Fachbereich Mathematik & Informatik
AG Algorithmik
Prof. Dr. C. Komusiewicz

Wintersemester 2021/2022

Master-Arbeit zum Thema

# On the Complexity of Community-aware Network Sparsification

Autor: Emanuel Herrendorf
Betreuer: Prof. Dr. C. Komusiewicz
Nils Morawietz
Frank Sommer

Datum der Abgabe: 31. März 2022

Hiermit versichere ich, Emanuel Herrendorf, dass ich die vorliegende Arbeit mit dem Titel "On the Complexity of Community-aware Network Sparsification" selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Master-Arbeit wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.

Marburg, den 31.03.2022

## Abstract

Graphs are a natural data model in the field of social network analysis. This model can be augmented by the notion of communities which are subsets of vertices of the graph. Since many graphs that arise from applications in the field of social networks are large, an important operation before further analysis is reducing the number of edges. This operation is called sparsification. In this work, we study the family of Π-Network Sparsification problems where Π refers to a graph property. Given an undirected graph $G = (V, E)$, a set of subsets over $V$ called communities and an integer $\ell$, the question is whether there exists a sparsified graph, that is a spanning subgraph of $G$, with at most $\ell$ edges such that each subgraph induced by a community satisfies the graph property Π. For Π-Network Sparsification restricted to communities of size at most 3, we give a complexity dichotomy. Then, we analyze Π-Network Sparsification specifically for three properties: A minimum density requirement, the containment of a spanning star and the property of being connected. Since Π-Network Sparsification is NP-hard for these three graph properties, we study them in the framework of parameterized complexity and their fine-grained complexity. For example we show that Π-Network Sparsification is not solvable in $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$-time for these three specific graph properties under the Exponential Time Hypothesis.

## Zusammenfassung

Graphen sind ein natürliches Datenmodell im Bereich der Analyse von sozialen Netzwerken. Eine typische Erweiterung diese Modells sind Communities, welche Teilmengen der Knotenmenge des Graphs sind. Da viele Graphen, die aus Anwendungen im Bereich der sozialen Netzwerke stammen, sehr groß sind, ist eine wichtige Operation, bevor das Netzwerk analysiert wird, die Verringerung der Anzahl der Kanten. Dieser Vorgang wird Sparsification genannt. In dieser Arbeit beschäftigen wir uns mit der Familie von Π-Network Sparsification Problemen, wobei Π eine Eigenschaft wie beispielweise Zusammenhang ist. Gegeben seien ein ungerichteter Graph $G = (V, E)$, eine Menge von Teilmengen über $V$, welche Communities genannt werden, und eine natürliche Zahl $\ell$. Die Problemstellung ist, ob ein Teilgraph mit der gleichen Knotenmenge und maximal $\ell$ Kanten existiert, sodass jeder Teilgraph, der von einer Community induziert wird, die Eigenschaft Π erfüllt. Für Π-Network Sparsification eingeschränkt auf Instanzen, deren Communities eine maximale Größe von drei haben, geben wir eine Komplexitäts-Dichotomie an. Dann analysieren wir Π-Network Sparsification speziell für drei Eigenschaften von Graphen: Eine minimale Dichte, die Existenz eines zentralen Knotens, der zu allen anderen Knoten benachbart ist und Zusammenhang. Da Π-Network Sparsification für

diese drei Eigenschaften NP-schwer ist, analysieren wir deren fine-grained Komplexität und betrachten das Problem für diese drei Eigenschaften aus der Sicht der Parametrisierten Komplexität. Zum Beispiel zeigen wir, dass Π-Network Sparsification für diese drei Eigenschaften unter der Exponential Time Hypothesis nicht in $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$-Zeit gelöst werden kann.

# Contents

# 1. Introduction

Graphs are a natural data model for a wide range of applications. For example in transport planning road networks or rail networks are modelled as graphs [29]. Other applications where graphs are omnipresent are biological networks [26] and social networks [28].

In this work, we focus on a family of problems in the context of social networks. In the field of social network analysis, a common task is to identify the central actors of a social network and important connections between these actors. A typical extension to the graph model in the context of social networks is the notion of communities which are groups of actors having something in common. These communities can overlap and might not be disjoint. For example, a community in a social network might represent a group of people who are interested in the same topic.

Since many graphs that arise in applications are large, it is useful to reduce their size before further analysis. This leads to the task of network sparsification which aims to reduce the number of edges of a graph. A typical requirement on the sparsified network is that it preserves some property which is of importance in further analysis of that network.

Because communities are a central concept in social network analysis, it is important for a sparsified social network to preserve some graph property for each community and not only for the whole network. The combination of network sparsification and the notion of communities leads to the task of community-aware network sparsification which aims to reduce the number of edges of a graph while preserving some graph property for each community. This leads to the family of Π-Network Sparsification (Π-NWS) problems which was introduced by Gionis et al. [16]:

Π-Network Sparsification (Π-NWS)

**Input:** An undirected graph $G = (V, E)$, a set $\mathcal{C} = \{C_1, \dots, C_c\}$ of communities with $C_1, \dots, C_c \subseteq V$ and an integer $\ell$.

**Question:** Is there a sparsified graph $G' = (V, E')$ with $E' \subseteq E$ and at most $\ell$ edges such that each subgraph induced by a community $C_i \in \mathcal{C}$ satisfies the graph property Π?

A problem belonging the problem family Π-NWS gets as input an undirected graph $G = (V, E)$, a set of subsets over $V$ called communities and an integer $\ell$. In other words, the input consists of an undirected graph and a hypergraph over the same set of vertices where the hyperedges are the communities and an integer $\ell$. It then asks whether there exists a subgraph $G'$ of $G$ with the same vertex set $V$ and at most $\ell$ edges such that each subgraph in $G'$ induced by a community satisfies the graph property $\Pi$. In this work we assume that the graph property $\Pi$ is computable in polynomial time.

Gionis et al. [16] studied $\Pi$-NWS with respect to three specific graph properties in their work. The first property is a minimum density requirement, the second property is the containment of a spanning star, and third property is the property of being connected. In this work, we study the complexity of $\Pi$-NWS for these graph properties specificall. Since $\Pi$-NWS is NP-hard for these three graph properties, we study $\Pi$-NWS for these three graph properties in the framework of parameterized complexity. In addition to the parameterized complexity, we study the fine-grained complexity for these three graph properties. Moreover, we study the complexity of $\Pi$-NWS restricted to communities of size at most 3 in general.

## 1.1 Problem Definitions

In this section, we introduce the definitions of $\Pi$-NWS for the three specific graph properties we study in this work.

