \ldots \mathcal{C}_t$ be the family of all possible set systems on $\{T_1, \ldots, T_s\}$ such that whenever two distinct T_j and $T_{j'}$ are in $\mathcal{C}_i$ then $N(T_j) \cap N(T_{j'}) = \emptyset$. Here, $\mathcal{C}_0$ is the empty set. These are introduced to describe a set of T_j's that can use a same label. For each $\mathcal{C}_i$, we prepare a set L_i of labels, which will be used during the execution of the algorithm to represent the set of labels that could be used for vertices in $T_j \in \mathcal{C}_i$. Note that $L_0, L_1, \ldots, L_t$ must be disjoint each other, and a label in L_i is used exactly once per T_j. We also define L_0 as the set of labels not used in $V \setminus X'$. Each L_i can be empty.

By Lemma 4, there is a good k-$L(2,1)$-labeling for X such that vertices in X only use labels in $\{0, 1, \ldots, 2p(\mathtt{tc}'-1)-p\} \cup \{k-2p(\mathtt{tc}'-1)+p, \ldots, k\}$ if the input is an yes-instance. Thus we try all the possible partial labelings for X, each of which uses only labels in $\{0, 1, \ldots, 2p(\mathtt{tc}'-1)-p\} \cup \{k-2p(\mathtt{tc}'-1)+p, \ldots, k\}$. Since the number of labels is $2(2p(\mathtt{tc}'-1)-p+1) \le 4p\mathtt{tc}'$, there are at most $(4p\mathtt{tc}')^{\mathtt{tc}'}$ possible labelings of X. For each of them we further try all the possible placement of labels in $\{0, 1, \ldots, 2p(\mathtt{tc}'-1)-1\} \cup \{k-2p(\mathtt{tc}'-1)+1, \ldots, k\}$ into $L_0, L_1, \ldots, L_t$, which is a little wider than above. The number of possible placements is at most $t^{4p\mathtt{tc}'}$ due to the disjointness of L_i's. Therefore, the total possible nonisomorphic partial labelings is at most $(4p\mathtt{tc}')^{\mathtt{tc}} \cdot t^{4p\mathtt{tc}'}$. Note that no vertex will be labeled by a label in $\{0, 1, \ldots, 2p(\mathtt{tc}'-1)-1\} \cup \{k-2p(\mathtt{tc}'-1)+1, \ldots, k\}$ hereafter. Thus we consider how we use labels in $\{2p(\mathtt{tc}'-1), \ldots, k-2p(\mathtt{tc}'-1), \ldots, k\}$ for $V \setminus X$, which does not yield any conflict with X.

We then formulate as Integer Linear Programming how many labels should be placed in $L_0, L_1, \ldots, L_t$ for one partial labeling using $\{0, 1, \ldots, 2p(\mathtt{tc}'-1)-p\} \cup \{k-2p(\mathtt{tc}'-1)+p, \ldots, k\}$. For a fixed partial labeling, let a_i be the number of labels that have been already assigned to L_i there, and x_i be a variable representing the number of labels used in L_i in the desired labeling.

The following is the ILP formulation.

$$\begin{cases} x_0 + \cdots + x_t \le k+1 & \\ x_i \ge a_i, & \text{for } i \in \{0, \ldots, t\} \\ \sum_{i:T_j \in \mathcal{C}_i} x_i = |T_j|, & \text{for } j \in \{1, \ldots, s\} \end{cases}$$

The first constraint shows that the total number of labels is at most $k+1$. Note that the number of unused labels is x_0. The second one is for consistency to the partial labeling. The last one, which is the most important, guarantees that every vertex in T_j can receive a label; the number of usable labels is $|\{i \mid T_j \in \mathcal{C}_i\}|$, because a label in L_i is used exactly once per T_j.

If the above ILP has a feasible solution, it is possible to assign labels to all the vertices in $V \setminus X$ if we ignore the distance-1 condition inside of each clique. Actually, we can see that the information is sufficient to give a proper k-$L(p,1)$-labeling. At the beginning of the algorithm, we take twin cover X', which means that for every $T_i \subseteq V \setminus X$, $n_i \ge p\omega_i$ holds. Since cliques in $G[T_i]$ have common neighbors and $n_i \ge p\omega_i$, only the number of available labels matters by Lemma 3. Since the existence of an ILP solution guarantees this, we can decide whether a partial labeling can be extended to a proper k-$L(p,1)$-Labeling, or not.

