The Small Set Vertex Expansion Problem

Soumen Maity(✉)

Indian Institute of Science Education and Research, Pune 411008, India
soumen@iiserpune.ac.in

Abstract. Given a graph $G = (V, E)$, the vertex expansion of a set $S \subset V$ is defined as $\Phi^V(S) = \frac{|N(S)|}{|S|}$. In the SMALL SET VERTEX EXPANSION (SSVE) problem, we are given a graph $G = (V, E)$ and a positive integer $k \leq \frac{|V(G)|}{2}$, the goal is to return a set $S \subset V(G)$ of k nodes minimizing the vertex expansion $\Phi^V(S) = \frac{|N(S)|}{k}$; equivalently minimizing $|N(S)|$. SSVE has not been as well studied as its edge-based counterpart SMALL SET EXPANSION (SSE). SSE, and SSVE to a less extend, have been studied due to their connection to other hard problems including the Unique Games Conjecture and Graph Colouring. Using the hardness of MINIMUM k-UNION problem, we prove that SMALL SET VERTEX EXPANSION problem is NP-complete. We enhance our understanding of the problem from the viewpoint of parameterized complexity by showing that (1) the problem is W[1]-hard when parameterized by k, (2) the problem is fixed-parameter tractable (FPT) when parameterized by the neighbourhood diversity `nd`, and (3) it is fixed-parameter tractable (FPT) when parameterized by treewidth `tw` of the input graph.

Keywords: Parameterized complexity · FPT · W[1]-hard · Treewidth · Neighbourhood diversity

1 Introduction

Covering problems are very well-studied in theoretical computer science. Given a set of elements $\{1, 2, ..., n\}$ (called the universe) and a collection $\mathcal{S}$ of m sets whose union equals the universe, the SET COVER problem is to identify the smallest sub-collection of $\mathcal{S}$ whose union equals the universe, and MAX k-COVER is the problem of selecting k sets from $\mathcal{S}$ such that their union has maximum cardinality. MAX k-COVER is known to admit a $(1-\frac{1}{e})$-approximation algorithm (which is also known to be tight) [5]. A natural variation of MAX k-COVER problem is instead of covering maximum number of elements, the problem is to cover minimum number of elements of the universe by the union of k sets. MINIMUM k-UNION [2,16] is one of such problems, where we are given a family of sets within a finite universe and an integer k and we are asked to choose k sets from this family in order to minimise the number of elements of universe

The author's research was supported in part by the Science and Engineering Research Board (SERB), Govt. of India, under Sanction Order No. MTR/2018/001025.

W. Wu and Z. Zhang (Eds.): COCOA 2020, LNCS 12577, pp. 257–269, 2020.
https://doi.org/10.1007/978-3-030-64843-5_18

that are covered. MINIMUM k-UNION has not been studied until recently, when an $O(\sqrt{m})$-approximation algorithm is given by Eden Chlamtác et al. [3], where m is the number of sets in $\mathcal{S}$. Given an instance of MINIMUM k-UNION, we can construct the obvious bipartite graph in which the left side represents sets and the right side represents elements and there is an edge between a set node and an element node if the set contains the element. Then MINIMUM k-UNION is clearly equivalent to the problem of choosing k left nodes in order to minimize the size of their neighbours. This is known as the SMALL SET BIPARTITE VERTEX EXPANSION (SSBVE) problem [2]. This is the bipartite version of the SMALL SET VERTEX EXPANSION, in which we are given an arbitrary graph and are asked to choose a set S of k nodes minimizing the vertex expansion $\Phi^V(S) = |N(S)|$. SMALL SET VERTEX EXPANSION problem is vertex version of the SMALL SET EXPANSION (SSE) problem, in which we are asked to choose a set of k nodes to minimize the number of edges with exactly one endpoint in the set. SSVE has not been as well studied as SSE, but has recently received significant attention [12]. SSE, and SSVE to a less extend, have been studied due to their connection to other hard problems including the Unique Games Conjecture [8]. These problems recently gained interest due to their connection to obtain sub-exponential time, constant factor approximation algorithm for may combinatorial problems like Sparsest Set and Graph Colouring [1].

A problem with input size n and parameter k is said to be 'fixed-parameter tractable (FPT)' if it has an algorithm that runs in time $\mathcal{O}(f(k)n^c)$, where f is some (usually computable) function, and c is a constant that does not depend on k or n. What makes the theory more interesting is a hierarchy of intractable parameterized problem classes above FPT which helps in distinguishing those problems that are not fixed parameter tractable. For the standard concepts in parameterized complexity, see the recent textbook by Cygan et al. [4].

Our main results are the following:

- The SMALL SET VERTEX EXPANSION (SSVE) problem is NP-complete.
- SSVE is W[1]-hard when parameterized by k.
- SSVE is fixed-parameter tractable (FPT) when parameterized by neighbourhood diversity of the input graph.
- SSVE is FPT when parameterized by treewidth of the input graph.