The first variant of $\Pi$-NWS is called Density Network Sparsification (Density NWS) and demands a minimum density requirement for each subgraph induced by a community. As an extension to $\Pi$-NWS we allow for each community a different density requirement instead of demanding on the same density requirement for all communities. Thus, in addition to the input of $\Pi$-NWS an instance of Density NWS takes a mapping describing the density requirement of each subgraph induced by a community. An example of an instance of Density NWS and a sparsified graph is shown in Figure 1.1.

Density Network Sparsification (Density NWS)

**Input:** An undirected graph $G = (V, E)$, a set $\mathcal{C} = \{C_1, \ldots, C_c\}$ of communities with $C_1, \ldots, C_c \subseteq V$, a mapping $\alpha : \mathcal{C} \to [0, 1]$ and an integer $\ell$.

**Question:** Is there a sparsified graph $G' = (V, E')$ with $E' \subseteq E$ and at most $\ell$ edges such that each subgraph induced by a community $C_i \in \mathcal{C}$ has a density of at least $\alpha(C_i)$?

The second variant of $\Pi$-NWS is called Stars Network Sparsification (Stars NWS) and demands that each subgraph induced by community contains a spanning star. An example of an instance of Stars NWS and a sparsified graph is shown in Figure 1.2.

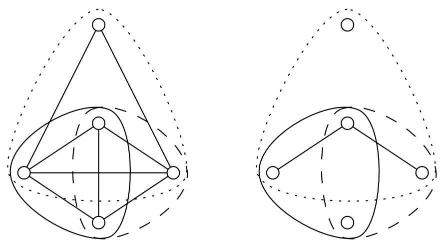

Figure 1.1: The left side shows the graph and the communities of the input of an instance of Density NWS. The right side shows a sparsified graph with two edges such that each subgraph induced by a community has density of at least 1/3

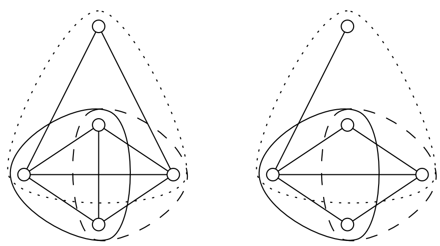

Figure 1.2: The left side shows the graph and the communities of the input of an instance of Stars NWS. The right side shows a sparsified graph with six edges such that each subgraph induced by a community contains a spanning star.

Stars Network Sparsification (Stars NWS)

**Input:** An undirected graph $G = (V, E)$, a set $\mathcal{C} = \{C_1, \ldots, C_c\}$ of communities with $C_1, \ldots, C_c \subseteq V$ and an integer $\ell$.

**Question:** Is there a sparsified graph $G' = (V, E')$ with $E' \subseteq E$ and at most $\ell$ edges such that each subgraph induced by a community $C_i \in \mathcal{C}$ contains a star with $|C_i| - 1$ leaves?

The third variant of Π-NWS is called Connectivity Network Sparsification (Connectivity NWS) and demands that each subgraph induced by community is connected. An example of an instance of Connectivity NWS and a sparsified graph is shown in Figure 1.3.

Connectivity Network Sparsification (Connectivity NWS)

**Input:** An undirected graph $G = (V, E)$, a set $\mathcal{C} = \{C_1, \ldots, C_c\}$ of communities with $C_1, \ldots, C_c \subseteq V$ and an integer $\ell$.

**Question:** Is there a sparsified graph $G' = (V, E')$ with $E' \subseteq E$ and at most $\ell$ edges such that each subgraph induced by a community $C_i \in \mathcal{C}$ is connected?

In addition, we denote with $d$-Density NWS, $d$-Stars NWS, and $d$-

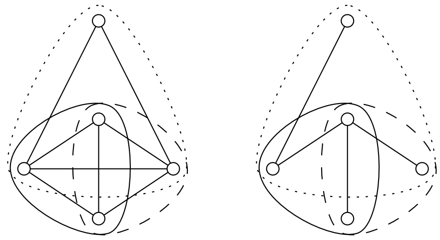

Figure 1.3: The left side shows the graph and the communities of the input of an instance of Connectivity NWS. The right side shows a sparsified graph with four edges such that each subgraph induced by a community is connected.

Connectivity NWS the respective problem restricted to communities of size at most $d$.

For instances of these three problem variants, we use the following convention. First, we assume that for each vertex $v \in V$ there exists a community $C_i \in \mathcal{C}$ with $v \in C_i$, otherwise $v$ can be deleted. Second, we assume that for each community $C_i \in \mathcal{C}$ the subgraph induced by $C_i$ satisfies the respective graph property because otherwise it is a trivial no-instance. This is the case because the three graph properties are closed under the operation of adding additional edges to the graph. Finally, we assume that each community has size at least 2.

## 1.2 Known Results

Gionis et al. [16] showed NP-hardness for Density NWS by a reduction from Hitting Set and NP-hardness for Stars NWS by a reduction from 3D Matching. They also provided a factor-$O(\log |\mathcal{C}|)$ approximation for Density NWS with a running time of $O(\sum_{C_i \in \mathcal{C}} |C_i| \cdot m \cdot |\mathcal{C}|)$ where $m$ is the number of edges in the input graph. Moreover, they gave an factor-$z$ approximation algorithm for Stars NWS and Connectivity NWS with running time $O(|\mathcal{C}| \cdot n \log n + |\mathcal{C}| \cdot \sum_{C_i \in \mathcal{C}} |C_i|)$ where $n$ is the number of vertices in the input graph and $z$ is the size of the maximal set of communities such that these communities are pairwise intersecting.

A special case of Connectivity NWS, where the input graph $G$ is a clique, was studied under the names Subset Interconnection Design [8], Network Construction [1] and Interconnection Graph Problem [14]. NP-hardness for instances restricted to communities of size at most 3 was shown by Fan et al. [14]. Chen et al. [8] proved fixed-parameter tractability of Subset Interconnection Design with respect to parameter $|\mathcal{C}|$ by giving a problem kernel with $O(8^{|\mathcal{C}|})$ vertices and gave an FPT-algorithm for parameter $d+t$ with running time $O(d^{18dt} \cdot d \cdot n \cdot |\mathcal{C}| + n \cdot |\mathcal{C}| \cdot d^2)$

where $d$ is the size of the biggest community and $t$ is the size of a minimal feedback edge set of the sparsified graph. Angulin et al. [1] gave a factor-$O(\log |\mathcal{C}|)$ approximation algorithm.

## 1.3 Further Related Work

First, we present past work closely related to Density NWS, Stars NWS and Connectivity NWS. Then, we give an overview of past work related to network sparsification in general.

