


Finding a Highly Connected Steiner Subgraph and its Applications

Eduard Eiben  

Royal Holloway, University of London, Egham, UK

Diptapriyo Majumdar¹  

Indraprastha Institute of Information Technology Delhi, New Delhi, India

M. S. Ramanujan  

University of Warwick, Coventry, UK

Abstract

Given a (connected) undirected graph G , a set $X \subseteq V(G)$ and integers k and p , the STEINER SUBGRAPH EXTENSION problem asks whether there exists a set $S \supseteq X$ of at most k vertices such that $G[S]$ is a p -edge-connected subgraph. This problem is a natural generalization of the well-studied STEINER TREE problem (set $p = 1$ and X to be the terminals). In this paper, we initiate the study of STEINER SUBGRAPH EXTENSION from the perspective of parameterized complexity and give a fixed-parameter algorithm (i.e., FPT algorithm) parameterized by k and p on graphs of bounded degeneracy (removing the assumption of bounded degeneracy results in W-hardness).

Besides being an independent advance on the parameterized complexity of network design problems, our result has natural applications. In particular, we use our result to obtain new single-exponential FPT algorithms for several vertex-deletion problems studied in the literature, where the goal is to delete a smallest set of vertices such that: (i) the resulting graph belongs to a specified hereditary graph class, and (ii) the deleted set of vertices induces a p -edge-connected subgraph of the input graph.

2012 ACM Subject Classification Theory of computation \rightarrow Parameterized complexity and exact algorithms; Theory of computation \rightarrow Dynamic programming

Keywords and phrases Parameterized Complexity, Steiner Subgraph Extension, p -edge-connected graphs, Matroids, Representative Families

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.45

Funding *M. S. Ramanujan*: Supported by Engineering and Physical Sciences Research Council (EPSRC) grants EP/V007793/1 and EP/V044621/1.

1 Introduction

Given a simple undirected graph $G = (V, E)$ and a set $T \subseteq V(G)$, called *terminals*, the STEINER TREE problem asks if there are at most k edges $F \subseteq E(G)$ such that there is a path between every pair of vertices of T in $G' = (V, F)$. STEINER TREE is one of the fundamental problems in network design and is a well-studied problem in parameterized complexity ([14, 9, 7, 3, 21]). We refer to Section 2 for definitions related to parameterized complexity and graph theory. In this paper, we study the STEINER SUBGRAPH EXTENSION problem, which is formally defined below.

STEINER SUBGRAPH EXTENSION

Input: A simple undirected graph $G = (V, E)$, $X \subseteq V(G)$ and integers $k, p \in \mathbb{N}$.

Parameter: $k + p$

Goal: Is there $S \supseteq X$ of size at most k such that $G[S]$ is p -edge-connected?

¹ Corresponding author



Observe that this is a natural generalization of STEINER TREE problem. To the best of our knowledge, the parameterized complexity status of STEINER SUBGRAPH EXTENSION is unexplored even for arbitrary fixed constant p . When $p = 2$, STEINER SUBGRAPH EXTENSION is closely related to a special variant of EDGE-CONNECTED SURVIVABLE NETWORK DESIGN (defined by Feldman et al. [15]) problem. The goal of EDGE-CONNECTED SURVIVABLE NETWORK DESIGN is to find a collection of “at most k edges” so that there are two edge-disjoint paths between every pair of vertices in the terminal set. Abhinav et al. [1] studied the above problem when $p = n - k$, with k as the parameter. Moreover, they aim to find an $(n - k)$ -edge-connected steiner subgraph with exactly ℓ vertices. In our problem, observe that $p \leq k - 1$ as any graph with k vertices can be $(k - 1)$ -edge-connected and not p -edge-connected for $p \geq k$. If we set $p = k - 1$, then our problem becomes precisely the CLIQUE problem, where we want to decide if a graph has a clique with exactly k vertices, a W[1]-hard problem. Hence, one must place further restrictions on the input when aiming for fixed-parameter tractability. In this paper, we consider the STEINER SUBGRAPH EXTENSION problem when η is the degeneracy of the input graph G and η is a fixed-constant. Note that many well-known sparse graph classes are subclasses of graphs of bounded degeneracy. For instance, planar graphs are 5-degenerate, graphs with treewidth (or treedepth or pathwidth) at most η are $(\eta + 1)$ -degenerate.

Our Contributions. The input to our problem STEINER SUBGRAPH EXTENSION is a simple undirected graph with n vertices and η is a fixed constant. Recall that the parameter is $k + p$. The first part of our paper proves that STEINER SUBGRAPH EXTENSION is FPT when the input graph has constant degeneracy. In particular, we give an FPT algorithm with running time $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$ -time for STEINER SUBGRAPH EXTENSION when the input graph G is η -degenerate. The formal statement of the theorem is given below.

► **Theorem 1.** STEINER SUBGRAPH EXTENSION can be solved in time $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$, where η is the degeneracy of the input graph.

In particular, on graphs of constant degeneracy and for constant p , the above result gives a $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ -time algorithm, which is useful in several applications as we show in this paper.

The above result crucially relies on the use of the *out-partition* matroid, its linear representability in deterministic polynomial-time, and a dynamic programming subroutine using the notion of representative sets. We would like to highlight that Einarson et al. [13] have studied the same problem when X is a vertex cover. Our dynamic programming algorithm over representative sets has some similarities with the algorithm of Einarson et al. [13] but X is not necessarily a vertex cover of G for STEINER SUBGRAPH EXTENSION. Despite the fact that G is a bounded degenerate graph, designing an algorithm for STEINER SUBGRAPH EXTENSION needs careful adjustment to the subproblem definitions and some additional conditions have to be incorporated while constructing the collection of sets in the DP formulation. Furthermore, our algorithm in Theorem 1 does not depend on η in the exponent of n .

The second part of our paper describes some applications of our main result (Theorem 1) to some natural problems in graph theory with connectivity constraints. Einarson et al. [13] have initiated the study of p -EDGE-CONNECTED VERTEX COVER with stronger connectivity constraints. Being motivated by their results, we illustrate how Theorem 1 lays us a foundation to *design efficient deterministic parameterized singly exponential-time algorithms* for BOUNDED DEGREE DELETION SET, η -TREEDPTH DELETION SET, PATHWIDTH-ONE DELETION SET and η -PATH VERTEX COVER with p -edge-connectivity constraints. Each of

these problems are well-studied without the connectivity constraints (see [18, 5, 6] for more details). We state the problem definitions below. Given an undirected graph $G = (V, E)$, the following questions are asked by these problems.

- p -EDGE-CONNECTED η -DEGREE DELETION SET (p -EDGE-CON-BDDS) asks if there is $S \subseteq V(G)$ such that $G - S$ is a graph of maximum degree at most η and $G[S]$ is p -edge-connected.
- p -EDGE-CONNECTED η -TREEDEPTH DELETION SET (p -EDGE-CON- η -TDDS) asks if there is $S \subseteq V(G)$ such that $G - S$ has treedepth at most η and $G[S]$ is p -edge-connected.
- p -EDGE CONNECTED PATHWIDTH-1 VERTEX DELETION (p -EDGE-CON-PW1DS) asks if there is $S \subseteq V(G)$ such that $G - S$ has pathwidth at most 1 and $G[S]$ is p -edge-connected.
- p -EDGE-CONNECTED η -PATH VERTEX COVER (p -EDGE-CON- η -PVC) asks if there is $S \subseteq V(G)$ such that $G - S$ has no P_η as subgraph and $G[S]$ is p -edge-connected.