Related Results: Despite being a very natural problem, MINIMUM k-UNION/ SSBVE has received surprisingly little attention. Chlamtác et al. [3] gave an $O(\sqrt{n})$-approximation algorithm for SSBVE and equivalently $O(\sqrt{m})$-approximation algorithm for MINIMUM k-UNION problem. Louis and Makarychev [12] studied approximation algorithms for hypergraph small set expansion and small set vertex expansion problem. They provided a polylogarithmic approximation when k is very close to n, namely, $k \geq \frac{n}{\text{ploylog}(n)}$. To the best of our knowledge, the parameterized complexity of SSVE and SSE problems have not been studied before. Raghavendra and Steurer [13] have investigated the connection between Graph Expansion and the UNIQUE GAMES CONJECTURES. They proved that a simple decision version of the problem of approximately small set expansion reduces to UNIQUE GAMES.

2 Preliminaries

The vertex and edge expansion in graphs have been a subject of intense study with applications in almost all branches of theoretical computer science. From an algorithmic standpoint SSVE and SSE are fundamental optimization problems with numerous applications. The computational complexity of computing and approximating expansion is still not very well understood. Throughout this article, $G = (V, E)$ denotes a finite, simple and undirected graph of order $|V(G)| = n$. For a vertex $v \in V$, we use $N(v) = \{u : (u, v) \in E(G)\}$ to denote the (open) neighbourhood of vertex v in G, and $N[v] = N_G(v) \cup \{v\}$ to denote the closed neighbourhood of v. The degree $d(v)$ of a vertex $v \in V(G)$ is $|N(v)|$. For a subset $S \subseteq V(G)$, we define its closed neighbourhood as $N[S] = \bigcup_{v \in S} N[v]$ and its open neighbourhood as $N(S) = N[S] \setminus S$. Given a graph $G = (V, E)$, the vertex expansion of a set $S \subset V$ is defined as

$$\Phi^V(S) = \frac{|N(S)|}{|S|}.$$

Definition 1. [2] In the Small Set Vertex Expansion (SSVE) problem, we are given a graph $G = (V, E)$ and an integer $k \leq \frac{|V|}{2}$. The goal is to return a subset $S \subset V$ with $|S| = k$ minimizing the vertex expansion $\Phi^V(S) = \frac{|N(S)|}{k}$; equivalently minimizing $|N(S)|$.

The edge expansion of a subset of vertices $S \subset V$ in a graph G measures the fraction of edges that leaves S. For simplicity we consider regular graphs in the definition of Small Set Expansion (SSE). In a d-regular graph, the edge expansion $\Phi(S)$ of a subset $S \subset V$ is defined as

$$\Phi(S) = \frac{|E(S, V \setminus S)|}{d|S|}$$

where $E(S, V \setminus S)$ denotes the set of edges with one endpoint in S and other endpoint in $V \setminus S$.

Definition 2. [2] In the Small Set Expansion (SSE) problem, we are given a d-regular graph $G = (V, E)$ and an integer $k \leq \frac{|V|}{2}$. The goal is to return a subset $S \subset V$ with $|S| = k$ minimizing the edge expansion $\Phi(S) = \frac{|E(S, V \setminus S)|}{kd}$; equivalently minimizing $|E(S, V \setminus S)|$.

Among the two notions of expansion, this work will concern with vertex expansion. The decision version of the problem studied in this paper is formalized as follows:

Small Set Vertex Expansion
Input: An undirected graph $G = (V, E)$ and two positive integers $k \leq \frac{|V|}{2}$, $\ell \leq |V(G)|$.
Question: Is there a set $S \subset V(G)$ with $|S| = k$ such that the vertex expansion $\Phi^V(S) = |N(S)| \leq \ell$?

We now recall some graph parameters used in this paper. The graph parameters we explicitly use in this paper are neighbourhood diversity nd and treewidth tw. We now review the concept of a tree decomposition, introduced by Robertson and Seymour in [14].

Definition 3. A *tree decomposition* of a graph G is a pair $(T, \{X_t\}_{t \in V(T)})$, where T is a tree and each node t of the tree T is assigned a vertex subset $X_t \subseteq V(G)$, called a bag, such that the following conditions are satisfied:

1. Every vertex of G is in at least one bag.
2. For every edge $uv \in E(G)$, there exists a node $t \in T$ such that bag X_t contains both u and v.
3. For every $u \in V(G)$, the set $\{t \in V(T) \mid u \in X_t\}$ induces a connected subtree of T.

Definition 4. The *width* of a tree decomposition is defined as $width(T) = max_{t \in V(T)} |X_t| - 1$ and the treewidth $tw(G)$ of a graph G is the minimum width among all possible tree decomposition of G.