For the special case of Connectivity NWS, where the graph $G$ is a clique, various Integer Linear Programming and Mixed Integer Linear Programming formulations were proposed [10, 4]. For a variant of Stars NWS, called Complete Optimal Stars Clustering Tree Problem, Korach and Stern [21] gave a polynomial-time algorithm. In this variant, $G$ is a complete graph and the sparsified graph $G'$ must be a tree $T$ and the closed neighborhood of each center vertex $c_i$ of a community $C_i$ in $T$ is the community itself. This implies that the centers of the stars in the subgraphs induced by two different communities are different which is not the case for Stars NWS. An instance of Density NWS can be expressed as variant of Hitting Set where the hitting set has to hit each set more than once by setting the universe to the edge set and the collection of subsets to the communities. This variant has been studied under the names Multiple Hitting Set [24] and Hitting Set with Quotas [18].

Network sparsification for graphs without the notion of communities was already studied in the past for different graph properties and different extensions to graphs. Thereby the graph property is either completely preserved or only approximately preserved in the sparsified graph.

Graph sparsification which aims to preserve the distance except some constant $c$ between $p$ given pairs of vertices was studied by Bodwin [3]. He showed that every undirected graph has a subgraph with $O(np^{2/7})$ edges which preserves the distance of the given $p$ pairs except the constant $c = 4$. Chekuri et al. [5, 6] proposed an approach to sparsify a graph while preserving element connectivity which is a generalization of edge connectivity. For a graph $G = (V, E)$ and a subset $T \subseteq V$, element connectivity between two vertices $u, v \in V$ is defined as the maximum number of $(u, v)$-paths that are pairwise disjoint in their edges and the vertices $V \setminus T$. A sparsification method for weighted graphs which retains a high connectivity was provided by Zhou et al. [30]. Parchas et al. [25] extended undirected graphs by uncertainty which assigns each edge a probability whether it exists in the graph. Lindner et al. [22] compared different edge scoring methods which are used to filter the edges to sparsify a social network. For example, they scored edges by the number of triangles that contain these edges, by the local similarity of the endpoints of the edge or by the highest degree of the endpoints of the edge. Then, they compared how these different scores affect the preservation of different global graph properties like the diameter or the clustering coefficient. Mathioudakis et al. [23] proposed an algorithm which sparsifies a social network with the goal to approximately preserve the information flow in the sparsified graph based on past information flows. Given a social network and a history of information flows, that is for example how a message propagated through the network, their goal is to preserve the $\ell$-most important edges in the network with respect to the history of information flow.

## 1.4 Our Results

In Section 3.1, we give a complexity dichotomy of Π-NWS restricted to communities of size at most 3. Then, we show in Section 3.2 that neither Density NWS nor Stars NWS nor Connectivity NWS is solvable in $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$ time under the Exponential Time Hypothesis, where $n$ is the number of vertices in the input graph.

In Section 4, we study Density NWS, Stars NWS and Connectivity NWS in the framework of parameterized complexity. A problem is fixed-parameter tractable if there is an algorithm solving the problem such that its running time can be divided into a possibly exponential part and a polynomial part meeting the following requirements. The exponential part depends only on a parameter which is independent from the size of the input and the polynomial part depends on the size of the input. If a problem is $W[1]$-hard or $W[2]$-hard, then it is unlikely that the problem is fixed parameter tractable. The absence of polynomial kernel for a problem means that there does not exist an effective preprocessing for a problem to reduce the size of the input.

For parameter $\ell$, the number of edges in the sparsified graph, we show fixed-parameter tractability for Stars NWS and Connectivity NWS and $W[2]$-hardness for Density NWS. Furthermore, we show that there are no polynomial kernels for Stars NWS and Connectivity NWS with respect to parameter $\ell$ unless NP $\subseteq$ coNP/poly. For parameter $k := m - \ell$, we show $W[1]$-hardness for Density NWS, Stars NWS and Connectivity NWS. With respect to parameter $t$, the size of a minimal feedback edge set of the sparsified graph, we give a polynomial-time algorithm for Stars NWS for $t = 0$ and show $W[2]$-hardness for Connectivity NWS. For parameter $|\mathcal{C}|$, we prove fixed-parameter tractability for Density NWS and Stars NWS. In addition, we show that there are no polynomial kernels for Density NWS and Stars NWS with respect to parameter $|\mathcal{C}|$ unless NP $\subseteq$ coNP/poly.

# 2. Preliminaries

In this section, we introduce the notation used throughout this work and give an overview of the most important concepts of parameterized complexity theory and complexity theory in general.

## 2.1 Graph Theory

For an in-depth introduction to the topic of graph theory, we refer to the book of Diestel [11]. We define an *undirected graph* as $G := (V, E)$ where $V$ is a finite set of vertices and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ a set of edges. For brevity, we always refer to an undirected graph when we say graph. For a graph $G$, we denote the set of vertices with $V(G)$ and the set of edges with $E(G)$. The number of vertices is abbreviated by $n := |V|$ and the number of edges is abbreviated by $m := |E|$. Two vertices $u, v \in V(G)$ are *adjacent* in graph $G$, if $\{u, v\} \in E(G)$. We say that an edge $e$ is *incident* to a vertex $v$, if $v$ is an endpoint of $e$. For two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we define $G_1 \cup G_2 := (V_1 \cup V_2, E_1 \cup E_2)$.

The *open neighborhood* of a vertex $v$ is defined as $N_G(v) := \{u \in V \mid \{u, v\} \in E\}$. The *closed neighborhood* of a vertex $v$ is defined as $N_G[v] := N(v) \cup \{u\}$. In general, we omit the subscript $G$ in cases where $G$ is unanimous. We call $G' := (V', E')$ a *subgraph* of an graph $G$, if $V' \subseteq V(G)$ and $E' \subseteq \{\{u, v\} \in E(G) \mid u, v \in V'\}$. For a subset $S \subseteq V(G)$, we say that $G[S] := (S, \{\{u, v\} \in E \mid u, v \in S\})$ is the subgraph of $G$ induced by $S$. A sequence of distinct vertices $(v_1, \ldots, v_k)$ of a graph $G$ is called $(v_1, v_k)$*-path*, if $\{v_i, v_{i+1}\} \in E(G)$ for each $i \in [1, k-1]$. If $(v_1, \ldots, v_k)$ is a path in $G$ for $k \geq 3$ and $\{v_1, v_k\} \in E(G)$, then we call $(v_1, \ldots, v_k)$ a *cycle* in $G$. A graph without a cycle is called *acyclic*. A *feedback edge set* of a graph $G$ is defined as a set of edges that needs to be removed from $G$ such that $G$ is acyclic. We say two vertices $u, v$ are *connected* in $G$, if there exists an $(u, v)$-path in $G$. A graph $G$ is called *connected*, if each pair of vertices $u, v \in V(G)$ is connected. We say $S \subseteq V$ is a *connected component* of $G$, if $G[S]$ is connected and $S$ is inclusion-maximal.