Applications to each of the above mentioned problems crucially rely on a property. The property is that all minimal vertex-deletion sets that must be part of any optimal solution can be enumerated in $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ -time for some fixed constant η . Since a graph of maximum degree η is also an η -degenerate graph, we have the following result as a direct application of our main result.

► **Corollary 2.** p -EDGE-CON-BDDS admits a $2^{\mathcal{O}(pk+k\eta)}n^{\mathcal{O}(1)}$ -time algorithm.

Our second application is p -EDGE-CON-PW1DS problem. The graphs of pathwidth at most one are also 2-degenerate. But it is not straightforward to enumerate all the minimal pathwidth one vertex deletion sets. So we use some additional characterizations of graphs of pathwidth one and exploit some problem specific structures to prove our next result.

► **Theorem 3.** p -EDGE-CON-PW1DS admits an algorithm that runs in $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time.

Note that the algorithm for the above result does not directly invoke the subroutine from Theorem 1. Instead, it uses some dynamic programming ideas that are closely similar to that of Theorem 1 proof but also makes careful local adjustments to take care of some additional constraints. Finally, our last two applications are p -EDGE-CONNECTED η -TREEDEPTH DELETION SET and p -EDGE-CON- η -PVC problems and we have the following two results.

► **Theorem 4.** p -EDGE-CON- η -TDDS admits an algorithm that runs in $2^{2^{2^\eta} + \mathcal{O}((p+\eta)k)}n^{2^{2^\eta}}$ -time.

► **Theorem 5.** p -EDGE-CON- η -PVC admits an algorithm that runs in $2^{\mathcal{O}((p+\eta)k)}n^{\mathcal{O}(1)}$ -time.

Organization of our paper. We organize the paper as follows. Initially in Section 2, we introduce the basic notations related to graph theory, parameterized complexity and matroids. Then, in Section 3, we prove our main result, i.e. Theorem 1. Then, in Section 4, we illustrate the applications of our main result to design singly exponential-time algorithms for p -EDGE-CON-BDDS, p -EDGE-CON-PW1DS, p -EDGE-CON- η -TDDS and p -EDGE-CON- η -PVC.

Related Work. Heggernes et al. [19] studied the parameterized complexity of p -CONNECTED STEINER SUBGRAPH that is the vertex-connectivity counterpart of our problem. The authors in their paper have proved that when parameterized by k , the above mentioned problem is FPT for $p = 2$ and W[1]-hard when $p = 3$. Nutov [23] has studied a variant of p -CONNECTED STEINER SUBGRAPH problem in which they have studied VERTEX CONNECTIVITY AUGMENTATION problem. Given an undirected graph G , a p -connected subgraph $G[S]$, the

VERTEX CONNECTIVITY AUGMENTATION problem asks if at most k additional edges can be added to $G[S]$ to make the subgraph $(p+1)$ -connected. In particular, Nutov [23] provided a parameterized algorithm for the above mentioned problem. Feldman et al. [15] have studied parameterized complexity of VERTEX/EDGE-CONNECTED SURVIVABLE NETWORK DESIGN PROBLEM where given fixed constant p , they want to compute a subgraph that has minimum number of edges and provides p -vertex/edge-connectivity between every pair of vertices in the terminals.

2 Preliminaries

Sets, numbers and graph theory. We use \mathbb{N} to denote the set of all natural numbers. For $r \in \mathbb{N}$, we use $[r]$ to denote the set $\{1, \dots, r\}$. Given a set S and an integer k , we use $\binom{S}{\leq k}$ and $\binom{S}{k}$ to denote the collection of all subsets of S of size at most k and of size exactly k respectively. We use standard graph theoretic notations from Diestel's book [10] for all notations of undirected and directed graphs. For undirected graphs, we use $uv \in E(G)$ to denote that there is an edge between u and v . On the other hand for directed graphs, we are more explicit. We use (u, v) to represent that the edge is directed from u to v . For the directed graphs, the directed edges are also called arcs. We use the term arc and edge interchangeably. In an undirected graph G , we use $\deg_G(v)$ to denote the degree of v in G . When the graph is clear from the context, we drop this subscript and simply use $\deg(v)$. An undirected graph G is called a *degree- η -graph* if every vertex of G has degree at most η . We use $\Delta(G)$ to denote the $\max\{\deg_G(v) : v \in V(G)\}$, i.e. the maximum degree of any vertex in G . It is clear from the definition that if a graph G is a degree- η -graph then $\Delta(G) \leq \eta$. When we consider directed graphs, we have *in-degree* and *out-degree* for all the vertices. For a vertex v , the *in-degree* of v is the number of arcs of the form $(v, u) \in A$ and the *out-degree* of v is the number of arcs of the form $(u, v) \in A$. A connected undirected graph $G = (V, E)$ is said to be *p -edge-connected* if at least two vertices and $G - Y$ remains connected after deleting at most $p - 1$ edges. Due to the Menger's Theorem, a connected graph is said to be *p -edge-connected* if and only if there are p edge-disjoint paths between every pair of vertices. Given an undirected graph, a set $S \subseteq V(G)$ is said to be a *p -segment* of G if for every $u, v \in S$, there are p edge-disjoint paths from u to v in G . An undirected graph is said to be an *η -degenerate graph* if every subgraph has a vertex of degree at most η . Given an undirected η -degenerate graph $G = (V, E)$, a sequence of vertices $\rho_G = (v_1, \dots, v_n)$ is said to be an *η -degeneracy sequence* if for every $2 \leq i \leq n$, v_i has at most η neighbors from the vertices $\{v_1, \dots, v_{i-1}\}$. For an $\ell \in \mathbb{N}$, we use P_ℓ to denote a path containing ℓ vertices and C_ℓ to denote a cycle containing ℓ vertices. A graph is said to be a *degree- η -graph* if every vertex has degree at most η . It follows from the definition that every degree- η -graph is an η -degenerate graph, but the converse does not hold true. An undirected graph is said to be a *caterpillar graph* if every connected component is an induced path with hairs attached to each of its pendant vertices. Given a directed graph $D = (V, A)$, we define an *outbranching* of D rooted at $v \in V(D)$ is a subset $A' \subseteq A$ such that v has in-degree 0 and every other vertex has in-degree exactly one in $D' = (V, A')$.

We define the following two graph parameters treedepth and pathwidth that we use in our paper.

► **Definition 6 (Treedepth).** Given an undirected graph $G = (V, E)$, $\text{td}(G)$, i.e. the treedepth of G is defined as follows. If $|V(G)| = 1$, then $\text{td}(G) = 1$. If G is connected, then $\text{td}(G) = 1 + \min_{u \in V(G)} \text{td}(G - \{u\})$. Finally, if G_1, \dots, G_r are the connected components of G , then $\text{td}(G) = \max_{i=1}^r \text{td}(G_i)$.