A special type of tree decomposition, known as a *nice tree decomposition* was introduced by Kloks [9]. The nodes in such a decomposition can be partitioned into four types.

Definition 5. A tree decomposition $(T, \{X_t\}_{t \in V(T)})$ is said to be *nice tree decomposition* if the following conditions are satisfied:

1. All bags correspond to leaves are empty. One of the leaves is considered as root node r. Thus $X_r = \emptyset$ and $X_l = \emptyset$ for each leaf l.
2. There are three types of non-leaf nodes:
   - **Introduce node:** a node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$; we say that v is *introduced* at t.
   - **Forget node:** a node t with exactly one child t' such that $X_t = X_{t'} \setminus \{w\}$ for some $w \in X_{t'}$; we say that w is *forgotten* at t.
   - **Join node:** a node with two children t_1 and t_2 such that $X_t = X_{t_1} = X_{t_2}$.

Note that, by the third property of tree decomposition, a vertex $v \in V(G)$ may be introduced several time, but each vertex is forgotten only once. To control introduction of edges, sometimes one more type of node is considered in nice tree decomposition called introduce edge node. An *introduce edge node* is a node t, labeled with edge $uv \in E(G)$, such that $u, v \in X_t$ and $X_t = X_{t'}$, where t' is the only child of t. We say that node t introduces edge uv. It is known that if a graph G admits a tree decomposition of width at most tw, then it also admits a nice tree decomposition of width at most tw, that has at most $O(n \cdot \texttt{tw})$ nodes [4].

3 Proving Small Set Vertex Expansion is NP-complete

Using the hardness of Minimum k-Union problem, we prove that Small Set Vertex Expansion problem is NP-complete. We state the decision version of Minimum k-Union problem.

Definition 6. [2] In Minimum k-Union problem, we are given an universe $U = \{1, 2, \ldots, n\}$ of n elements and a collection of m sets $\mathcal{S} \subseteq 2^U$, as well as two integers $k \leq m$ and $\ell \leq n$. Does there exist a collection $T \subseteq \mathcal{S}$ with $|T| = k$ such that $|\cup_{S \in T} S| \leq \ell$?

It is known that Minimum k-Union problem is NP-complete [16]. Now we prove the following hardness result.

Theorem 1. *The* Small Set Vertex Expansion *problem is NP-complete.*

Proof. We first show that Small Set Vertex Expansion problem is in NP. Given a graph $G = (V, E)$ with n vertices and two integers $k \leq \frac{n}{2}$ and $\ell \leq n$, a certificate could be a set $S \subset V$ of size k. We could then check, in polynomial time, there are k vertices in S, and the vertex expansion $\Phi^V(S) = |N(S)|$ is less than or equal to ℓ. We prove the Small Set Vertex Expansion problem is NP-hard by showing that Minimum k-Union $\leq_P$ Small Set Vertex Expansion. Given an instance $(U, \mathcal{S}, k, \ell)$ of Minimum k-Union problem, we construct a graph H with vertex sets X and Y. The vertices in $X = \{s_1, s_2, \ldots, s_m\}$ correspond to sets in $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$; the vertices in $Y = \{u_1, u_2, \ldots, u_n\}$ are the elements in U. We make $s_j \in X$ adjacent to $u_i \in Y$ if and only if $u_i \in S_j$. Additionally, for each vertex u_i, we add a clique of size $n+1$, K^i_{n+1} and we make u_i adjacent to each vertex in K^i_{n+1}.

We show that there is a collection of k sets $\{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\} \subseteq \mathcal{S}$ such that $|\cup_{j=1}^k S_{i_j}| \leq \ell$, for Minimum k-Union problem if and only if there is a set $S \subset V(H)$ of $k \leq \frac{|V(H)|}{2}$ vertices such that $|N_H(S)| \leq \ell$, for Small Set Vertex Expansion problem. Suppose there is a collection of k sets $\{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\} \subseteq \mathcal{S}$ such that $|\cup_{j=1}^k S_{i_j}| \leq \ell$. We choose the vertices $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\} \subseteq X$ correspond to sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$. As the size of the union of these k sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ is less or equal to ℓ, the vertex expansion of $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ is also at most ℓ. If $k > \frac{|V(H)|}{2}$, then S is any size $\frac{|V(H)|}{2}$ subset of $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ and it has vertex expansion at most ℓ.

Conversely, suppose there is a subset $S \subseteq V(H)$ of k vertices such that $\Phi^V(S) \leq \ell$ where $\ell \leq n$. Note that S cannot contain any vertex from Y as each vertex in Y has at least $n+1$ neighbours in H. Similarly S cannot contain any vertex from K^i_{n+1}, as each vertex in K^i_{n+1} has at least $n+1$ neighbours in H. Thus $S \subseteq X$ and let $S = \{s_{j_1}, s_{j_2}, \ldots, s_{j_k}\}$. We consider the k sets $S_{j_1}, S_{j_2}, \ldots, S_{j_k}$ correspond to these k vertices. As $\Phi^V(S) = |N(S)| \leq \ell$, we have $|\cup_{i=1}^k S_{j_i}| \leq \ell$. This completes the proof.