For a graph $G$, a set $S \subseteq V(G)$ with $|E(S)| = \binom{|S|}{2}$ is called a *clique*. A size-three clique is also called *triangle*. A graph $G$ is called a *star* of size $n - 1$ with *center* $z \in V(G)$, if $E(G) = \{\{z, v\} \mid v \in V \setminus \{z\}\}$. The *density* of a graph $G$ is defined as $\text{dens}(G) := m/\binom{n}{2}$. We say that $G$ contains a *spanning star*, if there exist a subgraph $G'$ of $G$ such that $G'$ is a star of size

$n-1$. A set $X \subseteq V(G)$ is a *vertex cover* of graph $G$, if each edge of $G$ is incident to at least one vertex in $X$. A set $X \subseteq V(G)$ is an *independent set* of graph $G$, if the vertices of $X$ are pairwise not adjacent in $G$.

We define a hypergraph as $\mathcal{H} \coloneqq (X, \mathcal{C})$ where $X$ is a finite set of elements and $\mathcal{C}$ is a collection of subsets of $X$ called *hyperedges*. The *line graph* of a hypergraph $\mathcal{H} = (X, \mathcal{C})$ is a graph defined as $L(\mathcal{H}) \coloneqq (\mathcal{C}, \{\{C_i, C_j\} \mid C_i, C_j \in \mathcal{C}, C_i \neq C_j, C_i \cap C_j \neq \emptyset\})$. Its vertex set are the hyperedges and the edges are pairs of hyperedges that are not disjoint in $\mathcal{H}$. A hypergraph $\mathcal{H}$ is *connected*, if its line graph $L(\mathcal{H})$ is connected. A set $S \subseteq X$ is a *connected component* of $\mathcal{H}$, if there exists a connected component $S'$ in $L(\mathcal{H})$ such that $S = \bigcup_{C_i \in S'} C_i$. For a family of sets $\mathcal{S}$ over a universe $U$, we say that $X \subseteq U$ is a *hitting set* if $X \cap S_i \neq \emptyset$ for each $S_i \in \mathcal{S}$.

## 2.2 Complexity Theory

In this section, we give a brief overview over the most important concepts in complexity theory. A detailed introduction to complexity theory can be found in the standard monographs [17, 2]. A decision problem is a language $L \subseteq \{0,1\}^*$. A decision algorithm for $L$ is some computable function $f_L : \{0,1\}^* \to \{0,1\}$ such that $f_L(x) = 1$ if and only if $x \in L$. A word $x \in \{0,1\}^*$ is called a yes-instance of $L$ if and only if $x \in L$, otherwise it is called a no-instance. A decision problem $A$ is polynomial-time reducible to a decision problem $B$ if there exists a function $g : \{0,1\}^* \to \{0,1\}^*$ which is computable in polynomial time in $|I|$ such that $I \in A$ if and only if $g(I) \in B$ for all instances $I \in \{0,1\}^*$. The algorithm computing the function $g$ is called polynomial-time reduction. The polynomial-time reducible relation is denoted by $\leq_p$. Note that $\leq_p$ is transitive.

A complexity class $C$ is a set of languages. A language $L$ is in the complexity class P if $f_L(I)$ is computable in time polynomial in $|I|$ for each instance $I$ of $L$. A language $L$ is in the complexity class NP if there exists a polynomial $p$ and a verifier $V$ such that for each $x \in \{0,1\}^*$ there exists a certificate $c \in \{0,1\}^{p(|x|)}$ such that the verifier $V$ accepts the input $(x, c)$ in time polynomial in $|x|$ if and only if $x \in L$. Note that P $\subseteq$ NP and P $\subset$ NP is a widely believed hypothesis. A language $B$ is NP-hard if every language $A \in$ NP is polynomial-time reducible to $B$. A language $B$ is NP-complete if it is in the complexity class NP and NP-hard. An example for a NP-complete problem is 3-SAT.

3-SAT

**Input:** A boolean formula $\phi$ in conjunctive normal form with at most three literals per clause.

**Question:** Is $\phi$ satisfiable, that is does exist a variable assignment setting at least one literal to `true` for each clause?

The Exponential Time Hypothesis (ETH) is a complexity theoretic hardness assumption about 3-SAT which has been formulated by Impagliazzo et al. [19]. The ETH basically states that 3-SAT cannot be solved in $2^{o(n)}$ time, where $n$ denotes the number of variables. Note that the ETH implies $\mathrm{P} \neq \mathrm{NP}$. In addition with the Sparsification Lemma [20] the ETH implies that 3-SAT cannot be solved in $2^{o(n+m)}$ time where $n$ denotes the number of variables and $m$ denotes the number of clauses. If a polynomial-time reduction from 3-SAT to a decision problem $A$ exists which transforms every instance $I$ of 3-SAT with $n$ variables and $m$ clauses to an instance $I'$ of $A$ with $|I'| \leq p(n+m)$ for some polynomial $p$ such that $I \in$ 3-SAT if and only if $I' \in A$, then assuming the ETH an algorithm with running time $2^{o(p^{-1}(|I'|))}$ does not exist where $p^{-1}$ denotes the inverse function of $p$ [12, 9].

## 2.3 Parameterized Complexity

In this section, we give a brief overview over the most important concepts in parameterized complexity. For a detailed introduction to the field of parameterized complexity, we refer to the book of Downey and Fellows [12] and the book of Cygan et al. [9]. A parameterized problem is a language $L \subseteq \{0,1\}^* \times \mathbb{N}$. An element $(x,k) \in \{0,1\}^* \times \mathbb{N}$ is called an instance $I$ of a parameterized problem where $x$ denotes the input of the decision problem and $k$ the parameter. Two instances $I$ and $I'$ are called equivalent if $I \in L$ if and only if $I' \in L$. The instance $I$ is called a yes-instance if and only if $I \in L$ and a no-instance if and only if $I \notin L$. A decision algorithm for a parameterized problem $L$ computes some computable function $f_L : L \subseteq \{0,1\}^* \times \mathbb{N} \to \{0,1\}$ such that $f_L(x,k) = 1$ if and only if $(x,k) \in L$.

A set $C$ of parameterized problems is a complexity class. A parameterized problem $A$ is fixed-parameter tractable, if there exists a decision algorithm running in $f(k) \cdot \mathrm{poly}(|x|)$ time for every instance $(x,k) \in \{0,1\}^* \times \mathbb{N}$. The corresponding complexity class is called FPT. If there exists a decision algorithm running in $f(k) \cdot |x|^{f(k)}$ time for every instance $(x,k) \in \{0,1\}^* \times \mathbb{N}$ of a parameterized problem $A$, then $A$ is contained in the complexity class XP. Note that $\mathrm{FPT} \subseteq \mathrm{XP}$.

