Distance constrained labeling on graphs with bounded neighborhood diversity *†

Jiří Fiala¹, Tomáš Gavenčiak¹, Dušan Knop¹, Martin Koutecký¹, and Jan Kratochvíl¹

¹ Department of Applied Mathematics, Charles University
Malostranské nám. 25, Prague
{fiala,gavento,knop,koutecky,honza}@kam.mff.cuni.cz

Abstract

We study the complexity of a group of distance-constrained graph labeling problems when parameterized by the neighborhood diversity (\(nd\)), which is a natural graph parameter between vertex cover and clique width. Neighborhood diversity has been used to generalize and speed up FPT algorithms previously parameterized by vertex cover, as is also demonstrated by our paper.

We show that the Uniform Channel Assignment problem is fixed parameter tractable when parameterized by \(nd\) and the largest weight and that every \(L(p_1, p_2, \ldots, p_k)\)-labeling problem is FPT when parameterized by \(nd\), maximum \(p_i\) and \(k\).

These results furthermore yield an FPT algorithms for \(L(p_1, p_2, \ldots, p_k)\)-labeling and Channel Assignment problems when parameterized by vertex cover size, answering an open question of Fiala et al.: Parameterized complexity of coloring problems: Treewidth versus vertex cover, generalizing their results beyond \(L(2,1)\)-labeling, and tightening the complexity gap to clique width, where the problems are already hard for constant clique width.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases distance labeling, channel assignment, bounded vertex cover, fixed parameter tractability

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The task of assigning frequencies to radio transmitters in wireless networks yields an abundance of various mathematical models and related problems. We study a wide group of such discrete optimization problems in terms of parameterized computational complexity, which is one of the central paradigms of contemporary theoretical computer science. We study parameterization of the problems by neighborhood diversity (\(nd\)), a graph parameter lying between clique width and the size of vertex cover. While all the problems are generally NP-hard even for constant clique width, we show them to be in FPT w.r.t. \(nd\). The complexity was previously known to be FPT only for the special case of \(L(p,1)\) labeling when parameterized by vertex cover.

Before further discussion, history remarks and our results we give several formal definitions.

---

* Paper supported by project Kontakt LH12095 and by GAUK project 1784214.
† Second, third and fourth author are supported by the project SVV–2015–260223. First, third and fifth author is supported by project CE-ITI P202/12/G061 of GA ČR.
1.1 Distance constrained labelings

Given a $k$-tuple of positive integers $p_1, \ldots, p_k$, called distance constraints, an $L(p_1, \ldots, p_k)$-labeling of a graph is an assignment $l$ of integer labels to the vertices of the graph such that whenever vertices $u$ and $v$ are at distance $i$, the assigned labels differ by at least $p_i$, formally $|l(u) - l(v)| \geq p_i$ for all $i \in \{1, \ldots, k\}$. Often only non-increasing sequences of distance constraints are considered.

This concept generalizes graph coloring, equivalent to graph labeling. Here, every edge has a prescribed weight. Formally, for each edge $e$, the labeling is required to separate labels of adjacent vertices by at least the given weight. Formally, for each edge $e$, the assigned labels differ by at least $w(e)$, formally $|l(u) - l(v)| \geq w(u,v)$ for all $i \in \{1, \ldots, k\}$. Often only non-increasing sequences of distance constraints are considered.

The $L(1,1)$-labeling is an optimal $\lambda$-labeling if and only if it has an $L(\lambda)$-labeling of span $\lambda$ if and only if it has an $L(cp_1, \ldots, cp_k)$-labeling of span $\lambda c$ for any positive integer $c$.

**Problem 1 (L(p_1, . . . , p_k)-LABELING).**

| Input: | Graph $G$, positive integers $\lambda, p_1, \ldots, p_k$ |
| Question: | Is there a $L(p_1, \ldots, p_k)$ labeling of $G$ using labels form the interval $[0, \lambda]$? |

The $L(2,1)$ problem has been shown to be an $\mathsf{NP}$-complete by Griggs and Yeh\cite{11} by a reduction to the HAMILTONIAN CYCLE problem (with $\lambda = |V_G|$). Fiala, Kratochvíl and Kloks\cite{7} showed that this problem remains $\mathsf{NP}$-complete also for all fixed $\lambda \geq 4$ and that for all $\lambda \leq 3$ $L(2,1)$-LABELING is solvable in polynomial time.