Because $s \leq 2^{\mathtt{tc}}$ and $t \leq 2^{2^{\mathtt{tc}}}$, the number of variables of the ILP is at most $2^{2^{\mathtt{tc}}}$; it can be solved in FPT time with respect to $\mathtt{tc}$ [27]. Since $\mathtt{tc}' \leq \mathtt{tc} + 2^{\mathtt{tc}} \cdot p \cdot \omega$, the total running time is FPT time with respect to $\mathtt{tc} + \omega$. □

5.2 $L(1,1)$-Labeling parameterized by twin cover number

Unlike $L(p,1)$-labeling with $p \geq 2$, the distance-1 condition of $L(1,1)$-labeling requires just that the labels between adjacent vertices are different. Thus, $L(1,1)$-Labeling seems to be easier than $L(p,1)$-Labeling with $p \geq 2$. Actually, we can show that $L(1,1)$-Labeling is fixed-parameter tractable parameterized only by twin cover number.

Lemma 5. *For a graph G, let u and v be twins with edge $\{u,v\}$. For $G' = (V, E')$ with $E' = E \setminus \{\{u,v\}\}$, any $L(1,1)$-labeling on G' is also an $L(1,1)$-labeling on G and verse visa.*

Corollary 4. *For G' defined as above, $\lambda_{1,1}(G') = \lambda_{1,1}(G)$ holds.*

Let X be a twin cover again, and then each connected component in $G[V \setminus X]$ forms a clique, each of the edges in which are twin edges. Lemma 4 implies that graph G' obtained by removing all the edges in $G[V \setminus X]$ has the same $L(1,1)$-labeling number of G. The above deletion shows that X is also a vertex cover of G'. Since $L(1,1)$-Labeling is fixed-parameter tractable when parameterized by vertex cover number [13], we have the following theorem.

Theorem 4. $L(1,1)$-Labeling *is fixed-parameter tractable when parameterized by twin cover number.*

Since $\lambda_{1,1}(G) \leq \lambda_{p,1}(G) \leq \lambda_{p,p}(G) = p\lambda_{1,1}(G)$ holds, an $L(1,1)$-labeling gives an approximation for $L(p,1)$-Labeling. In fact, by replacing the labels of an optimal $L(1,1)$-labeling of G with multiples of p, we obtain an $L(p,1)$-labeling whose approximation factor is at most p.

Corollary 5. *For $L(p,1)$-*Labeling*, there is a fixed-parameter p-approximation algorithm with respect to twin cover number.*

Reference

[1] Yuichi Asahiro, Hiroshi Eto, Tesshu Hanaka, Guohui Lin, Eiji Miyano, and Ippei Terabaru. Parameterized algorithms for the happy set problem. In *International Conference and Workshops on Algorithms and Computation (WALCOM 2020)*, pages 323–328. Springer, 2020.

[2] Hans L Bodlaender, Pål Gr 朧 nås Drange, Markus S Dregi, Fedor V Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

[3] Hans L Bodlaender, Ton Kloks, Richard B Tan, and Jan Van Leeuwen. Approximations for λ-colorings of graphs. *The Computer Journal*, 47(2):193–204, 2004.

[4] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph Theory*. Springer, 2008.

[5] Tiziana Calamoneri. The $L(h,k)$-labelling problem: an updated survey and annotated bibliography. *The Computer Journal*, 54(8):1344–1371, 2011.

[6] Gerard J Chang and David Kuo. The L(2,1)-labeling problem on graphs. *SIAM Journal on Discrete Mathematics*, 9(2):309–316, 1996.

[7] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

[8] Nicole Eggemann, Frédéric Havet, and Steven D. Noble. k-$L(2,1)$-labelling for planar graphs is NP-complete for $k \geq 4$. *Discrete Applied Mathematics*, 158(16):1777–1788, 2010.