A parameterized problem $A$ is parameterized reducible to a parameterized problem $B$ if there exists a computable function $g : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^* \times \mathbb{N}$ and a polynomial $h$, such that $g$ turns each instance instance $(x,k)$ of $A$ into an instance $(x',k')$ of $B$ in $f(k) \cdot \mathrm{poly}(|x|)$ time for some computable function $f$ such that $k' \leq h(k)$ and $(x',k') \in B$ if and only if $(x,k) \in A$. The algorithm computing the function $g$ is called parameterized reduction. The parameterized reducible relation is denoted by $\leq_{\mathrm{FPT}}$. Note that $\leq_{\mathrm{FPT}}$ is transitive. An example for a fixed-parameter tractable problem is Vertex Cover parameterized by the size of the vertex cover.

Vertex Cover

| | |
|---|---|
| **Input:** | An undirected graph $G = (V, E)$ and an integer $k$. |
| **Question:** | Does $G$ contain a vertex cover of size most $k$? |

The W-Hierarchy is a collection of the complexity classes $W[i]$ for $i \geq 1$ with FPT $\subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[t]$. A common assumption is that FPT $\neq W[1]$. Based on this assumption a parameterized problem $B$ is fixed-parameter intractable if a $W[i]$-hard problem $A$ is parameterized reducible to $B$. An example for a $W[1]$-hard problem is Independent Set parameterized by the size of the independent set.

Independent Set

| | |
|---|---|
| **Input:** | An undirected graph $G = (V, E)$ and an integer $k$. |
| **Question:** | Does $G$ contain an independent set of size most $k$? |

An example for a $W[2]$-hard problem is Hitting Set parameterized by the size of the hitting set.

Hitting Set

| | |
|---|---|
| **Input:** | A set of elements $U$, a collection $\mathcal{S}$ of subsets of $U$ and an integer $k$. |
| **Question:** | Does $\mathcal{S}$ has a hitting set of size most $k$? |

Another concept in parameterized complexity is kernelization. A reduction rule is an algorithm for a parameterized problem $A$ transforming an instance $(x, k)$ of $A$ into an equivalent instance $(x', k')$ of $A$ with $k' \leq k$. We also allow reduction rules which state that an instance is a no-instance. A kernelization algorithm for a parameterized problem $A$ is a function $k_A : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^* \times \mathbb{N}$ which turns for some computable function $g$ every instance $(x, k)$ of $A$ in an instance $(x', k')$ of $A$ in time polynomial in $|x|$ such that $(x, k) \in A$ if and only if $(x', k') \in A$ and $|x'| + k' \leq g(k)$. In case a kernelization algorithm exists, we say that $A$ admits a kernel. In case of $g$ being a polynomial function, we say that $A$ admits a polynomial kernel. Note that a parameterized problem admits a kernel if and only if it is fixed-parameter tractable.

For two parameterized problems $A$ and $B$ a polynomial parameter transformation is an algorithm computing a function $f : \{0,1\}^* \times \mathbb{N} \to \{0,1\} \times \mathbb{N}$ which turns every instance $(x, k)$ of $A$ in an instance $(x', k')$ of $B$ in polynomial time in $|x|$ such that $(x, k) \in A$ if and only if $(x', k') \in B$ and $k' \leq p(k)$ for some polynomial $p$. If there exists a polynomial parameter transformation from a parameterized problem $A$ without a polynomial kernel to a parameterized problem $B$, then $B$ also does not admit a polynomial kernel. A common assumption under which fixed-parameter tractable problems do not admit a polynomial kernel is that NP $\not\subseteq$ coNP/ poly.

# 3. Fine-grained Complexity

## 3.1 NP-Hardness

The NP-hardness of Density NWS and Stars NWS was already shown by Gionis et al. [16]. The gave a reduction from Hitting Set to Density NWS and a reduction from 3D Matching to Stars NWS. In the following, we adapt the idea of the reduction from Hitting Set to Density NWS to reduce from Vertex Cover to Density NWS obtaining NP-hardness even if the problem instances are restricted to communities of size at most 3.

**Theorem 3.1.** Density NWS *is NP-hard even if restricted to communities of size at most* 3.

*Proof.* We reduce from Vertex Cover. Let $I_{\mathrm{VC}} = (G = (V, E), k)$ be an instance of Vertex Cover. First, we give some intuition. The vertices of the Vertex Cover instance are expressed as edges in the the 3-Density NWS instance. The edges of the Vertex Cover instance are expressed as communities in the the 3-Density NWS instance. Next, we define the graph $G_{\mathrm{3DNS}} = (V_{\mathrm{3DNS}}, E_{\mathrm{3DNS}})$. The set $V_{\mathrm{3DNS}}$ contains each vertex of $V$ and one additional vertex $z$. The edge set is $E_{\mathrm{3DNS}} \coloneqq \{\{v, z\} \mid v \in V\}$. Thus, $G_{\mathrm{3DNS}}$ is a star with $z$ as center. Each edge of this star represents a vertex of $V$. We define the set of communities $\mathcal{C} \coloneqq \{\{u, v, z\} \mid \{u, v\} \in E\}$. Hence, the set $\mathcal{C}$ contains for each edge in $E$ a community representing it. Now, we define the density requirement $\alpha : \mathcal{C} \to [0, 1]$, $C_i \mapsto \frac{1}{3}$ to encode that a vertex cover of $G$ contains at least one endpoint of each edge. Finally, we set the parameter $\ell \coloneqq k$. Let $I_{\mathrm{3DNS}} = (G_{\mathrm{3DNS}}, \mathcal{C}, \alpha, \ell)$ denote the resulting instance of 3-Density NWS. An example of the construction is shown in Figure 3.1.

**Correctness** We show that $I_{\mathrm{VC}}$ is a yes-instance of Vertex Cover if and only if $I_{\mathrm{3DNS}}$ is a yes-instance of 3-Density NWS.

($\Rightarrow$) Let $X$ be a vertex cover of size at most $k$ of $G$. We show how to obtain a sparsified graph $G'_{\mathrm{3DNS}} = (V_{\mathrm{3DNS}}, E'_{\mathrm{3DNS}})$ with $|E'_{\mathrm{3DNS}}| \leq |X|$. We set $E'_{\mathrm{3DNS}} \coloneqq \{\{x, z\} \mid x \in X\}$ noting that $|E'_{\mathrm{3DNS}}| \leq |X|$. For each community $C_i = \{u, v, z\} \in \mathcal{C}$, we observe that $\{u, z\} \in E'_{\mathrm{3DNS}}$ or $\{v, z\} \in E'_{\mathrm{3DNS}}$ because $X$ is a vertex cover implying $u \in X$ or $v \in X$. Hence, for each community $C_i \in \mathcal{C}$, the density requirement $\alpha(C_i) = \frac{1}{3}$ is satisfied because

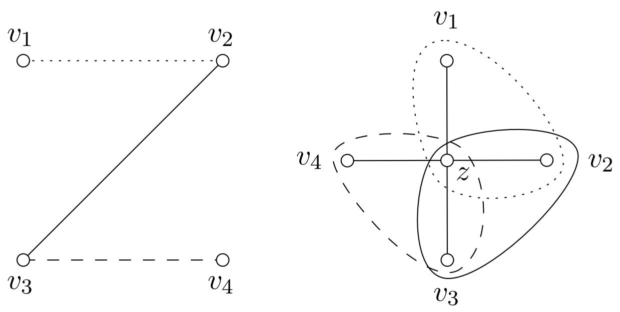


Figure 3.1: An example of the construction. The left side shows the Vertex Cover instance, the right side shows the 3-Density NWS instance. The different line styles show which community of the 3-Density NWS instance corresponds to which edge of the Vertex Cover instance.