Despite a conjecture that the $L(2,1)$-labeling remains $\mathsf{NP}$-complete on trees\cite{11}, Chang and Kuo\cite{2} showed a dynamic programming algorithm for this problem, as well as for any $L(p_1, p_2)$-LABELING where $p_2$ is a divisor of $p_1$. The remaining cases have been shown to be $\mathsf{NP}$-complete by Fiala, Golovach and Kratochvíl\cite{3}. Furthermore, as the $L(2,1)$-LABELING of series-parallel graphs is $\mathsf{NP}$ complete\cite{5}. Note that this also implies $\mathsf{NP}$-hardness of the problem on graphs of clique width at most 9.

On the other hand, when $\lambda$ is fixed, then the existence of an $L(p_1, \ldots, p_k)$-labeling of $G$ can be expressed in MSO$_1$, hence it allows a linear time-algorithm on any graph of bounded clique width\cite{12}.

We generalize and the work of Fiala et al.\cite{2} who show that the problem of $L(p,1)$-LABELING is $\mathsf{FPT}$ when parameterized by $p$ and the size of the vertex cover. They also propose the complexity of CHANNEL ASSIGNMENT parameterized by vertex cover as an open question, which we answer in Theorem\cite{9} below.

1.2 Channel assignment

The channel assignment is a related concept to graph labeling. Here, every edge has a prescribed weight $w(e)$ and the labeling is required to separate labels of adjacent vertices by the given weight. Formally, for each $(u,v) \in E_G$ it must hold, that $|l(u) - l(v)| \geq w(u,v)$. The associated decision problem is called CHANNEL ASSIGNMENT.

**Problem 2 (CHANNEL ASSIGNMENT).**

| Input: | Graph $G$, $\lambda \geq 0$, edge weights $w(u,v) \geq 0$ |
| Question: | Is there a labeling $l$ of the vertices $[0, \lambda]$ such that $|l(u) - l(v)| \geq w(u,v)$ for all $uv \in E_G$? |
For any bipartite graph (this class includes all trees) the maximal given weight \( w(e) \), which is an obvious necessary lower bound for the span of any labeling. However it is also an upper bound, since a labeling that assigns 0 to one class of the bipartition and \( w(e) \) to the other satisfies all edge constraints. McDiarmid and Reed [16] showed that it is NP-complete to decide whether a graph of tree width 3 allows a channel assignment of given span \( \lambda \). Again, this also implies NP-hardness of the problem already on graphs of clique width at most 17. The case of series-parallel graphs is open for already two decades and only partial results are known [17].

Observe that any instance \( G, \lambda \) of the \( L(p_1, \ldots, p_k)\)-labeling problem of a graph \( G \) can be reduced to an instance \( G^k, w, \lambda \) of the Channel Assignment problem — simply let \( w(u, v) = p_i \) whenever \( u \) and \( v \) are in \( G \) at distance \( i \).

We focus on these labeling problems on bounded neighborhood diversity graphs while keeping the assumption that the bound \( \lambda \) on the span of the labeling is a part of the input. To this purpose we show that on more restricted class of graphs, namely on graphs of bounded neighborhood diversity, the labeling problem allows a polynomial time algorithm.

### 1.3 Neighborhood diversity

**Definition 3 (Neighborhood diversity).** The neighborhood diversity of a graph \( G \), denoted by \( \text{nd}(G) \), is the minimum number \( t \) of classes of a partition \( V_1, \ldots, V_t \) of the vertex set \( V_G \) such that:

- each \( V_i \) induces either an empty subgraph or a complete subgraph of \( G \), and
- for each distinct \( V_i \) and \( V_j \) there are either no edges between \( V_i \) and \( V_j \), or every vertex of \( V_i \) is adjacent to all vertices of \( V_j \).