Informally, a *treedepth decomposition* of an undirected graph G can be considered as a rooted forest Y with vertex set V such that for each $uv \in E(G)$, either u is an ancestor of v or v is an ancestor of u in Y . The context of treedepth is also sometimes referred to as *elimination tree* of G . It follows from the (recursive) definition above that treedepth of a graph is referred to as the minimum depth of a treedepth decomposition of G , where depth is defined as the maximum number of vertices in a root to leaf path.

► **Definition 7 (Path Decomposition).** A path decomposition of an undirected graph $G = (V, E)$ is a sequence (X_1, \dots, X_r) of bags $X_i \subseteq V(G)$ such that **(i)** every vertex belongs to at least one bag, **(ii)** for every edge $uv \in E(G)$, there is X_i such that $u, v \in X_i$, and **(iii)** for every vertex v , the bags containing v forms a contiguous subsequence, i.e. $(X_i, X_{i+1}, \dots, X_j)$. The width of a decomposition is $\max_{i \in [r]} |X_i| - 1$.

The *pathwidth* of a graph is defined as the smallest number η such that there exists a path decomposition of width η . Informally, pathwidth is a measure how much a graph is close to a path (or a linear forest). We use $\text{pw}(G)$ to denote the pathwidth of G .

Given a class of graphs \mathcal{G} , we say that \mathcal{G} is *polynomial-time recognizable*, if given a graph G , there is a polynomial-time algorithm that can correctly check if $G \in \mathcal{G}$. A graph class is said to be *hereditary* if it is closed under induced subgraphs.

Parameterized Complexity and W-hardness. A parameterized problem L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ . An instance of a parameterized problem is a pair (x, k) where $x \in \Sigma^*$ is the *input* and k is the *parameter*. A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is said to be *fixed-parameter tractable* if there exists an algorithm \mathcal{A} that given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm \mathcal{A} runs in $f(k)|x|^c$ -time for some constant c independent of n and k and correctly decides L . The algorithm \mathcal{A} that runs in $f(k)|x|^{\mathcal{O}(1)}$ -time is called a *fixed-parameter algorithm* (or FPT algorithm). A fixed-parameter algorithm is said to be a *singly exponential FPT algorithm* if it runs in $c^k|x|^{\mathcal{O}(1)}$ -time for some fixed constant c independent of $|x|$ and k . There is a hardness theory in parameterized complexity that is associated with the notion of *parameterized reduction* and the hierarchy of parameterized complexity classes. Broadly, the *W-hierarchy* (of parameterized complexity classes) is denoted by $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{XP}$. Given two distinct parameterized problems L_1 and L_2 , there is a *parameterized reduction* from L_1 to L_2 if given an instance (x, k) of L_1 , an algorithm \mathcal{A} runs in $g(k)|x|^{\mathcal{O}(1)}$ -time and outputs an equivalent instance (x', k') of L_2 such that $k' = f(k)$ for some function depending only on k . For more details on parameterized complexity and its associated hardness theory, we refer to the books [8, 22, 11].

Matroids and Representative Families. We use the following definitions and results related to matroid theory to design our algorithms.

► **Definition 8.** Given a universe U and a subfamily $\mathcal{I} \subseteq 2^U$, a set system $\mathcal{M} = (U, \mathcal{I})$ is said to be a matroid if **(i)** $\emptyset \in \mathcal{I}$, **(ii)** if $A \in \mathcal{I}$, then for all $A' \subseteq A$, $A' \in \mathcal{I}$ (*hereditary property*), and **(iii)** if there exists $A, B \in \mathcal{I}$ such that $|B| > |A|$, then there is $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$ (*exchange property*). The set U is called ground set of \mathcal{M} and a set $A \in \mathcal{I}$ is called an independent set of matroid \mathcal{M} .

It follows from the definition that all maximal independent sets are of the same size. A maximal independent set is called a *basis*. Let U be a universe with n elements and $\mathcal{I} = \binom{U}{\leq r}$. The set system (U, \mathcal{I}) is called a *uniform matroid*. Let $G = (V, E)$ be an undirected graph and $\mathcal{I} = \{F \subseteq E(G) \mid G' = (V, F) \text{ is a forest}\}$. The set system $(E(G), \mathcal{I})$ is called a *graphic matroid*. Let U be partitioned as $U_1 \uplus \dots \uplus U_r$ and $\mathcal{I} = \{A \subseteq U : |A \cap U_i| \leq 1 \text{ for all } i \in [r]\}$.

We say that (U, \mathcal{I}) is a *partition matroid*. A matroid M is said to be *representable over a field* \mathbb{F} if there is a matrix A over \mathbb{F} and a bijection $f : U \rightarrow \text{Col}(A)$ where $\text{Col}(A)$ is the set of columns of A such that $B \subseteq U$ is an independent set of U if and only if the set of columns $\{f(b) \mid b \in B\}$ are linearly independent. A matroid representable over a field \mathbb{F} is called a *linear matroid*.

Given two matroids $M_1 = (U_1, \mathcal{I}_1)$ and $M_2 = (U_2, \mathcal{I}_2)$, the *direct sum* $M = M_1 \oplus M_2$ is the matroid $(U_1 \uplus U_2, \mathcal{I})$ such that $I \in \mathcal{I}$ if and only if $I \cap U_1 \in \mathcal{I}_1$ and $I \cap U_2 \in \mathcal{I}_2$. If M_1 and M_2 are represented by matrices A_1 and A_2 respectively then $M = M_1 \oplus M_2$ also admits a matrix representation.

Given a matroid M , a *truncation* of M to rank r is the matroid $M' = (U, \mathcal{I}')$ where a set $A \subseteq U$ is independent in M' if and only if $A \in \mathcal{I}$ and $|A| \leq r$. Given a matroid M with its representation (in matrix-form), the truncation of M can be computed in polynomial-time. Let $M = (U, \mathcal{I})$ be a matroid and $X, Y \subseteq U$. We say that X *extends* Y in M if $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$. Moreover, let $\mathcal{S} \subseteq 2^U$ be a family. A subfamily $\hat{\mathcal{S}} \subseteq \mathcal{S}$ is a *q-representative* of \mathcal{S} if the following holds: for every set $Y \subseteq U$ with $|Y| \leq q$, there is a set $X \in \mathcal{S}$ such that X extends Y if and only if there is a set $\hat{X} \in \hat{\mathcal{S}}$ such that \hat{X} extends Y . We use $\hat{\mathcal{S}} \subseteq_{\text{rep}}^q \mathcal{S}$ to denote that $\hat{\mathcal{S}}$ is a *q-representative family* of \mathcal{S} . The following result holds true due to Fomin et al. [17, 20].