$|E(G'_{\text{3DNS}}[C_i])| \geq 1$. Therefore, $G'_{\text{3DNS}}$ is a sparsified graph implying $I_{\text{3DNS}}$ is a yes-instance of 3-Density NWS.

($\Leftarrow$) Let $I_{\text{3DNS}}$ be a yes-instance of 3-Density NWS and let $G'_{\text{3DNS}} = (V_{\text{3DNS}}, E'_{\text{3DNS}})$ be a sparsified graph. We show how to obtain a vertex cover $X$ of $G$ with $|X| \leq |E'_{\text{3DNS}}|$. We set $X := \{x \mid \{x, z\} \in E'_{\text{3DNS}}\}$. For each edge $\{u, v\} \in E$, we observe that $u \in X$ or $v \in X$ because there exists a community $C_i = \{u, v, z\} \in \mathcal{C}$ with a satisfied density requirement $\alpha(C_i) = \frac{1}{3}$ implying $\{u, z\} \in E'_{\text{3DNS}}$ or $\{v, z\} \in E'_{\text{3DNS}}$. Therefore, $X$ is vertex cover of $G$ with $|X| \leq |E'_{\text{3DNS}}|$ implying that $I_{\text{VC}}$ a yes-instance of Vertex Cover. □

The NP-hardness proof for 3-Density NWS in Theorem 3.1 constructs instances with small density requirements. It would be interesting to see whether any other non-trivial density requirement makes an instance easier. A trivial density requirement would be 0 or $1 - \frac{1}{d} + \epsilon$ for $\epsilon > 0$ in an instance of Density NWS restricted to communities of size at most $d$. Unfortunately, the problem remains NP-hard regardless of which non-trivial density requirement is selected as shown in Theorem 3.3. This applies even if all communities have the same density requirement. Before we present the proof, we make an observation about a yes-instance of Density NWS regarding edges which are definitely contained in the sparsified graph.

**Lemma 3.2.** *Let $I = (G = (V, E), \mathcal{C}, \alpha, \ell)$ be an instance of* Density NWS. *If $I$ is a yes-instance with the sparsified graph $G' = (V, E')$, then $E(G[C_i]) \subseteq E'$ for each community $C_i \in \mathcal{C}$ with $|E(G[C_i])| = \lceil \binom{|C_i|}{2} \cdot \alpha(C_i) \rceil$.*

*Proof.* We assume that $I$ is a yes-instance of Density NWS with the sparsified graph $G' = (V, E')$. Let $C_i \in \mathcal{C}$ be a community with $|E(G[C_i])| = \lceil \binom{|C_i|}{2} \cdot \alpha(C_i) \rceil$. Now, assume that $E(G[C_i]) \not\subseteq E'$. This leads to the contradiction that $G'$ is not a sparsified graph because $E(G'[C_i]) < |E(G[C_i])| =$

$\lceil \binom{|C_i|}{2} \cdot \alpha(C_i) \rceil$ implies that the density requirement of community $C_i$ is not satisfied in $G'$. □

Using Lemma 3.2 we are able to adapt the proof of Theorem 3.1 for any non-trivial density requirement and a community size of an arbitrary but fixed $d$.

**Theorem 3.3.** *$d$-Density NWS is NP-hard for every constant mapping $\alpha(x) = c$ with $c \in (0, 1 - \frac{1}{d}]$.*

*Proof.* Again, we give a polynomial reduction from Vertex Cover. Let $I_{\mathrm{VC}} = (G = (V, E), k)$ be an instance of Vertex Cover. Let $I_{\mathrm{3DNS}} = (G_{\mathrm{3DNS}} = (V_{\mathrm{3DNS}}, E_{\mathrm{3DNS}}), \mathcal{C}, \alpha, \ell)$ be the instance of 3-Density NWS obtained from the reduction in Theorem 3.1. Recall that the instance $I_{\mathrm{3DNS}}$ has the following properties:

1. $G_{\mathrm{3DNS}}$ is a star with the center vertex $z$ and $z \notin V$.
2. Each community $C_i \in \mathcal{C}$ has size 3.
3. For each community $C_i \in \mathcal{C}$ the induced subgraph $G_{\mathrm{3DNS}}[C_i]$ has two edges.

We create for each $C_i \in \mathcal{C}$ a community $C'_i = C_i \cup \{w^i_1, \ldots, w^i_{d-3}\}$ where $w^i_j$ are $d-3$ new vertices. Let $\mathcal{C}'$ denote the set of these communities. Next, we add $\lceil \binom{d}{2} \cdot c \rceil - 1$ arbitrary new edges to each subgraph induced by a community $C'_i \in \mathcal{C}'$. Let $E'$ denote the set of these new edges. Next, we create for each edge $\{u, v\} \in E'$ a new community $\hat{C}_i = \{u, v\} \cup \{y^i_1, \ldots, y^i_{d-2}\}$ where $y^i_j$ are $d-2$ new vertices. The purpose of these communities is to ensure that a sparsified graph contains all edges of $E'$. Let $\hat{\mathcal{C}}$ denote the set of these communities. Next, we add $\lceil \binom{d}{2} \cdot c \rceil - 1$ arbitrary new edges to each subgraph induced by a community $\hat{C}_i \in \hat{\mathcal{C}}$. Let $\hat{E}'$ denote the set of these new edges.

Finally, let $I_{d\mathrm{DNS}} = (G_{d\mathrm{DNS}} = (V', E_{\mathrm{3DNS}} \cup E' \cup \hat{E}'), \mathcal{C}' \cup \hat{\mathcal{C}}, \ell + |E'| + |\hat{E}'|)$ denote the instance resulting from the construction. An example of the adapted construction is shown in Figure 3.2.

**Correctness** We show that $I_{\mathrm{3DNS}}$ is a yes-instance of 3-Density NWS if and only if $I_{d\mathrm{DNS}}$ is a yes-instance of $d$-Density NWS . Because of the correctness of the reduction in Theorem 3.1 this implies that $I_{\mathrm{VC}}$ is a yes-instance of Vertex Cover if and only if $I_{d\mathrm{DNS}}$ is a yes-instance of $d$-Density NWS.