[9] Hiroshi Eto, Tesshu Hanaka, Yasuaki Kobayashi, and Yusuke Kobayashi. Parameterized Algorithms for Maximum Cut with Connectivity Constraints. In *International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148, pages 13:1–13:15, 2019.

[10] Jiří Fiala, Tomáš Gavenčiak, Dušan Knop, Martin Kouteckỳ, and Jan Kratochvíl. Fixed parameter complexity of distance constrained labeling and uniform channel assignment problems. In *International Computing and Combinatorics Conference (COCOON 2016)*, pages 67–78. Springer, 2016.

[11] Jiří Fiala, Tomáš Gavenčiak, Dušan Knop, Martin Kouteckỳ, and Jan Kratochvíl. Parameterized complexity of distance labeling and uniform channel assignment problems. *Discrete Applied Mathematics*, 248:46–55, 2018.

[12] Jiří Fiala, Petr A Golovach, and Jan Kratochvíl. Distance constrained labelings of graphs of bounded treewidth. In *International Colloquium on Automata, Languages, and Programming (ICALP 2005)*, pages 360–372. Springer,

2005.

[13] Jirí Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513–2523, 2011.

[14] Jiří Fiala, Ton Kloks, and Jan Kratochvíl. Fixed-parameter complexity of λ-labelings. *Discrete Applied Mathematics*, 113(1):59 – 72, 2001.

[15] Robert Ganian. Improving vertex cover as a graph parameter. *Discrete Mathematics and Theoretical Computer Science*, 17(2):77–100, 2015.

[16] Serge Gaspers and Kamran Najeebullah. Optimal surveillance of covert networks by minimizing inverse geodesic length. In *AAAI Conference on Artificial Intelligence (AAAI 2019)*, pages 533–540, 2019.

[17] John P. Georges and David W. Mauro. Generalized vertex labelings with a condition at distance two. In *Congressus Numerantium*, volume 109, pages 141–159, 1995.

[18] D. Gonçalves. On the $L(p, 1)$-labelling of graphs. *Discrete Mathematics*, 308(8):1405 – 1414, 2008.

[19] Jerrold R Griggs and Roger K Yeh. Labelling graphs with a condition at distance 2. *SIAM Journal on Discrete Mathematics*, 5(4):586–595, 1992.

[20] William K Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980.

[21] Magnús M Halldórsson. Approximating the $L(h, k)$-labelling problem. *International Journal of Mobile Network Design and Innovation*, 1(2):113–117, 2006.

[22] Tesshu Hanaka, Kazuma Kawai, and Hirotaka Ono. Computing $L(p, 1)$-labeling with combined parameters, 2020. `arXiv:2009.10502`.

[23] Toru Hasunuma, Toshimasa Ishii, Hirotaka Ono, and Yushi Uno. A linear time algorithm for $L(2, 1)$-labeling of trees. *Algorithmica*, 66(3):654–681, 2013.

[24] Toru Hasunuma, Toshimasa Ishii, Hirotaka Ono, and Yushi Uno. Algorithmic aspects of distance constrained labeling: a survey. *International Journal of Networking and Computing*, 4(2):251–259, 2014.

[25] Bart M. P. Jansen and Astrid Pieterse. Optimal data reduction for graph coloring using low-degree polynomials. *Algorithmica*, 81(10):3865–3889, 2019.

[26] Dusan Knop, Tomás Masarík, and Tomás Toufar. Parameterized Complexity of Fair Vertex Evaluation Problems. In *International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138, pages 33:1–33:16, 2019.

[27] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[28] Neeldhara Misra and Harshil Mittal. Imbalance parameterized by twin cover revisited. In *Computing and Combinatorics (COCOON 2020)*, pages 162–173. Springer, 2020.

[29] S. Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms*, 5(1):10:1–10:20, 2008.

[30] Fred S. Roberts. T-colorings of graphs: recent results and open problems. *Discrete Mathematics*, 93(2):229 – 245, 1991.

[31] Ioan Todinca. Coloring powers of graphs of bounded clique-width. In *Graph-Theoretic Concepts in Computer Science (WG 2003)*, pages 370–382. Springer, 2003.

[32] Xiao Zhou, Yasuaki Kanari, and Takao Nishizeki. Generalized vertex-colorings of partial k-trees. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E83-A(4):671–678, 2000.