4 W[1]-Hardness Parameterized by *k*

The input to the decision version of SSVE is a graph G with two integers $k \leq \frac{n}{2}$ and $\ell \leq n$, and (G, k, ℓ) is a yes-instance if G has a set S of k vertices such that the vertex expansion $\Phi^V(S) = |N(S)| \leq \ell$. In this section we show that SSVE is $W[1]$-hard when parameterized by k, via a reduction from CLIQUE.

Theorem 2. *The* SMALL SET VERTEX EXPANSION *problem is* $W[1]$*-hard when parameterized by* k*.*

Proof. Let (G, k) be an instance of CLIQUE. We construct an instance $(G', \frac{k(k-1)}{2}, k)$ of SMALL SET VERTEX EXPANSION problem as follows. We construct a graph G' with vertex sets X and Y, where $X = V(G) = \{v_1, v_2, \ldots, v_n\}$ and $Y = E(G) = \{e_1, e_2, \ldots, e_m\}$, the edge set of G. We make v_i adjacent to e_j if and only if v_i is an endpoint of e_j. We further add a set $P = \{p_1, p_2, \ldots, p_{k^2}\}$ of k^2 vertices; the vertices in P are adjacent to every element of X and all vertices in P are pairwise adjacent.

We claim that there is a set S of $\frac{k(k-1)}{2}$ vertices in G' with vertex expansion $\Phi^V(S) = |N(S)| \leq k$ if and only if G contains a clique on k vertices. Suppose first that G contains a clique on k vertices $\{v_1, v_2, \ldots, v_k\}$; we set S to be the set of edges belonging to this clique, and notice that in G all endpoints of edges in S belong to the set $\{v_1, v_2, \ldots, v_k\}$. Thus the vertex expansion of S in G' is exactly $\{v_1, v_2, \ldots, v_k\}$ and $\Phi^V(S) = |N(S)| = k$, so we have a yes-instance for $(G', \frac{k(k-1)}{2}, k)$.

Conversely, suppose that G' contains a set S of $\frac{k(k-1)}{2}$ vertices such that $\Phi^V(S) = |N(S)| \leq k$. As $d(v) \geq k + 1$ for every vertex $v \in X \cup P$, we cannot include any vertex of X or P in the set S. So we conclude that $S \subseteq Y$ is a set of edges in G. All edges in S belong to the subgraph of G induced by $N(S)$, which by assumption has at most k vertices. Since $|S| = \frac{k(k-1)}{2}$, this is only possible if $|N(S)| = k$ and $N(S)$ in fact induces a clique in G, as required.

5 FPT Algorithm Parameterized by Neighbourhood Diversity

In this section, we present an FPT algorithm for the SMALL SET VERTEX EXPANSION problem parameterized by neighbourhood diversity. We say two vertices u and v have the same type if and only if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. The relation of having the same type is an equivalence relation. The idea of neighbourhood diversity is based on this type structure.

Definition 7. [10] The neighbourhood diversity of a graph $G = (V, E)$, denoted by $\mathtt{nd}(G)$, is the least integer w for which we can partition the set V of vertices into w classes, such that all vertices in each class have the same type.

If neighbourhood diversity of a graph is bounded by an integer w, then there exists a partition $\{C_1, C_2, \ldots, C_w\}$ of $V(G)$ into w type classes. It is known that such a minimum partition can be found in linear time using fast modular decomposition algorithms [15]. Notice that each type class could either be a clique or an independent set by definition. For algorithmic purpose it is often useful to consider a *type graph* H of graph G, where each vertex of H is a type class in G, and two vertices C_i and C_j are adjacent iff there is complete bipartite clique between these type classes in G. It is not difficult to see that there will be either a complete bipartite clique or no edges between any two type classes. The key property of graphs of bounded neighbourhood diversity is that their type graphs have bounded size. In this section, we prove the following theorem:

Theorem 3. *The* SMALL SET VERTEX EXPANSION *problem is fixed-parameter tractable when parameterized by the neighbourhood diversity.*

Given a graph $G = (V, E)$ with neighbourhood diversity $\mathtt{nd}(G) \leq w$, we first find a partition of the vertices into at most w type classes $\{C_1, \ldots, C_w\}$. Next we guess a set of type classes C_i for which $C_i \cap S \neq \emptyset$, where S is a set with k vertices such that the vertex expansion $\Phi^V(S) = |N(S)|$ is minimum. Let $\mathcal{P} \subseteq \{C_1, \ldots, C_w\}$ be a collection of type classes for which $C_i \cap S \neq \emptyset$. There are at most 2^w candidates for $\mathcal{P}$. Finally we reduce the problem of finding a set S that minimizes the vertex expansion $\Phi^V(S)$ to 2^w integer linear programming (ILP) optimizations with at most w variables in each ILP optimization. Since ILP optimization is fixed parameter tractable when parameterized by the number of variables [6], we conclude that our problem is fixed parameter tractable when parameterized by the neighbourhood diversity w.