Lampis [14] significantly reduced (from the tower function to double exponential) the hidden constants of the generic polynomial algorithms for MSO\(_2\) model checking on graphs with bounded vertex cover. He further introduced the parameter neighborhood diversity of a graph. Classes of graphs of bounded neighborhood diversity reside between classes of bounded vertex cover and graphs of bounded clique width. Though several non-MSO\(_1\) problems, e.g. Hamiltonian cycle cannot be solved in polynomial time on graphs of bounded clique width [18] Lampis showed that some of these problems, including Hamiltonian cycle, are indeed fixed parameter tractable on graphs of bounded neighborhood diversity [14].

Gajarský and Obdržálek [10] further deepened the result of Lampis and showed that also MSO\(_1\) with cardinality constraints (cardMSO\(_1\)) expressible problems are fixed parameter tractable when parameterized by \( \text{vc}(G) \) and/or \( \text{nd}(G) \).

It is easy to see that for a graph \( G \) it holds that \( \text{nd}(G) \leq 2^{\text{vc}(G)} + \text{vc}(G) \) where \( \text{vc}(G) \) is the size of minimal vertex cover of the graph \( G \). This is also used in more detail in the proof of Theorem 9.

Observe that an \( n \)-vertex graph of bounded neighborhood diversity can be described in significantly more effective way, using only \( O(\log\, n \cdot \text{nd}(G)^2) \) space:

**Definition 4 (Type graph).** The type graph \( T(G) \) of a graph \( G \) is a graph on \( \text{nd}(G) \) vertices \( \{t_1, \ldots, t_{\text{nd}(G)}\} \), where each \( t_i \) is assigned weight \( s(t_i) = |V_i| \) the size of the corresponding partition. \( T(G) \) contains an edge \( \{t_i, t_j\} \) if and only if the two corresponding partitions \( V_i \) and \( V_j \) form a complete bipartite graph. \( T(G) \) contains a loop \( \{t_i\} \) if and only if the corresponding partition \( t_i \) is a clique. We treat singleton types as cliques. For a fixed \( T(G) \), let \( u \sim_T u' \) indicate that \( u \) and \( u' \) belong to the same vertex class in \( T(G) \).
4 Distance constrained labeling

A graph

Corresponding type-graph

\[ L_{2,1,1} \text{ labelling as } \text{Uniform Channel Assignment} \]

Figure 1 An example of a graph with its neighborhood diversity decomposition, the corresponding type graph and an instance of the \( L_{2,1,1} \) labeling problem expressed as an instance of the Uniform Channel Assignment problem instance.

Observe that up to an isomorphism (of \( G \) as well as of \( T(G) \)) there is one-to-one correspondence between a graph and its type graph. Note also that all labelings considered are maintained valid if we apply an automorphism of the given graph.

Observation 5. Let \( G \) be a graph. Then for any positive integer \( k \) it holds that \( \text{nd}(G) \geq \text{nd}(G^k) \).

In the tendency of simplification of the considered graph of we also restrict the possible edge constraints of the Channel Assignment problem, but we will keep the assumption general enough so any \( L(p_1,\ldots,p_k) \)-labeling problem could be expressed also in this restricted way.

1.4 Our contribution

As noted above, we generalise previous results to \( L(p_1,\ldots,p_k) \)-labeling and Channel Assignment on bounded \( \text{nd} \). However, permitting arbitrary weights (or even just \( \{0,1\} \) weights) on a \( K_n \) (which has \( \text{nd}(K_n) = 1 \)) would already encode a coloring problem on an arbitrary graph. We naturally restrict the weights to be uniform over every edge bundle between vertex type classes. In other words, we define the weights on the edges of the type graph \( T(G) \) (including the loops).
Problem 6 (Uniform Channel Assignment).

**Input:** Graph $G$ with type graph $T(G)$, $\lambda \geq 1$, edge weights $w(u, v) \geq 0$ such that $w(u, v) = w(u', v')$ for all $u \sim_T u'$ and $v \sim_T v'$.

**Question:** Is there a labeling $l$ of the vertices $[0, \lambda]$ such that $|l(u) - l(v)| \geq w(u,v)$ for all $uv \in E_G$?