($\Rightarrow$) Let $I_{\mathrm{3DNS}}$ be a yes-instance of 3-Density NWS and let $G'_{\mathrm{3DNS}}$ be a sparsified graph. We show how to obtain a sparsified graph $G'_{d\mathrm{DNS}} = (V_{d\mathrm{DNS}}, E'_{d\mathrm{DNS}})$ with $|E'_{d\mathrm{DNS}}| \leq \ell + |E'| + |\hat{E}'|$ for the instance $I_{d\mathrm{DNS}}$. We set $E'_{d\mathrm{DNS}} \coloneqq E(G'_{\mathrm{3DNS}}) \cup E' \cup \hat{E}'$. This satisfies $|E'_{d\mathrm{DNS}}| \leq \ell + |E'| + |\hat{E}'|$. By the definition of $E'_{d\mathrm{DNS}}$, we observe that the density requirement of the

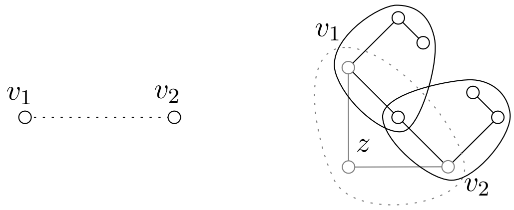


Figure 3.2: The adaption of the example shown in Figure 3.1 for $d = 4$ and $c = \frac{1}{2}$. The left side shows a part of the VERTEX COVER instance $I_{\text{VC}}$, the right side shows the corresponding part of the 4-DENSITY NWS instance $I_{4\text{DNS}}$ (the black vertices, edges, and communities are the added ones).

communities in $\hat{\mathcal{C}}$ is satisfied. Since $I_{3\text{DNS}}$ is a yes-instance of 3-DENSITY NWS, we have $|E(G'_{3\text{DNS}}[C_i])| \geq 1$ for each community $C_i \in \mathcal{C}$. Because of $E' \subseteq E'_{d\text{DNS}}$ we have $|E(G'_{d\text{DNS}}[C'_i])| = \lceil \binom{d}{2} \cdot c \rceil - 1 + |E(G'_{3\text{DNS}}[C_i])| \geq \lceil \binom{d}{2} \cdot c \rceil$ for each community $C'_i \in \mathcal{C}'$. Hence, the density requirement of each community is satisfied in $G'_{d\text{DNS}}$ implying that $I_{d\text{DNS}}$ is a yes-instance of $d$-DENSITY NWS.

($\Leftarrow$) Let $I_{d\text{DNS}}$ be a yes-instance of $d$-DENSITY NWS and let $G'_{d\text{DNS}}$ be a sparsified graph. We show how to obtain a sparsified graph $G'_{3\text{DNS}} = (V_{3\text{DNS}}, E'_{3\text{DNS}})$ with at most $\ell$ edges for the instance $I_{3\text{DNS}}$. For each community $\hat{C}_i \in \hat{\mathcal{C}}$ the induced subgraph $G_{d\text{DNS}}[\hat{C'}_i]$ contains $\lceil \binom{d}{2} \cdot c \rceil$ edges allowing us to apply Lemma 3.2. By Lemma 3.2 we conclude that $(E' \cup \hat{E}') \subseteq E(G'_{d\text{DNS}})$. We set $E'_{3\text{DNS}} := E(G'_{d\text{DNS}}) \setminus (E' \cup \hat{E}')$ satisfying $|E'_{3\text{DNS}}| = |E(G'_{d\text{DNS}})| - |E'| - |\hat{E}'| \leq \ell$. Observe that $E'_{3\text{DNS}} \subseteq E_{3\text{DNS}}$. Since $I_{d\text{DNS}}$ is a yes-instance of $d$-DENSITY NWS, we have $\lceil \binom{d}{2} \cdot c \rceil \geq E(G'_{d\text{DNS}}[C'_i])$ for each community $C'_i \in \mathcal{C}'$. Since we added only $\lceil \binom{d}{2} \cdot c \rceil - 1$ edges to each community $C'_i \in \mathcal{C}'$, there exists an edge $e \in E_{3\text{DNS}}$ with $e \in E(G'_{d\text{DNS}}[C'_i])$. This implies that there exists for each community $C_i \in \mathcal{C}$ an edge $e' \in E'_{3\text{DNS}}$ with $e' \in E(G'_{3\text{DNS}}[C_i])$. Hence, the density requirement of each community is satisfied in $G'_{3\text{DNS}}$ implying that $I_{3\text{DNS}}$ is a yes-instance of 3-DENSITY NWS. □

Before we get to the proofs of NP-hardness for STARS NWS and CONNECTIVITY NWS, we make an observation about the relation of these problems. We observe that in subgraphs induced by communities of size at most 3 the property of containing a spanning star is equivalent to the property of being connected. This means that the problems 3-STARS NWS and 3-CONNECTIVITY NWS are essentially the same in this case. Furthermore, we observe that each instance of 3-STARS NWS and 3-CONNECTIVITY NWS can be expressed as an instance of 3-DENSITY NWS by assigning each community a density requirement of $\frac{2}{3}$. These observations are summarized by Lemma 3.4.

**Lemma 3.4.** *Let $G = (V, E)$ be an undirected graph, $\mathcal{C}$ a set of communities, $\alpha : \mathcal{C} \to (\frac{1}{3}, \frac{2}{3}]$ a density requirement and $\ell \in \mathbb{N}$. Let $I_{3\text{DNS}} = (G, \mathcal{C}, \alpha, \ell)$ be an instance of* 3-Density NWS*, let $I_{3\text{SNS}} = (G, \mathcal{C}, \ell)$ be an instance of* 3-Stars NWS*, and let $I_{3\text{CNS}} = (G, \mathcal{C}, \ell)$ be an instance of* 3-Connectivity NWS*. Then, the instances $I_{3\text{DNS}}$, $I_{3\text{SNS}}$, and $I_{3\text{CNS}}$ are equivalent.*

*Proof.* A subgraph induced by a community satisfies the condition in each problem variant if and only if it has at least two edges:

- A graph on three vertices whose density is contained in the interval $(\frac{1}{3}, \frac{2}{3}]$ has at least two edges.
- A graph on three vertices contains a size-two star if and only if it has at least two edges.
- A graph on three vertices is connected if and only if it has at least two edges.

□

Fan et al. [14] showed NP-hardness of the Interconnection Graph Problem which can be seen as a special case of Connectivity NWS where the graph in the input is a clique. The hardness applies even if the instances are restricted to communities of size at most 3 [14]. With Lemma 3.4 in mind, this directly implies NP-hardness of Connectivity NWS and Stars NWS. In the following proof, we recall their construction for the sake of completeness.