ILP Formulation: For each C_i, we associate a variable x_i that indicates $|S \cap C_i| = x_i$. As the vertices in C_i have the same neighbourhood, the variables x_i determine S uniquely, up to isomorphism. We define

$$r(C_i) = \begin{cases} 1 & \text{if } C_i \text{ is adjacent to some } C_j \in \mathcal{P}; i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathcal{C}$ be a subset of $\mathcal{P}$ consisting of all type classes which are cliques; $\mathcal{I} = \mathcal{P} \setminus \mathcal{C}$ and $\mathcal{R} = \{C_1, \ldots, C_w\} \setminus \mathcal{P}$. Given a $\mathcal{P} \subseteq \{C_1, \ldots, C_w\}$, our goal is to minimize

$$\Phi^V(S) = |N(S)| = \sum_{C_i \in \mathcal{R}} r(C_i)|C_i| + \sum_{C_i \in \mathcal{C}} (|C_i| - x_i) + \sum_{C_i \in \mathcal{I}} r(C_i)(|C_i| - x_i) \quad (1)$$

under the condition $x_i \in \{1, \ldots, |C_i|\}$ for all $i : C_i \in \mathcal{P}$ and $x_i = 0$ for all $i : C_i \in \mathcal{R}$ and the additional conditions described below. Note that if $C_i \in \mathcal{R}$ and it is adjacent to some type class in $\mathcal{P}$, then C_i is contained in $N(S)$; if $C_i \in \mathcal{C}$ then $|C_i| - x_i$ vertices of C_i are in $N(S)$; finally if $C_i \in \mathcal{I}$ and it is adjacent to some type class in $\mathcal{P}$, then $|C_i| - x_i$ vertices of C_i are in $N(S)$. It is easy to see that minimizing the expansion $\Phi^V(S)$ in Eq. 1 is equivalent to maximizing $\sum_{C_i \in \mathcal{C}} x_i + \sum_{C_i \in \mathcal{I}} r(C_i)x_i$. Given a $\mathcal{P} \subseteq \{C_1, \ldots, C_w\}$, we present ILP formulation of SSVE problem as follows:

$$\text{Maximize} \sum_{C_i \in \mathcal{C}} x_i + \sum_{C_i \in \mathcal{I}} r(C_i) x_i$$
$$\text{Subject to}$$
$$\sum x_i = k$$
$$x_i \in \{1, \ldots, |C_i|\} \text{ for all } i \; : \; C_i \in \mathcal{C} \cup \mathcal{I}$$

Solving the ILP: Lenstra [11] showed that the feasibility version of p-ILP is FPT with running time doubly exponential in p, where p is the number of variables. Later, Kannan [7] designed an algorithm for p-ILP running in time $p^{O(p)}$. In our algorithm, we need the optimization version of p-ILP rather than the feasibility version. We state the minimization version of p-ILP as presented by Fellows et al. [6].

p-Variable Integer Linear Programming Optimization (p-Opt-ILP): Let matrices $A \in Z^{m \times p}$, $b \in Z^{p \times 1}$ and $c \in Z^{1 \times p}$ be given. We want to find a vector $x \in Z^{p \times 1}$ that minimizes the objective function $c \cdot x$ and satisfies the m inequalities, that is, $A \cdot x \geq b$. The number of variables p is the parameter. Then they showed the following:

Lemma 1. [6] p-Opt-ILP can be solved using $O(p^{2.5p+o(p)} \cdot L \cdot log(MN))$ arithmetic operations and space polynomial in L. Here L is the number of bits in the input, N is the maximum absolute value any variable can take, and M is an upper bound on the absolute value of the minimum taken by the objective function.

In the formulation for SSVE problem, we have at most w variables. The value of objective function is bounded by n and the value of any variable in the integer linear programming is also bounded by n. The constraints can be represented using $O(w^2 \log n)$ bits. Lemma 1 implies that we can solve the problem with the guess $\mathcal{P}$ in FPT time. There are at most 2^w choices for $\mathcal{P}$, and the ILP formula for a guess can be solved in FPT time. Thus Theorem 3 holds.