As the main result of the paper we show the following.

**Theorem 7.** The Uniform Channel Assignment problem is FPT when parameterized by $nd$ and $w_{\text{max}}$.

Namely we can solve it in time $2^{2^{2O(nd(G))w_{\text{max}}}} \log n$ on graph $G$ on $n$ vertices when encoded by $T(G)$ when the uniform weight function is at most $w_{\text{max}}$. A suitable labeling can be found in additional $2^{2^{2O(nd(G))w_{\text{max}}}} n$ time.

Note that the neighborhood diversity and the optimal type graph can be computed in $O(n^3)$ time [14]. From Theorem 7 we get the following result.

**Theorem 8.** The $p_1, \ldots, p_k$, the $L(p_1, \ldots, p_k)$-labeling problem is FPT when parameterized by $nd, k$ and maximum $p_i$ (or equivalently by $nd$ and the tuple $(p_1, \ldots, p_k)$).

Furthermore, our FPT result for Channel Assignment extends to vertex cover even without the uniformity requirement.

**Theorem 9.** The Channel Assignment problem is FPT when parameterized by $w_{\text{max}}$ and the size of vertex cover.

We would like to point out that the running time given in Theorem 9 may seem impractical, but actual implementations of ILP solvers take advantage of the sparsity of the nd-decomposition given by the corresponding proof, as we sketch in Section 4.

2 Representing labelings as sequences and walks

Without loss of generality we may assume that the given graph $G$ and its type graph $T(G)$ are connected, since connected components can be treated independently.

If the type graph $T(G)$ contains a type $t$ not incident with a loop, we may reduce the channel assignment problem to the graph $G'$, obtained from $G$ by deleting all but one vertices of the type $t$. Any channel assignment of $G'$ yields a valid channel assignment of $G$ by using the same label on all vertices of type $t$ in $G$ as was given to the only left vertex of type $t$ in $G$. Observe that adding a loop to a type occurring only once does not affect the resulting graph $G'$. Hence we assume without loss of generality that all types are incident with a loop. We call such type graph reflexive.

**Observation 10.** If the type graph $T(G)$ is reflexive, then no two vertices of $G$ of the same type are labeled the same by any channel assignment $l$.

Up to an isomorphism of the graph $G$, any channel assignment $l$ is uniquely characterized by a sequence of type sets as follows:

**Lemma 11.** For a reflexive $T(G)$ and any $w$ and $\lambda$ it holds that $(T(G), w, \lambda) \in \text{Uniform Channel Assignment}$ if and only if there exists a sequence of sets $T = T_0, \ldots, T_\lambda$ of the following properties:

(i) $T_i \subseteq V_{T(G)}$ for each $i \in [0, \lambda]$,
(ii) for each $t \in V_{T(G)} : s(t) = |\{T_i : t \in T_i\}|$,
(iii) for all \((t, r) \in E_{T(G)} : (t \in T_i \land r \in T_j) \Rightarrow |i - j| \geq w(t, r)\)

**Proof.** Given a channel assignment \(l : V_G \rightarrow [0, \lambda]\), we define the desired sequence \(\mathcal{T}\), such that the \(i\)-th element is the set of types that contain a vertex labeled by \(i\). Formally \(T_i = \{ t : 3u \in V_i : l(u) = i \}\). Now
(i) each element of the sequence is a set of types, possibly empty,
(ii) as all vertices of \(V_i\) are labeled by distinct labels by Observation 10 any type \(t\) occurs in \(s(t)\) many elements of the sequence
(iii) if \(u\) of type \(T_i\) is labeled by \(i\), and it is adjacent to \(v\) of type \(r\) labeled by \(j\), then \(|i - j| = |l(u) - l(v)| \geq w(u, v) = w(t, r)\), i.e. adjacent types \(t\) and \(r\) may appear in sets at distance at least \(w(t, r)\).

In the opposite direction assume that the sequence \(\mathcal{T}\) exists. Then for each set \(T_i\) and type \(t_j \in T_i\) we choose the unique vertex \(u \in V_j\) and label it by \(i\), i.e. \(l(u) = i\).