► **Proposition 9.** *Let $M = (U, \mathcal{I})$ be a linear matroid of rank n and $p, q \leq n$ over a field \mathbb{F} and let $\mathcal{S} = \{S_1, \dots, S_t\} \subseteq \mathcal{I}$ each having cardinality p . Then, there exists an algorithm that computes a *q-representative subfamily* $\hat{\mathcal{S}} \subseteq_{\text{rep}}^q \mathcal{S}$ consisting of at most $\binom{p+q}{q}$ sets using $\mathcal{O}\left(\binom{p+q}{p}^2 tp^3 n^2 + t \binom{p+q}{q}^\omega np\right) + (n + |U|)^{\mathcal{O}(1)}$ field operations over \mathbb{F} . Here $\omega < 2.37$ is the matrix multiplication exponent.*

Let $G = (V, E)$ be an undirected graph and $D_G = (V, A_E)$ is defined as follows. For every $uv \in E(G)$, we add (u, v) and (v, u) into A_E and fix $v_r \in V$. Since the definition of D_G is based on $G = (V, E)$, we call the pair (D_G, v_r) an *equivalent digraph of G with root v_r* . Then, an *out-partition matroid with root v_r* for D_G is the partition matroid with ground set A_E where arcs are partitioned according to their heads and arcs (u, v_r) are dependent. Equivalently, what it means is that a set of arcs $F \subseteq A_E$ is an *independent* in the out-partition matroid with root v_r if and only if v_r has in-degree 0 in F and every other vertex has in-degree at most 1 in F . The graphic matroid in the ground set A_E is the graphic matroid for G where every arc is represented by its underlying undirected edge and the antiparallel arcs $(u, v), (v, u)$ represent distinct copies of uv . Then, $\{(u, v), (v, u)\}$ becomes a dependent set. The following two propositions are proved by Agrawal et al. [2] and Einarson et al. [13] respectively.

► **Proposition 10** (Agrawal et al. [2]). *Let $G = (V, E)$ be an undirected graph, $v_r \in V$ and $D_G = (V, A_E)$ as defined above. Then, G is p -edge-connected if and only if D_G has pairwise arc-disjoint out-branchings rooted at v_r .*

► **Proposition 11** (Einarson et al. [13]). *Let $G = (V, E)$ be an undirected graph, $v_r \in V$ and $D_G = (V, A_E)$ as defined above. Then, F is the arc set of an out-branching rooted at v_r if and only if $|F| = |V(G)| - 1$ and F is independent both in the out-partition matroid for D_G with root v_r and the graphic matroid for G with ground set A_E .*

Let G be an undirected graph, $X \subseteq V(G)$ and p be an integer. It is not immediate whether in polynomial-time we can check whether there exists a *feasible* solution, that is, a set $S \supseteq X$ such that $G[S]$ is p -edge-connected. The following lemma illustrates that the above can be achieved in polynomial-time. In fact, in this case the input graph does not have to be a bounded degenerate graph.

► **Lemma 12** (\star).² Let $G = (V, E)$ be a connected undirected graph and $X \subseteq V(G)$. Then, we can check in polynomial-time if there exists a set $S \supseteq X$ such that $G[S]$ is a p -edge-connected subgraph.

We refer to Oxley [24] for more details on matroid theory and a survey by Panolan and Saurabh [25] for more information on use of matroids in FPT algorithms.

3 Algorithm for STEINER SUBGRAPH EXTENSION

This section is devoted to the proof of the main contribution of our paper. We first provide a singly exponential algorithm for STEINER SUBGRAPH EXTENSION (we restate below) when the input graph has bounded degeneracy. We assume that a fixed constant η is the degeneracy of G . We restate the problem definition.

STEINER SUBGRAPH EXTENSION

Input: An undirected graph $G = (V, E)$, $X \subseteq V(G)$ and integers $k, p \in \mathbb{N}$.

Parameter: $k + p$

Goal: Is there $S \supseteq X$ of size at most k such that $G[S]$ is p -edge-connected?

Let (G, X, k) be given as an input instance and σ be a degeneracy sequence for the vertices of $G - X$ witnessing that $G - X$ is also an η -degenerate graph. Note that one can compute a degeneracy sequence easily in polynomial time by iteratively picking the minimum degree vertex, hence, we assume an ordering $\sigma = (u_1, \dots, u_n)$ of the vertices of $G - X$ is given along with the input. Due to Lemma 12, we can check if there exists a feasible solution $S \supseteq X$ (not necessarily of size at most k) such that $G[S]$ is p -edge-connected subgraph. So, we can assume without loss of generality that a feasible solution actually exists. We first state a proposition that we use later in the proof of our result.

► **Proposition 13** ([13]). Let $G = (V, E)$ be an undirected graph, $v_r \in V$, and $D_G = (V, A_E)$ such that (D_G, v_r) is an equivalent digraph with root v_r . We also assume that M is a direct sum $M_1 \oplus \dots \oplus M_{2p+1}$ as follows. The matroids $M_1, M_3, \dots, M_{2p-1}$ are the copies of graphic matroid of G on ground set E , the matroids M_2, M_4, \dots, M_{2p} are the copies of out-partition matroids with ground set A_E with root v_r , and the matroid M_{2p+1} is a uniform matroid over A_E with rank $p(k-1)$. Furthermore, suppose that $F \subseteq A_E$, then the followings two are equivalent.

- (i) F is the set of p pairwise arc-disjoint out-branchings rooted at v_r in $D_G[S]$ for some $S \in \binom{V(G)}{k}$ and $v_r \in S$.
- (ii) $|V(F)| = k$, $|F| = p(k-1)$, $v_r \in V(F)$, and there is an independent set I in M such that every arc $a \in F$ occurs in I precisely in its copies in matroids M_{2i-1}, M_{2i} and M_{2p+1} for some $i \in \{1, \dots, p\}$.

In addition, a linear representation of M and the truncation of M to \hat{M} of rank $3p(k-1)$ can be computed in deterministic polynomial-time.

Our algorithm for STEINER SUBGRAPH EXTENSION works as follows. There are two cases. If $X = \emptyset$, then we choose an arbitrary vertex $u \in V(G)$ and set $X = \{u\}$. There are $n = |V(G)|$ possible choices of X . For each such choice we assign $v_r = u \in X$. Otherwise, it is already the case that $X \neq \emptyset$. Therefore, we can assume without loss of generality that $X \neq \emptyset$.

² Due to lack of space, the lemmas marked \star and the other omitted proofs can be found in the full version.

On the other hand, if $X \neq \emptyset$ and $G[X]$ is p -edge-connected, then we can trivially output yes-instance since $|X| \leq k$. So, we are in the situation that $G[X]$ is not p -edge-connected and $|X| < k$. In the algorithm, we use the above characterization and representative sets framework to check if X can be extended to a p -edge-connected subgraph $G[S]$ with at most k vertices. We fix an arbitrary vertex $v_r \in X$. Due to Proposition 13, there is an independent set I such that **(i)** $X \subset V(I)$ and $|V(I)| = k$, **(ii)** $|I| = 3p(k-1)$, and **(iii)** every arc that is in I is represented precisely in three matroids M_{2i-1}, M_{2i} and M_{2p+1} for some $i \in [p]$.