6 FPT Algorithm Parameterized by Treewidth

This section presents an FPT algorithm using dynamic programming for the Small Set Vertex Expansion problem parameterized by treewidth. Given a graph $G = (V, E)$, an integer $k \leq \frac{n}{2}$ and its nice tree decomposition $(T, X_t : t \in V(T))$ of width at most $\texttt{tw}$, subproblems will be defined on $G_t = (V_t, E_t)$ where V_t is the union of all bags present in subtree of T rooted at t, including X_t and E_t is the set of edges e introduced in the subtree rooted at t. We define a colour function $f : X_t \mapsto \{0, 1, \hat{1}\}$ that assigns three different colours to the vertices of X_t. The meanings of three different colours are given below:

1 (black vertices): all black vertices are contained in set S whose vertex expansion $\Phi^V(S)$ we wish to calculate in G_t.
0 (white vertices): white vertices are adjacent to black vertices, these vertices are in the expansion $N(S)$ in G_t.
$\hat{1}$ (gray vertices): gray vertices are neither in S nor in $N(S)$.

Now we introduce some notations. Let $X \subseteq V$ and consider a colouring $f : X \mapsto \{1, 0, \hat{1}\}$. For $\alpha \in \{1, 0, \hat{1}\}$ and $v \in V(G)$ a new colouring $f_{v \mapsto \alpha} : X \cup \{v\} \mapsto \{1, 0, \hat{1}\}$ is defined as follows:

$$f_{v \mapsto \alpha}(x) = \begin{cases} f(x) & \text{when } x \neq v \\ \alpha & \text{when } x = v \end{cases}$$

Let f be a colouring of X, then the notation $f_{|_Y}$ is used to denote the restriction of f to Y, where $Y \subseteq X$.

For a colouring f of X_t and an integer i, a set $S \subseteq V_t$ is said to be compatible for tuple (t, f, i) if

1. $|S| = i$,
2. $S \cap X_t = f^{-1}\{1\}$ which is the set of vertices of X_t coloured black, and
3. $N(S) \cap X_t = f^{-1}\{0\}$, which is the set of vertices of X_t coloured white.

We call a set S a *minimum compatible set* for (t, f, i) if its vertex expansion $\Phi^V(S) = |N_{V_t}(S)|$ is minimum. We denote by $c[t, f, i]$ the minimum vertex expansion for (t, f, i), that is, $c[t, f, i]$ equals to $|N_{V_t}(S)|$, where S is a minimum compatible set for (t, f, i). If no such S exists, then we put $c[t, f, i] = \infty$ also $c[t, f, i < 0] = \infty$. Since each vertex in X_t can be coloured with 3 colours $(1, 0, \hat{1})$, the number of possible colouring f of X_t is $3^{|X_t|}$ and for each colouring f we vary i from 0 to k. The smallest value of vertex expansion $\Phi^V(S) = |N(S)|$ for a set S with k nodes will be $c[r, \phi, k]$, where r is the root node of tree decomposition of G as $G = G_r$ and $X_r = \emptyset$. We only show that $\Phi^V(S)$ can be computed in the claimed running time in Theorem 4. Corresponding set S can be easily computed in the same running time by remembering a corresponding set for each tuple (t, f, i) in the dynamic programming above. Now we present the recursive formulae for the values of $c[t, f, i]$.

Leaf Node: If t is a leaf node, then the corresponding bag X_t is empty. Hence the colour function f on X_t is an empty colouring; the number i of vertices coloured black cannot be greater than zero. Thus we have

$$c[t, \emptyset, i] = \begin{cases} 0 & \text{if } i = 0 \\ \infty & \text{otherwise} \end{cases}$$

Introduce Node: Suppose t is an introduce node with child t' such that $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$. The introduce node introduces the vertex v but does not introduce the edges incident to v in G_t. So when v is introduced by node t it is an isolated vertex in G_t. Vertex v cannot be coloured white 0; as it is isolated and it cannot be neighbour of any black vertex. Hence if $f(v) = 0$, then $c[t, f, i] = \infty$. When $f(v) = 1$, v is contained in S. As v is an isolated vertex, it does not contribute towards the size of $N_{V_t}(S)$, hence $c[t, f, i] = c[t', f_{|X_{t'}}, i-1]$. When $f(v) = \hat{1}$, v does not contribute towards the size of $N_{V_t}(S)$. Here the sets compatible for $(t', f_{|X_{t'}}, i)$ are also compatible for (t, f, i). So, $c[t, f, i] = c[t', f_{|X_{t'}}, i]$. Combining all the cases together, we get

$$c[t, f, i] = \begin{cases} \infty & \text{if } f(v) = 0 \\ c[t', f_{|X_{t'}}, i-1] & \text{if } f(v) = 1 \\ c[t', f_{|X_{t'}}, i] & \text{if } f(v) = \hat{1} \end{cases}$$

Introduce Edge Node: Let t be an introduce edge node that introduces the edge (u, v), let t' be the child of t. Thus $X_t = X_{t'}$; the edge (u, v) is not there in $G_{t'}$, but it is there in G_t. Let f be a colouring of X_t. We consider the following cases:

- Suppose $f(u) = 1$ and $f(v) = \hat{1}$. This means $u \in S$ and v is non-adjacent to black vertices in G_t. But u and v are adjacent in G_t. Thus $c[t, f, i] = \infty$. The same conclusion can be drawn when v is coloured black and u is coloured gray.
- Suppose $f(u) = 1$ and $f(v) = 0$. This means $u \in S$ and $v \in N(S)$ in G_t. In order to get a solution for (t, f, i), we consider two cases.
  Case 1: While considering precomputed solution for t' we can relax the colour of v from white to gray. Then the minimum vertex expansion for (t, f, i) is one more than the minimum vertex expansion for $(t', f_{v \to \hat{1}}, i)$, that is, $c[t, f, i] = 1 + c[t', f_{v \mapsto \hat{1}}, i]$.
  Case 2: While considering precomputed solution for t' we keep the colour of v be white. Then the minimum vertex expansion for (t, f, i) is equal to the minimum vertex expansion for (t', f, i), that is, $c[t, f, i] = c[t', f, i]$.
  Combining above two cases we get
  $$c[t, f, i] = \min\Big\{c[t', f, i], 1 + c[t', f_{v \mapsto \hat{1}}, i]\Big\}$$
  The same conclusion can be drawn when v is coloured black and u is coloured white.
- Other colour combinations of u and v do not affect the size of $N(S)$ or do not contradict the definition of campatability. So the compatible sets for (t', f, i) are also compatible for (t, f, i) and hence $c[t, f, i] = c[t', f, i]$.

Combining all the cases together, we get

$$c[t, f, i] = \begin{cases} \infty & \text{if } (f(u), f(v)) = (\hat{1}, 1) \\ \infty & \text{if } (f(u), f(v)) = (1, \hat{1}) \\ \min\{c[t', f, i], 1 + c[t', f_{v \mapsto \hat{1}}, i]\} & \text{if } (f(u), f(v)) = (1, 0) \\ \min\{c[t', f, i], 1 + c[t', f_{u \mapsto \hat{1}}, i]\} & \text{if } (f(u), f(v)) = (0, 1) \\ c[t', f, i] & \text{otherwise} \end{cases}$$

Forget Node: Let t be a forget node with the child t' such that $X_t = X_{t'} \setminus \{w\}$ for some vertex $w \in X_{t'}$. Here the bag X_t forgets the vertex w. At this stage we decides the final colour of the vertex w. We observe that $G_{t'} = G_t$. The compatible sets for $(t', f_{w\mapsto 1}, i)$, $(t', f_{w\mapsto 0}, i)$, $(t', f_{w\mapsto \hat{1}}, i)$ are also compatible for (t, f, i). On the other hand compatible sets for (t, f, i) are also compatible for $(t', f_{w\mapsto 1}, i)$ if $w \in S$, for $(t', f_{w\mapsto 0}, i)$ if $w \in N(S)$ or for $(t', f_{w\mapsto \hat{1}}, i)$ if $w \notin N[S]$. Hence

$$c[t, f, i] = \min \left\{ c[t', f_{w\mapsto 1}, i], c[t', f_{w\mapsto 0}, i], c[t', f_{w\mapsto \hat{1}}, i] \right\}$$

Join Node: Let t be a join node with children t_1 and t_2, such that $X_t = X_{t_1} = X_{t_2}$. Let f be a colouring of X_t. We say that colouring f_1 of X_{t_1} and f_2 of X_{t_2} are consistent for colouring f of X_t, if the following conditions are true for each $v \in X_t$:

1. $f(v) = 1$ if and only if $f_1(v) = f_2(v) = 1$,
2. $f(v) = \hat{1}$ if and only if $f_1(v) = f_2(v) = \hat{1}$,
3. $f(v) = 0$ if and only if $(f_1(v), f_2(v)) \in \{(0, \hat{1}), (\hat{1}, 0), (0, 0)\}$.

Let f be a colouring of X_t; f_1 and f_2 be two colouring of X_{t_1} and X_{t_2} respectively consistent with f. Suppose S_1 is a compatible set for (t_1, f_1, i_1) and S_2 is a compatible set for (t_2, f_2, i_2), where $|S_1| = i_1$ and $|S_2| = i_2$. Set $S = S_1 \cup S_2$, clearly $|S| = |S_1| + |S_2| - |f^{-1}\{1\}|$. It is easy to see that S is a compatible set for (t, f, i), where $i = i_1 + i_2 - |f^{-1}\{1\}|$. According to Condition 3 in the definition of consistent function, each $v \in X_t$ that is white in f, we make it white either in f_1, f_2 or in both f_1 and f_2. Consequently, we have the following recursive formula:

$$c[t, f, i] = \min_{f_1, f_2} \left\{ \min_{i_1, i_2 \,:\, i = i_1 + i_2 - |f^{-1}\{1\}|} \left\{ c[t_1, f_1, i_1] + c[t_2, f_2, i_2] - \alpha_{f_1, f_2} \right\} \right\},$$

where $\alpha_{f_1, f_2} = |\{v \in X_t \mid f_1(v) = f_2(v) = 0\}|$.