Now the condition (ii) guarantees that all vertices are labeled, while condition (iii) guarantees that all distance constraints are fulfilled.

Our aim is to partition \(\mathcal{T}\) into repetitive patterns. Observe that Lemma 11 poses no constraints on sets \(T\) that are at distance at least \(w_{\text{max}}\). Hence, we build an auxiliary directed graph \(D\) on all possible sequences of sets of length at most \(z = w_{\text{max}} - 1\).

We note here that the sequences forming \(V_D\) are indeed independent on the function \(s\) assigning the number of vertices of given type. The edges of \(G\) connect sequences that overlap on a fragment of length \(z - 1\), i.e. when they could be consecutive. This construction is well known from the so called shift register graph.

**Definition 12.** For a graph \(F\) and constraints \(w : EF \rightarrow [1, z]\) we define a directed graph \(D\) such that
- the vertices of \(V_D\) are all \(z\)-tuples \((T_1, \ldots, T_z)\) of subsets of \(V_F\) such that for all \((t, r) \in E_F : (t \in T_i \land r \in T_j) \Rightarrow |i - j| \geq w(t, r)\)
- \(((T_1, \ldots, T_z), (T'_1, \ldots, T'_z)) \in ED \Leftrightarrow T_i' = T_i + 1\) for all \(i \in [1, z - 1]\).

As the first condition of the above definition mimics (iii) of Lemma 11 with \(F = T(G)\), any sequence \(\mathcal{T}\) that justifies a solution for \((T(G), w, \lambda)\), can be transformed into a walk of length \(\lambda - z + 1\) in \(D\).

In the opposite direction, i.e. to select only those walks in \(D\) that correspond to a valid channel assignments we need to guarantee also an analogue of the condition (ii) of Lemma 11 i.e. each type should occur sufficiently many times. In this concern we consider only special walks that allow us to count the occurrences within \(z\)-tuples.

Observe that \(V_D\) contains also the \(z\)-tuple \(\emptyset^z = (\emptyset, \ldots, \emptyset)\). In addition, any walk of length \(\lambda - z + 1\) can be converted into a closed walk from \(\emptyset^z\) of length \(\lambda + z + 1\) — as just the corresponding sequence \(\mathcal{T}\) can be padded with additional \(z\) empty sets at the front, and another \(\emptyset^z\) at the end.

**Corollary 13.** A closed walk \(W = W_1, \ldots, W_{\lambda+z+1}\) on \(D\) where \(W_1 = W_{\lambda+z+1} = \emptyset^z\), yields a solution of the Uniform Channel Assignment problem on the instance \((T(G), w, \lambda)\) with reflexive \(T(G)\) if and only if for each \(t \in V_{T(G)}\) holds that \(s(t) = |\{W_i : t \in (W_i)_1\}|\).

Note that our representation of the solution is similar to the hardness results, previously given by Grigs and Yeh 11 and generalized by Bodlaender et al. 11. The previous approach used a Hamilton Path problem on the complement graph to show hardness result of the (restricted) labeling problem.
3 The algorithm

Our proposed algorithm for the Uniform Channel Assignment problem finds a walk $W$ (if it exists) corresponding to the labeling $l$. As the walk $W$ is closed, we decompose it into a collection of cycles in the graph $D$. Even though the collection of cycles may correspond to several walks, we show that each such walk corresponds to a feasible labeling of the given graph of given span $\lambda$.

Let $\mathcal{C} = \{C_1, \ldots, C_c\}$ be the set of all directed cycles in the graph $D$. Let $|C_i|$ be the length of a cycle $c_i \in \mathcal{C}$. It is well known that any closed walk can be decomposed into cycles (not necessarily distinct) and that any multiset of cycles can be composed into a closed walk provided the graph induced by the union of these cycles is connected. This can be found for example as Proposition 1.9.2 in a monograph by Diestel [4].