We will build the set I via dynamic programming procedure. Since X is already included in $V(I)$ it allows us to replace the first condition with $|V(I) \setminus X| = k - |X|$. The purpose of this dynamic programming is to construct a table that keeps track of $|V(I) \setminus X|$ and $|I|$. Let $\{v_1, \dots, v_{n'}\}$ be a degeneracy sequence of the vertices of $G - X$ and $\mathcal{X} = \{A_i \mid A_i = N(v_i) \cap \{v_1, \dots, v_{i-1}\} \setminus X\}$. Each entry of the dynamic programming table $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ will contain a collection of independent sets of M that is a $(3p(k-1) - q)$ -representative family of all the independent sets I of M such that $|V(I) \setminus X| = i$, $|I| = q$, $Y = A_j \cap V(I)$, the largest index of $V(G) - X$ that occurs in $V(I)$ is j and $Z = A_\ell \cap V(I)$ for some $\ell > j$. Informally, it means that every independent set I has size q in M , $V(I)$ intersects A_j exactly in Y , and $V(I)$ spans i vertices from $G - X$, $v_j \in V(I)$, and $V(I)$ has no vertex from $\{v_{j+1}, \dots, v_{n'}\}$. Furthermore, for every $1 \leq j < n'$, $V(I)$ intersects A_ℓ exactly in Z for some $\ell > j$. Observe that for $j = n'$, there is no index $\ell > j$. Then we denote $\ell = n' + 1$ and $Z = \emptyset$ to keep the DP-states well-defined. We prove the following lemma that illustrates how a dynamic programming algorithm can construct all the entries of a table $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ for $i \leq k - |X|, j \leq n', q \leq 3p(k-1), Y \subseteq A_j, Z \subseteq A_\ell$ and $\ell > j$. Indeed, if $j = n'$, then $\ell = n' + 1$ and $Z = \emptyset$. Observe that there are at most $2^n n$ possible choices of Y and $2^n n$ possible choices for (Z, ℓ) in the DP table \mathcal{T} . The following lemma illustrates how we compute the DP-table entries.

► **Lemma 14.** *Given matroid M of rank $r = 3p(k-1)$ as described above, the entries of the table $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ for $i \leq k - |X|, j \leq n', j < \ell, q \leq 3p(k-1), Y \subseteq A_j$ and $Z \subseteq A_\ell$ can be computed in $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$ -time.*

Proof. We describe a procedure $\text{Construct}(\mathcal{T}[(i, j, q, Y), (Z, \ell)])$ for $i \leq k - |X|, j \leq n'$ and $q \leq 3p(k-1)$ as follows. Observe that every arc of I occurs in three copies, one in M_{2i-1} , one in M_{2i} and the other in M_{2p+1} . Given an arc $a \in A_E$, we use $F_{a,i}$ to denote the set that contains the copies of a in M_{2i-1}, M_{2i} and M_{2p+1} . In the first part, we describe constructing the table entries $\mathcal{T}[(0, 0, q, Y), (Z, \ell)]$ as follows.

- **(i)** For all $1 \leq \ell \leq n'$, we initialize $\mathcal{T}[(0, 0, 0, \emptyset), (\emptyset, \ell)] = \{\emptyset\}$.
- **(ii)** Consider the set of all the arcs in $D_G[X]$. We construct $\mathcal{T}[(0, 0, q+3, \emptyset), (\emptyset, \ell)]$ from $\mathcal{T}[(0, 0, q, \emptyset), (\emptyset, \ell)]$ as follows. For every $I \in \mathcal{T}[(0, 0, q, \emptyset), (\emptyset, \ell)]$, for every arc $a \in D_G[X]$, ($1 \leq j \leq m$), and $i \in \{1, \dots, p\}$, we add $I \cup F_{a,i}$ such that $F_{a,i}$ extends I .
- **(iii)** Finally, we invoke Proposition 9 to reduce $\mathcal{T}[(0, 0, q+3, \emptyset), (\emptyset, \ell)]$ into a $(3p(k-1) - q - 3)$ -representative family of size $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$.

When we consider the table entries $\mathcal{T}[(i, j, q+3, Y), (Z, \ell)]$ such that $j = 0$, observe that $Y = Z = \emptyset$. The reason is that for any $I \in \mathcal{T}[(i, j, q+3, Y), (Z, \ell)]$ with the assumption of $j = 0$ implies that $i = 0$ and no vertex from $G - X$ is part of $V(I)$. Since $Y, Z \subseteq V(G) \setminus X$, it must be that $Y = Z = \emptyset$. So, we consider only those entries in this first phase.

We analyze the running-time of the above process. Given $\mathcal{T}[(0, 0, q, \emptyset), (\emptyset, \ell)]$, computing $\mathcal{T}[(0, 0, q+3, \emptyset), (\emptyset, \ell)]$ requires polynomial in the size of $|\mathcal{T}[(0, 0, q, \emptyset), (\emptyset, \ell)]|$. Then, computing a $(3p(k-1) - q - 3)$ -representative family requires $2^{\mathcal{O}(pk)}$ -time.

The process of computing table entries for $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ for $Y \subseteq A_j$ and $Z \subseteq A_\ell$ for $\ell > j$ is more complex and needs more careful approach. We consider a lexicographic ordering of the indices $((i, j, q, Y), (Z, \ell))$ and consider one by one as follows. For every

$i \geq 1$, we compute the collection of independent sets in $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ using the collection of independent sets of $\mathcal{T}[(i', j', q', Y'), (Z', \ell')]$ for $j' < j$, $j' < \ell'$, $i = i' + 1$, and $q' < q$ as follows. We want to include v_j into $V(I)$ such that $|V(I) \setminus X| = i$. We look at all the previous subproblems in the lexicographic ordering one by one. Suppose that $I' \in \mathcal{T}[(i', j', q', Y'), (Z', \ell')]$. By definition, it holds true that $Y' = V(I') \cap A_{j'}$ and $Z' = A_{\ell'} \cap V(I')$. We say that I' is *extendable* to a set $I \in \mathcal{T}[(i, j, q, Y), (Z, \ell)]$ if $i = i' + 1$, $Z' = Y$, and $\ell' = j$. Observe that by definition $V(I)$ and $V(I')$ both intersect A_j exactly in Y , i.e. $V(I) \cap A_j = Y$ and $V(I') \cap A_j = Y$. Moreover, j is the next larger index vertex included in $V(I)$ after $v_{j'}$.

We construct the table entries of $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ using the table entries of $\mathcal{T}[(i', j', q', Y'), (Z', \ell')]$ as follows.

Let d be the number of arcs that are incident to v_j such that either the other endpoints have lower indices or the other endpoint is in X . Formally, we consider the arcs that are either of the form $(v_{j'}, v_j), (v_j, v_{j'}) \in A_E$ such that $j' < j$ or of the form $(u, v_j), (v_j, u)$ for some $u \in X$. Since G is an η -degenerate graph, observe that $d \leq 2\eta + 2|X| \leq 2(k + \eta)$. We create a set F of arcs as follows. For every arc a incident to v_j , we consider the sets $F_{a,h}$ for every $h \in [p]$. We either add $F_{a,h}$ into F for some $h \in [p]$, or do not add $F_{a,h}$ into F . This ensures that there are $(p+1)^d \leq (p+1)^{2\eta+2k}$ possible collections of arc-sets. For every nonempty such arc-set F and for every $I' \in \mathcal{T}[(i', j', q', Y'), (Z', \ell)]$ satisfying $Y = Z'$, $i = i' + 1$ and $\ell' = j$ (in other words I' that is extendable to a set in $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$), we add $I' \cup F$ into $\mathcal{T}[(i, j, q + |F|, Y), (Z, \ell)]$ when F extends I' and $q = q' + |F|$. Finally, invoke Proposition 9 to reduce $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ into its $(3p(k-1) - q)$ -representative family containing at most $2^{3p(k-1)}$ sets.