We now analyse the running time of the algorithm. We compute all entries $c[t, f, i]$ in a bottom-up manner. Clearly, the time needed to process each leaf node, introduce vertex node, introduce edge node or forget node is $3^{\mathtt{tw}+1} \cdot k^{O(1)}$ assuming that the entries for the children of t are already computed. The computation of $c[t, f, i]$ for join node takes more time and it can be done as follows. If a pair (f_1, f_2) is consistent with f, then for every $v \in X_t$, we have $(f(v), f_1(v), f_1(v)) \in \{(1, 1, 1), (\hat{1}, \hat{1}, \hat{1}), (0, 0, 0), (0, 0, \hat{1}), (0, \hat{1}, 0)\}$. Hence there are exactly $5^{|X_t|}$ triples of colouring (f, f_1, f_2) such that f_1 and f_2 are consistent with f, since for every vertex v, we have 5 possibilities for $(f(v), f_1(v), f_2(v))$. In order to compute $c(t, f, i)$, we iterate through all consistent pairs (f_1, f_2); then for each considered triple (f, f_1, f_2) we vary i_1 and i_2 from 0 to k such that $i = i_1 + i_2 - |f^{-1}\{1\}|$. As $|X_t| \leq \mathtt{tw} + 1$, the time needed to process each join node is $5^{\mathtt{tw}+1} k^{O(1)}$. Since we assume that the number of nodes in a nice tree decomposition is $O(n \cdot \mathtt{tw})$, we have the following theorem.

Theorem 4. *Given an n-vertex graph G and its nice tree decomposition of width at most* $\mathtt{tw}$*, the* Small Set Vertex Expansion *problem can be solved in $O(5^{\mathtt{tw}}n)$ time.*

7 Conclusion

In this work we proved that the Small Set Vertex Expansion problem is W[1]-hard when parameterized by k, the number of vertices in S; it is FPT when parameterized neighbourhood diversity; and the problem is FPT when parameterized by treewidth of the input graph. The parameterized complexity of the Small Set Vertex Expansion problem remains unsettle when parameterized by $k + \ell$, and when parameterized by other important structural graph parameters like clique-width, modular width and treedepth.

Acknowledgement. We are grateful to Dr. Kitty Meeks, University of Glasgow, for useful discussions and her comments on the proof of Theorem 2.

References

1. Arora, S., Ge, R.: New tools for graph coloring. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) APPROX/RANDOM -2011. LNCS, vol. 6845, pp. 1–12. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22935-0_1
2. Chlamtác, E., Dinitz, M., Makarychev, Y.: Minimizing the union: Tight approximations for small set bipartite vertex expansion. ArXiv, abs/1611.07866 (2017)
3. Chlamtáč, E., Dinitz, M., Konrad, C., Kortsarz, G., Rabanca, G.: The densest k-subhypergraph problem. SIAM J. Discrete Math. **32**(2), 1458–1477 (2018)
4. Cygan, M., et al.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3_15
5. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (1998)
6. Fellows, M.R., Lokshtanov, D., Misra, N., Rosamond, F.A., Saurabh, S.: Graph layout problems parameterized by vertex cover. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 294–305. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-92182-0_28
7. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
8. Khot, S.A., Vishnoi, N.K.: The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l/sub 1/. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), pp. 53–62 (2005)
9. Kloks, T. (ed.): Treewidth. LNCS, vol. 842. Springer, Heidelberg (1994). https://doi.org/10.1007/BFb0045375
10. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica **64**, 19–37 (2012)
11. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
12. Louis, A., Makarychev, Y.: Approximation algorithms for hypergraph small-set expansion and small-set vertex expansion. Theory Comput. **12**(17), 1–25 (2016)

13. Raghavendra, P., Steurer, D.: Graph expansion and the unique games conjecture. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing, STOC 2010, pp. 755–764. Association for Computing Machinery, New York (2010)
14. Robertson, N., Seymour, P.: Graph minors. iii. planar tree-width. J. Comb. Theory Ser. B **36**(1), 49–64 (1984)
15. Tedder, M., Corneil, D., Habib, M., Paul, C.: Simpler linear-time modular decomposition via recursive factorizing permutations. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008. LNCS, vol. 5125, pp. 634–645. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70575-8_52
16. Vinterbo, S.A.: A Note on the Hardness of the k-Ambiguity Problem. Technical report, Harvard Medical School, Boston, MA, USA, 06 2002