Lemma 14. For a reflexive $T(G)$ and any $w$ and $\lambda$ it holds that $(T(G), w, \lambda) \in$ Uniform Channel Assignment if and only if there exist non-negative integer coefficients $\alpha_1, \ldots, \alpha_c$ such that

(i) $\sum_{i=1}^c \alpha_i |C_i| = \lambda + z + 1$,
(ii) for each $t \in V_{T(G)}$ holds that $s(t) = \sum_{i=1}^c \alpha_i |\{W \in C : t \in (W)_1\}|$,
(iii) $\bigcup\{C_i : \alpha_i > 0\}$ induces a connected subgraph of $D$ containing the vertex $\emptyset^z$.

Proof. As a direct consequence of Corollary 13, the condition (i) express the length of the walk as a linear combination of the lengths of cycle factors. The condition (ii) is an analogous statement for the number of occurrences of a vertex in the walk. The last condition (iii) is necessary to be able to combine cycles together into a closed walk starting in $\emptyset^z$. ◼

We are ready to state an outline of our algorithm, see Algorithm 1.

\begin{algorithm}
\caption{Finding a channel assignment.}
\begin{algorithmic}[1]
  \Procedure{Finding a channel assignment}{A reflexive type graph $T(G)$, a constraint function $w$ and span $\lambda$.}
    \State \textbf{Input:} A reflexive type graph $T(G)$, a constraint function $w$ and span $\lambda$.
    \State \textbf{Output:} A channel assignment $l : G \to [0, \lambda]$ respecting constraints $w$, if it exists.
    \State \textbf{begin}
      \State 1 Compute $w_{\text{max}}, z$;
      \State 2 Construct the directed graph $D$;
      \State 3 Construct the set of cycles $\mathcal{C}$;
      \State 4 \textbf{foreach} $C' \subseteq \mathcal{C}$ such that $\bigcup C'$ induces a connected subgraph of $D$ containing the vertex $\emptyset^z$ \textbf{do}
        \State 5 solve the following ILP in variables $\alpha_1, \ldots, \alpha_c$:
          \State \hspace{1em} $\alpha_i > 0$ for each $i : C_i \in C'$
          \State \hspace{1em} $\alpha_i = 0$ for each $i : C_i \notin C'$
          \State \hspace{1em} $\sum_{i=1}^c \alpha_i |C_i| = \lambda + z + 1$,
          \State \hspace{1em} $\sum_{i=1}^c \alpha_i |\{W \in C : t \in (W)_1\}| = s(t)$ for each $t \in V_{T(G)}$;
        \State 6 \textbf{if} the ILP has a solution \textbf{then}
          \State 7 combine $\{\alpha_i C_i : i \in [1, c]\}$ into a closed walk $W$;
          \State 8 convert the walk $W$ into a labeling $l$ and \Return $l$;
        \State 6 \textbf{end}
      \State 4 \textbf{end}
      \State 3 return "No channel assignment $l$ exists."
  \State \textbf{end}
\end{algorithmic}
\end{algorithm}
The correctness of Algorithm 1 follows directly from Lemma 14. To complete the proof of Theorem 7, we argue the time complexity as follows:

- Line 1 needs $O(|E_{T(G)}|) = O(nd(G)^2)$ time.
- As $D$ has at most $(2^{nd(G)})^2$ nodes, line 2 needs $2^{O(nd(G)^{2})}$ time.
- We examine each subset of $V_D$, whether any of its ordering is a cycle. There are at most $\left(\left(2^{nd(G)}\right)^2\right)!$ cases, so the overall complexity for the construction of the set $\mathcal{C}$ at line 3 is $2^{O(nd(G)^{2})}$.
- The number of iterations of the foreach cycle at line 4 is bounded by $2^{2^{O(nd(G)^{2})}}$.
- Frank and Tardos [8] (improving the former result due to Lenstra [15]), showed that the time needed to solve the system of inequalities with $p$ integer variables is $O(p^{2.5p+o(p)}L)$, where $L$ is the number of bits needed to encode the input. As we have $2^{2^{O(nd(G)^{2})}}$ variables, the time needed to resolve the system of inequalities is $2^{2^{O(nd(G)^{2})}}\cdot \log n$, where $n$ is the total number of vertices of $G$.
- A solution $\alpha_1, \ldots, \alpha_n$ can be converted into the walk in time $2^{2^{O(nd(G)^{2})}}\cdot n$, and the same bound applies to the conversion of a walk to the labeling at lines 7 and 8.