Observe that for every set $I' \in \mathcal{T}[(i', j', q', Y'), (Z', \ell')]$, there are at most $(p+1)^{2\eta}$ sets $I = I' \cup F$ that are added to the slot $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$. Hence we have that

$$|\mathcal{T}[(i, j, q, Y), (Z, \ell)]| \leq (p+1)^{2|X|+2\eta} 2^{3pk} (p+k+\eta)^{\mathcal{O}(1)}$$

Since the number of indices is $4^\eta n^2$, the above implies that computing all the table entries can be performed in $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$ -time. This completes the proof of the lemma. \blacktriangleleft

Our next lemma ensures that the vertices of any independent set of size $3p(k-1)$ computed by the above lemma induces a p -edge-connected subgraph.

► Lemma 15. *There is a set $I \in \mathcal{T}[(k - |X|, j, 3p(k-1), Y), (Z, \ell)]$ for some $Y \subseteq A_j$ and $Z \subseteq A_\ell$ if and only if there exists $S \supset X$ such that $G[S]$ is a p -edge-connected subgraph of G with k vertices.*

Proof (Sketch).³ First part of the proof is forward direction (\Rightarrow). Let $I \in \mathcal{T}[(k - |X|, j, 3p(k-1), Y), (Z, \ell)]$ be an independent set for some $j \in [n']$, $Y = A_j \cap V(I)$ and $Z = V(I) \cap A_\ell$. Let $S = V(I)$ and it follows that $|S \setminus X| = k - |X|$. Note that $|I| = 3p(k-1)$. Due to Proposition 13, every arc of D_G occurs precisely in three copies. Recall that I was constructed by adding arcs $F_{a,i}$ for some $i \in [p]$ when the arc a incident to $v \in V(G) \setminus X$ was added. But, $F_{a,i}$ contains a copy of arc a in M_{2p+1} . Therefore, the construction of I ensures that no two distinct sets $F_{a,i}$ and $F_{a,i'}$ both are added. Hence $D_G[V(I)]$ has p pairwise arc-disjoint out-branchings rooted at v_r and it follows from Proposition 10 that $G[S]$ is p -edge-connected.

The other part of the proof is the backward direction (\Leftarrow). Let $G[S]$ be p -edge-connected with $|S| = k$, such that $X \subseteq S$ and v_j is the vertex with highest index from $\{v_1, \dots, v_n\} = V(G) \setminus X$. It follows from Proposition 10 that there are p pairwise arc-disjoint out-branchings

³ A complete detailed proof can be found in the full version.

45:10 Finding a Highly Connected Steiner Subgraph and its Applications

in $D_G[S]$ rooted at v_r . Then, by Proposition 13, there is an independent set I such that $V(I) = S$ containing the the set of arcs $F \subseteq A_E$ such that $|F| = p(k-1)$ and $|I| = 3p(k-1)$. We complete our proof by justifying that I is a candidate independent set of M for the slot $\mathcal{T}[(k-|X|, j, 3p(k-1), Y), (Z, \ell)]$ for some $Z = V(I) \cap A_\ell$ and $\ell > j$. If we prove that there is $Y \subseteq N(v_j) \cap \{v_1, \dots, v_{j-1}\}$ such that $\mathcal{T}[(k-|X|, j, 3p(k-1), Y), (Z, \ell)] \neq \emptyset$, then we are done. We can give a proof of this claim by induction on i, j, q that $\mathcal{T}[(k-|X|, j, 3p(k-1), Y), (Z, \ell)] \neq \emptyset$. ◀

Using Lemma 14 and Lemma 15, we are ready to prove our theorem statement of our main result, i.e. Theorem 1 (we restate below).

► **Theorem 1.** STEINER SUBGRAPH EXTENSION can be solved in time $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$, where η is the degeneracy of the input graph.

Proof. We assume without loss of generality that $X \neq \emptyset$ and $G[X]$ is not p -edge-connected. Also, we use (G, X, k, p) to denote the input instance. Let $\{v_1, \dots, v_{n'}\}$ be a degeneracy sequence of the vertices of $G - X$ and consider an arbitrary vertex $v_r \in X$. The first step is to invoke Proposition 13 and construct a matroid M . We can also assume without loss of generality that $(G, X, k-1, p)$ is a no-instance. Our next step is to invoke Lemma 14 and compute the table entries $\mathcal{T}[(i, j, q, Y), (Z, \ell)]$ for all $i \in \{1, \dots, k-|X|\}$, $j \in [n']$, $q \leq 3p(k-1)$, $Y \subseteq A_j$, $Z \subseteq A_\ell$ and $\ell > j$. It follows from Lemma 14 that all the table entries can be computed in $2^{\mathcal{O}(pk+\eta)} n^{\mathcal{O}(1)}$ -time. Moreover, it follows from the Lemma 15 that for any $I \in \mathcal{T}[(k-|X|, j, 3p(k-1), Y), (Z, \ell)]$, it holds that $G[V(I)]$ is a p -edge-connected subgraph. This completes the correctness proof of our algorithm. We finally output $S = V(I)$ as the solution to the input instance. ◀

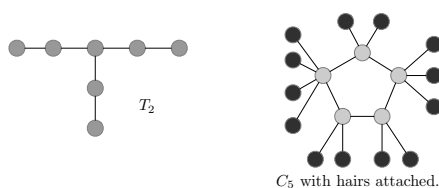
4 Applications of STEINER SUBGRAPH EXTENSION to some Graph Theoretic Problems

In this section, we describe some applications of our main result (Theorem 1) in parameterized algorithms. But before that, we prove the following lemma (proof is similar to the proof of Lemma 12).

► **Lemma 16** (\star). *Let G be an input graph, p be a fixed constant, and \mathcal{G} be a polynomial-time recognizable hereditary graph class. Then, there exists a polynomial-time algorithm that can check if there is a set $S \subseteq V(G)$ such that $G - S \in \mathcal{G}$ and $G[S]$ is p -edge-connected.*

The above lemma implies that for any polynomial-time recognizable (hereditary) graph class \mathcal{G} , we can test in polynomial-time if a feasible p -edge-connected vertex subset exists whose deletion results in a graph of class \mathcal{G} . Note that the class of all pathwidth-one graphs is a polynomial-time recognizable graph class. Moreover, for every fixed constant η , the class of all degree- η -graphs, or the class of all η -treedepth graphs, or the class of all graphs with no P_η as subgraphs are all polynomial-time recognizable graph classes. So, we can test the existence of feasible solutions for all our problems in polynomial-time.

Singly Exponential Algorithm for p -EDGE-CONNECTED- η -DEGREE DELETION SET. Now, we explain how Theorem 1 implies a singly exponential algorithm for p -EDGE-CONNECTED η -DEGREE DELETION SET problem. We formally state the problem below.



■ **Figure 1** An illustration of T_2 and a cycle with hairs attached.

p -EDGE-CONNECTED η -DEGREE DELETION SET (p -EDGE-CON-BDDS)

Input: An undirected graph $G = (V, E)$ and an integer k .

Parameter: k

Goal: Is there $S \subseteq V(G)$ such that $\Delta(G - S) \leq \eta$ and $G[S]$ is p -edge-connected?