Lines 7 and 8 need not to be executed if only the decision for the language \textsc{Uniform Channel Assignment} should be achieved.

Theorem 8 follows from Theorem 7 straightforwardly. Given a reflexive $T(G)$ and distance constraints $p_1, \ldots, p_k$, we form an instance of \textsc{Uniform Channel Assignment} as follows:

In the type graph we join vertices $t, r$ that are originally at distance at most $k$. We then assign $p_i$ to any edge $(t, r)$ of the resulting type graph, where $t$ and $r$ were originally at distance $i$. In addition, each loop has assigned weight $p_1$.

The equivalence of the two parameterizations follows from the fact that for a fixed $k$ and maximum $p$ there are only $(p+1)^k$ $k$-tuples to consider, and we can join the FPT algorithms for all these into one program always running the appropriate one.

It is worth noting that the neighborhood diversity and the type graph can be computed in $O(n^3)$ time [13].

4 The Channel Assignment parameterized by vertex-cover

We now use the results of the previous sections to derive an FPT algorithm proposed as Theorem 9. Further on, we will discuss in general, how to derive similar results based on the relation between vertex cover and neighborhood diversity parameters.

Proof of Theorem 9. Recall that for a graph $G$ it holds that $nd(G) \leq 2^{vc(G)} + vc(G)$ since it is possible to construct the type graph $T$ of $G$ as follows. Let $U$ be the optimal vertex cover of $G$. For each vertex $u \in U$, we introduce a vertex to the type graph (call this vertex $u$ again). Next let $I = V(G) \setminus U$ be the independent set of $G$. We subdivide $I$ into sets for every subset $X \subseteq U$:

$$I_{X} := \{v \in I: \{v, x\} \in E(G) \text{ for } x \in X \text{ and } \{v, x\} \notin E(G) \text{ for } x \notin X\}.$$ 

Now it is easy to see, that the sets defined form a neighborhood diversity decomposition and that the size of the decomposition is at most $2^{vc(G)} + vc(G)$.

Before we apply Theorem 7, we have refine sets $I_X$ further more, so that the edge-weights are uniform. For a set $X = \{x_1, \ldots, x_k\} \subseteq U$ and a tuple of positive integers $w = (w_1, \ldots, w_k)$ with $0 < w_i \leq w_{\text{max}}$ for every $1 \leq i \leq k$ we define the set $I^w_X$ as

$$I^w_X := \{v \in I_X: w(v, x_i) = w_i \text{ for } 1 \leq i \leq k\}.$$
Observe, that a refinement of a neighborhood diversity decomposition is again a decomposition. We now estimate the number of types of the refined decomposition. The number of types of the refined decomposition can be upper-bounded by $\text{vc}(G) + (2^{\text{vc}(G)})^{w\text{max}}$. We can now apply Theorem 7 on the refined decomposition. This finishes the proof. ◀

This proof might be interesting on its own as it yields a general method for finding an FPT algorithm for an edge-constrained problem parameterized by the size of vertex cover via its uniform version parameterized by the neighborhood diversity.

If a problem $P$ with edge-constraints allows the existence of an equivalence relation $R$ on the set of edges, a bound $b_G$ and a function $r : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, such that the number of equivalence classes of an independent set is at most $r(\text{vc}(G), b_G)$, and furthermore, if a problem UNIFORM $P$ admits an FPT algorithm when parameterized by neighborhood diversity, then the original problem $P$ admits an FPT algorithm when parameterized by $\text{vc}$ and $b$.

Moreover, if there exists a function $q : \mathbb{N} \to \mathbb{N}$ such that $r(\text{vc}(G), b_G) \leq q(\text{vc}(G))$, then problem $P$ becomes FPT when parameterized by the size of vertex cover only.

5 Conclusion

We present an algorithm for the UNIFORM CHANNEL ASSIGNMENT problem and several complexity consequences for $L(p_1, \ldots, p_k)$-labeling problem. In particular, Theorem 8 extends known results for $L(p, 1)$-labelings problem to labelings with arbitrarily many distance constraints, answering an open question of [?]. Simultaneously, we broaden the considered graph classes by restricting neighborhood diversity instead of vertex cover.

While the main technical tools of our algorithms are bounded-dimension ILP programs, ubiquitous in the FPT area, the paper shows an interesting insight on the nature of the
labelings over the type graph and the necessary patterns of such labelings of very high span. Note that the span of a graph is generally not bounded by any of the considered parameters and may be even proportional to the order of the graph.

Solving a generalized problem on graphs of bounded neighborhood diversity is a viable method for designing FPT algorithms for a given problem on graphs of bounded vertex cover, as demonstrated by this and previous papers. This promotes neighborhood diversity as a parameter that naturally generalizes the widely studied parameter vertex cover.

We would like to point out that the parameter modular width, proposed by Gajarský, Lampis and Ordyniak [9], offers further generalization of neighborhood diversity towards the clique width [3] (dependencies between these graph parameters are depicted in Fig. 3).

As an interesting open problem we ask whether it is possible to strengthen our results to graphs of bounded modular width or whether the problem might be already NP-complete for fixed modular width, as is the case with clique width. For example, the Graph Coloring problem ILP based algorithm for bounded neighborhood diversity translates naturally to an algorithm for bounded modular width. On the other hand, there is no apparent way how our labeling results could be adapted to modular width in a similar way.

References

1. Hans L. Bodlaender, Ton Kloks, Richard B. Tan, and Jan van Leeuwen. λ-coloring of graphs. In Horst Reichel and Sophie Tison, editors, STACS, volume 1770 of Lecture Notes in Computer Science, pages 395–406. Springer, 2000.
2. Gerald J. Chang and David Kuo. The $L(2,1)$-labeling problem on graphs. SIAM Journal on Discrete Mathematics, 9(2):309–316, 1996.
3. Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. Discrete Appl. Math., 101(1–3):77–114, 2000.
4. Reinhard Diestel. Graph Theory, volume 173. Springer-Verlag, graduate texts in mathematics edition, May 2000. chapt. 12.
5. Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Distance constrained labelings of graphs of bounded treewidth. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, ICALP, volume 3580 of Lecture Notes in Computer Science, pages 360–372. Springer, 2005.
6. Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Computational complexity of the distance constrained labeling problem for trees (extended abstract). In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, ICALP (1), volume 5125 of Lecture Notes in Computer Science, pages 294–305. Springer, 2008.
7. Jiří Fiala, Jan Kratochvíl, and Ton Kloks. Fixed-parameter complexity of λ-labelings. Discrete Applied Mathematics, 113(1):59–72, 2001.
8. András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica, 7(1):49–65, 1987.
9. Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. CoRR, abs/1308.2858, 2013.
10. Robert Ganian and Jan Obdržálek. Expanding the expressive power of monadic second-order logic on restricted graph classes. CoRR, abs/1306.5571, 2013.
11. 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.
12. Daniel Kobler and Udi Rotics. Polynomial algorithms for partitioning problems on graphs with fixed clique-width (extended abstract). In Symposium on Discrete algorithms, 12th SODA '01, Washington, pages 468–476. ACM-SIAM, 2001.
J. Fiala, T. Gavenčiak, D. Knop, M. Koutecký and J. Kratochvíl

13 Daniel Král. The channel assignment problem with variable weights. *SIAM J. Discrete Math.*, 20(3):690–704, 2006.

14 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

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

16 Colin McDiarmid and Bruce Reed. Channel assignment on graphs of bounded treewidth. *Discrete Mathematics*, 273(1–3):183–192, 2003.

17 Marián Škvarek. The channel assignment problem for series-parallel graphs. Bachelor’s thesis, Charles University, Prague, 2010. in Czech.

18 Egon Wanke. $k$-NLC graphs and polynomial algorithms. *Discrete Applied Mathematics*, 54(2–3):251–266, 1994.