The following is the first application of our main result (Theorem 1). As we have a guarantee from Lemma 16 that we can test if an input graph has a feasible solution to our problems, we assume without loss of generality that the input graph actually has a feasible solution. We use this assumption for all our subsequent problems.

► **Corollary 2.** p -EDGE-CON-BDDS admits a $2^{\mathcal{O}(pk+k\eta)}n^{\mathcal{O}(1)}$ -time algorithm.

The first part of the algorithm⁴ for the above result uses enumeration of all minimal vertex subsets the removal of which results in a graph of maximum degree at most η and then it invokes Theorem 1.

Singly Exponential Algorithm for p -EDGE-CON-PW1DS. Now, we describe a singly exponential time algorithm for p -EDGE CONNECTED PATHWIDTH-1 VERTEX DELETION problem using Theorem 1. We formally define the problem as follows.

p -EDGE-CON-PW1DS

Input: An undirected graph $G = (V, E)$ and an integer k .

Parameter: k

Goal: Is there $S \subseteq V(G)$ such that $G[S]$ is p -edge-connected and $\text{pw}(G - S) \leq 1$?

We use the following characterization that are related to pathwidth one graphs.

► **Proposition 17** ([16, 4]). A graph G has pathwidth at most one if and only if it does not contain a cycle or T_2 as a subgraph.

► **Proposition 18** ([26]). If G is a graph that does not contain any T_2, C_3, C_4 as subgraphs, then each connected component of G is either a tree, or a cycle with zero or more pendant vertices (“hairs”) attached to it. (See Figure 1 for an illustration)

Observe that the graphs of pathwidth one do not have T_2, C_3, C_4 as subgraphs. We prove the following lemma now.

► **Lemma 19** (*). Let G be an undirected graph that does not have any T_2, C_3, C_4 as subgraphs. Then, a 2-degeneracy sequence of G can be constructed in polynomial-time. Moreover, for every connected component D of G , a partition $D = C \uplus P$ can be computed in polynomial-time such that C is an induced path (or cycle) and P is the set of pendant vertices attached to C .

⁴ We refer to the full version for the proof

45:12 Finding a Highly Connected Steiner Subgraph and its Applications

It follows from Proposition 17 and Proposition 18 that any pathwidth one vertex deletion set must intersect all the subgraphs T_2, C_3, C_4 of a graph. But, once we have a set X such that $G - X$ has no T_2, C_3, C_4 as subgraphs, then there are some connected components of $G - X$ that can have cycles. In particular, due to Proposition 18, it holds that if a connected component of $G - X$ has a cycle, then it must be a cycle with some (possibly empty set of) pendant hairs attached to it. Then, we would need to find $S \supseteq X$ such that $G[S]$ is p -edge-connected and S contains at least one vertex from each of these cycles. This requires us to design an algorithm that uses the ideas similar to Lemma 14 and Lemma 15 but also has to satisfy an additional condition. We state the following lemma and give a proof for completeness.

► **Lemma 20** (\star). *Let $G = (V, E)$ be an undirected graph and $X \subseteq V(G)$ such that $G - X$ has no T_2, C_3, C_4 as subgraphs. Then, there exists an algorithm that runs in $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time and computes $S \supseteq X$ of size at most k such that $G - S$ has pathwidth at most one and $G[S]$ is a p -edge-connected subgraph.*

Using the above lemma, we provide an $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time algorithm for p -EDGE CONNECTED PATHWIDTH-1 VERTEX DELETION problem as follows.

► **Theorem 3.** *p -EDGE-CON-PW1DS admits an algorithm that runs in $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time.*

Proof. Let (G, k) be an instance of p -EDGE CONNECTED PATHWIDTH-1 VERTEX DELETION problem. First we enumerate all minimal vertex subsets X of size at most k such that $G - X$ has no T_2, C_3, C_4 as subgraphs. Since T_2 has 7 vertices, C_3 has 3 vertices and C_4 has 4 vertices, it takes $\mathcal{O}^*(7^k)$ -time to enumerate all such subsets the deletion of which results in a graph that has no C_3, C_4, T_2 as subgraphs. Let X be one such set such that $G - X$ has no C_3, C_4, T_2 as subgraphs. Due to Propositions 17 and 18, if D is a connected component of $G - X$, then either D is a caterpillar, or a cycle with hairs attached to it. It follows from Lemma 19 that there is a polynomial-time algorithm that gives a 2-degeneracy sequence ρ of the vertices of $G - X$. Moreover, if D is a connected component of $G - X$, then ρ provides a partition of $D = C \uplus P$ such that C is a cycle and P is the set of hairs attached to C . In particular, the vertices of C are put first, followed by the vertices of P in ρ . Furthermore, putting the vertices of C first followed by the vertices of P gives a 2-degeneracy ordering of D . For each such subset X , we invoke Lemma 20 to give an algorithm that runs in $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time and outputs S such that $X \subseteq S$ and $G - S$ has pathwidth at most one. The correctness of this algorithm also follows from the proof of Lemma 20. This completes the proof of this theorem. ◀

Singly Exponential Time Algorithms for p -EDGE-CON- η -TDDS and p -EDGE-CON- η -PVC.

Finally, we describe how we can get $2^{\mathcal{O}(pk)}n^{\mathcal{O}(1)}$ -time algorithm for p -EDGE-CONNECTED η -TREEDEPTH DELETION SET and p -EDGE-CONNECTED η -PATH VERTEX COVER problems. Note that both p and η are fixed constants. We restate the problem definitions below.

p -EDGE-CONNECTED η -TREEDEPTH DELETION SET (p -EDGE-CON- η -TDDS)

Input: An undirected graph $G = (V, E)$ and an integer k .

Parameter: k

Goal: Is there $S \subseteq V(G)$ such that $G[S]$ is p -edge-connected and $\text{td}(G - S) \leq \eta$.

p -EDGE-CONNECTED η -PATH VERTEX COVER

Input: An undirected graph $G = (V, E)$ and an integer k .

Parameter: k

Goal: Is there an $S \subseteq V(G)$ with at most k vertices such that $G - S$ has no P_η as subgraphs and $G[S]$ is p -edge-connected?

It is clear from the problem definition that a set S is called an η -path vertex cover of G if $G - S$ has no P_η as subgraph. The following proposition holds true for graphs of treedepth at most η .

► **Proposition 21** ([12]). *If a graph G has treedepth at least $\eta + 1$, then it has a connected subgraph H such that $\text{td}(H) > \eta$ and $|V(H)| \leq 2^{2^\eta}$.*

The above proposition implies that η -TREEDPTH DELETION SET problem can be characterized as \mathcal{H} -HITTING SET problem where \mathcal{H} contains only subgraphs of bounded size. It follows from the definition that p -EDGE-CON- η -PVC can be formulated in p -EDGE-CONNECTED \mathcal{H} -HITTING SET problem. It means that every minimal p -edge-connected η -treedepth deletion set contains a minimal η -treedepth deletion set and every minimal every minimal p -edge connected η -path vertex cover contains a minimal η -path vertex cover. We prove the following lemma that explains how we can construct a collection of all the minimal such solutions of size at most k .

► **Lemma 22** (\star). *Given a (connected) undirected graph $G = (V, E)$ and an integer k , the collection of all minimal η -treedepth deletion sets and the collection of all minimal η -path vertex covers can be obtained in $2^{2^\eta k} n^{\mathcal{O}(1)}$ -time and $\eta^k n^{\mathcal{O}(1)}$ -time respectively.*

The above lemma implies the next two results⁵ as other applications of our main result.

► **Theorem 4.** *p -EDGE-CON- η -TDDS admits an algorithm that runs in $2^{2^{2^\eta} + \mathcal{O}((p+\eta)k)} n^{2^{2^\eta}}$ -time.*

Observe that a graph with no P_η as subgraph has treedepth at most $\eta + 1$. It means that such a graph also has bounded degeneracy. So, we have the following theorem.

► **Theorem 5.** *p -EDGE-CON- η -PVC admits an algorithm that runs in $2^{\mathcal{O}((p+\eta)k)} n^{\mathcal{O}(1)}$ -time.*

5 Conclusions and Future Work

There are several possible directions of future work. Our main result proves that STEINER SUBGRAPH EXTENSION is FPT when the removal of terminals results in a bounded degenerate graph. It is unclear if STEINER SUBGRAPH EXTENSION is FPT even when $G - X$ is an arbitrary graph class and p is a fixed constant. Proving such a (positive or negative) result STEINER SUBGRAPH EXTENSION remains an interesting future work. If $p = 2$, then finding p -vertex/edge-connected steiner subgraph admits $k^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ -time algorithm [19, 15]. It remains open if a 2-vertex-connected steiner subgraph can be obtained in $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ -time even when the set of terminals is a vertex cover of the input graph. On the perspective of applications of Theorem 1, we have been successful in designing singly exponential-time FPT algorithms for p -EDGE-CON-BDDS, p -EDGE-CON- η -TDDS, p -EDGE-CON-PW1DS and p -EDGE-CON- η -PVC. But, our results do not capture several other graph classes. For instance,

⁵ The proofs of the next two theorems can be found in the full version.

the above algorithm crucially relies that all minimal vertex deletion sets without connectivity requirements can be enumerated in $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ -time and that a bounded degeneracy sequence can be computed in polynomial-time. Therefore, obtaining an FPT algorithm with singly exponential running time for each of FEEDBACK VERTEX SET, CLUSTER VERTEX DELETION, COGRAPH VERTEX DELETION with p -edge-connectivity constraints (even with $p = 2$) also remain interesting open problems.

References

- 1 Ankit Abhinav, Susobhan Bandopadhyay, Aritra Banik, and Saket Saurabh. Parameterized algorithms for finding highly connected solution. *Theor. Comput. Sci.*, 942:47–56, 2023.
- 2 Akanksha Agrawal, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Fast exact algorithms for survivable network design with uniform requirements. *Algorithmica*, 84(9):2622–2641, 2022.
- 3 MohammadHossein Bateni, Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dániel Marx. A PTAS for planar group steiner tree via spanner bootstrapping and prize collecting. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 570–583. ACM, 2016.
- 4 Kevin Burrage, Vladimir Estivill-Castro, Michael R. Fellows, Michael A. Langston, Shev Mac, and Frances A. Rosamond. The undirected feedback vertex set problem has a poly(k) kernel. In Hans L. Bodlaender and Michael A. Langston, editors, *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *Lecture Notes in Computer Science*, pages 192–202. Springer, 2006.
- 5 Radovan Cervený and Ondrej Suchý. Faster FPT algorithm for 5-path vertex cover. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPICs*, pages 32:1–32:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.
- 6 Radovan Cervený and Ondrej Suchý. Generating faster algorithms for d-path vertex cover. *CoRR*, abs/2111.05896, 2021. [arXiv:2111.05896](https://arxiv.org/abs/2111.05896).
- 7 Rajesh Hemant Chitnis, Andreas Emil Feldmann, Mohammad Taghi Hajiaghayi, and Dániel Marx. Tight bounds for planar strongly connected steiner subgraph with fixed number of terminals (and extensions). *SIAM J. Comput.*, 49(2):318–364, 2020.
- 8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 9 Celina M. H. de Figueiredo, Raul Lopes, Aleksander Andrade de Melo, and Ana Silva. Parameterized algorithms for steiner tree and dominating set: Bounding the leafage by the vertex leafage. In Petra Mutzel, Md. Saidur Rahman, and Slamun, editors, *WALCOM: Algorithms and Computation – 16th International Conference and Workshops, WALCOM 2022, Jember, Indonesia, March 24-26, 2022, Proceedings*, volume 13174 of *Lecture Notes in Computer Science*, pages 251–262. Springer, 2022.
- 10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 11 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 12 Zdenek Dvorák, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *Eur. J. Comb.*, 33(5):969–979, 2012.
- 13 Carl Einarson, Gregory Z. Gutin, Bart M. P. Jansen, Diptapriyo Majumdar, and Magnus Wahlström. p -edge/vertex-connected vertex cover: Parameterized and approximation algorithms. *J. Comput. Syst. Sci.*, 133:23–40, 2023.

- 14 Andreas Emil Feldmann and Dániel Marx. The complexity landscape of fixed-parameter directed steiner network problems. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPICs*, pages 27:1–27:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.
- 15 Andreas Emil Feldmann, Anish Mukherjee, and Erik Jan van Leeuwen. The parameterized complexity of the survivable network design problem. In Karl Bringmann and Timothy Chan, editors, *5th Symposium on Simplicity in Algorithms, SOSA@SODA 2022, Virtual Conference, January 10-11, 2022*, pages 37–56. SIAM, 2022.
- 16 Michael R. Fellows and Michael A. Langston. On search, decision and the efficiency of polynomial-time algorithms (extended abstract). In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 501–512. ACM, 1989.
- 17 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016.
- 18 Archontia C. Giannopoulou, Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. Uniform kernelization complexity of hitting forbidden minors. *ACM Trans. Algorithms*, 13(3):35:1–35:35, 2017.
- 19 Pinar Heggernes, Pim van ’t Hof, Dániel Marx, Neeldhara Misra, and Yngve Villanger. On the parameterized complexity of finding separators with non-hereditary properties. *Algorithmica*, 72(3):687–713, 2015.
- 20 Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. *ACM Trans. Algorithms*, 14(2):14:1–14:20, 2018.
- 21 Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On subexponential parameterized algorithms for steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 474–484. IEEE Computer Society, 2018.
- 22 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- 23 Zeev Nutov. Parameterized algorithms for node connectivity augmentation problems. *CoRR*, abs/2209.06695, 2022. [arXiv:2209.06695](https://arxiv.org/abs/2209.06695).
- 24 James G. Oxley. *Matroid theory*. Oxford University Press, 1992.
- 25 Fahad Panolan and Saket Saurabh. Matroids in parameterized complexity and exact algorithms. In *Encyclopedia of Algorithms*, pages 1203–1206. Springer, 2016.
- 26 Geevarghese Philip, Venkatesh Raman, and Yngve Villanger. A quartic kernel for pathwidth-one vertex deletion. In Dimitrios M. Thilikos, editor, *Graph Theoretic Concepts in Computer Science – 36th International Workshop, WG 2010, Zarós, Crete, Greece, June 28-30, 2010 Revised Papers*, volume 6410 of *Lecture Notes in Computer Science*, pages 196–207, 2010.