**Theorem 3.5.** Stars NWS *and* Connectivity NWS *are NP-hard even if restricted to communities with size at most* 3.

*Proof.* We reduce from Vertex Cover. Let $I_{\text{VC}} = (G = (V, E), k)$ be an instance of Vertex Cover. We start by defining the graph $G_{3\text{CNS}} = (V_{3\text{CNS}}, E_{3\text{CNS}})$ with $V_{3\text{CNS}} \coloneqq V \cup \{z\}$ and $E_{3\text{CNS}} \coloneqq \{\{v, x\} \mid v \in V\} \cup E$. We define the set of communities $\mathcal{C} \coloneqq \{\{u, v, z\}, \{u, v\} \mid \{u, v\} \in E\}$. Observe that we create for each edge a size-three community and a size-two community and each size-three community induces a triangle. Due to the size-two communities every valid sparsified graph $G'_{3\text{CNS}}$ contains all edges not adjacent to $z$. We set the parameter $\ell \coloneqq k + |E|$. Finally, let $I_{3\text{CNS}} = (G_{3\text{CNS}}, \mathcal{C}, \ell)$ denote the resulting instance of 3-Connectivity NWS. An example of the construction is shown in Figure 3.3.

**Correctness** We show that $I_{\text{VC}}$ is a yes-instance of Vertex Cover if and only if $I_{3\text{CNS}}$ is a yes-instance of 3-Connectivity NWS.

($\Rightarrow$) Let $X$ be a vertex cover of size at most $k$ of $G$. We show how to obtain a sparsified graph $G'_{3\text{CNS}} = (V_{3\text{CNS}}, E'_{3\text{CNS}})$ with $|E'_{3\text{CNS}}| \leq |X| + |E|$. We set $E'_{3\text{CNS}} \coloneqq \{\{x, z\} \mid x \in X\} \cup E$, noting that $E'_{3\text{CNS}} \leq |X| + |E|$. For

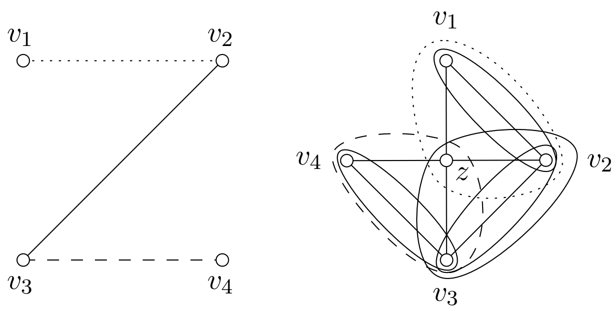


Figure 3.3: An example of the construction. The left side shows the VERTEX COVER instance $I_{\text{VC}}$, the right side shows the 3-CONNECTIVITY NWS instance $I_{\text{3CNS}}$. The different line styles show which community of the 3-CONNECTIVITY NWS instance corresponds to which edge of the VERTEX COVER instance.

each community $C_i = \{u, v, z\} \in \mathcal{C}$, we observe that $\{u, v\} \in E'_{\text{3CNS}}$ and either $\{u, z\} \in E'_{\text{3CNS}}$ or $\{v, z\} \in E'_{\text{3CNS}}$ because $X$ is a vertex cover implying $u \in X$ or $v \in X$. Hence, for each community $C_i \in \mathcal{C}$, the induced subgraph $G'_{\text{3CNS}}[C_i]$ is connected because $|E(G'_{\text{3CNS}}[C_i])| \geq 2$. Therefore, $G'_{\text{3CNS}}$ is a sparsified graph implying that $I_{\text{3CNS}}$ is a yes-instance of 3-CONNECTIVITY NWS..

($\Leftarrow$) Let $I_{\text{3CNS}}$ be a yes-instance of 3-CONNECTIVITY NWS and let $G'_{\text{3CNS}} = (V_{\text{3CNS}}, E'_{\text{3CNS}})$ be a sparsified graph. We show how to obtain a vertex cover $X$ of $G$ with $|X| \leq |E'_{\text{3CNS}}| - |E|$. We set $X := \{x \mid \{x, z\} \in E'_{\text{3CNS}}\}$. Note that $|X| \leq |E'_{\text{3CNS}} \setminus E| = |E'_{\text{3CNS}}| - |E|$ because $E \subset E'_{\text{3CNS}}$ due to the size-two communities in $\mathcal{C}$. For each edge $\{u, v\} \in E$, we observe that $u \in X$ or $v \in X$: There exists a community $C_i = \{u, v, z\} \in \mathcal{C}$ such that $G'_{\text{3CNS}}[C_i]$ is connected. This implies $\{u, z\} \in E'_{\text{3CNS}}$ or $\{v, z\} \in E'_{\text{3CNS}}$. Therefore, $X$ is vertex cover of $G$ with $|X| \leq |E'_{\text{3DNS}}| - |E|$ and $I_{\text{VC}}$ a yes-instance of VERTEX COVER. □

Next, we analyze the complexity of Π-NWS restricted to communities of size at most 3 in general. In the following, we characterize a graph property Π by the maximum set $\mathcal{G}$ of graphs with three vertices which fulfill Π. Since two graphs with three vertices are isomorphic if and only if they have the same number of edges, it is sufficient to replace the set $\mathcal{G}$ with the set of edge counts of the graphs in $\mathcal{G}$. For a graph property Π, we denote its characterization with $c(\Pi)$. For example, the property of being connected in a graph with three vertices is characterized by the set $\{2, 3\}$ because a graph with three vertices is connected if and only if it has at least two edges. There are 15 different characterizations of graph properties in graphs with three vertices because there are 15 non-empty subsets of $\{0, 1, 2, 3\}$.

Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of Π-NWS where each community is of size at most 3. We begin with the characterizations of graph properties for which Π-NWS is in P.

First, we observe that the instance $I$ is trivially solvable if $0 \in c(\Pi)$, that is if the graph property Π is fulfilled in a graph without any edges. This is the case because the graph $G' = (V, \emptyset)$ is a sparsified graph with at most $\ell$ edges for the instance $I$ of Π-NWS. Second, we observe that the instance $I$ is trivially solvable if $c(\Pi) = \{3\}$, that is if the graph property Π is only fulfilled by a triangle in graphs with three vertices. This is the case because it only has to be checked whether $|E| \leq \ell$ and whether each subgraph induced by a community is a triangle.

We continue with the characterizations of graph properties for which Π-NWS is NP-hard. The characterization $\{1, 2, 3\}$ is equal to the characterization of a density requirement of 1/3. Hence, the NP-hardness for this characterization is implied by Theorem 3.1, the NP-hardness of 3-Density NWS. Since the reduction in the proof of Theorem 3.1 constructs only instance such that each size-three community contains two edges, NP-hardness for graph properties characterized by $\{1, 2\}$ is also implied. As mentioned before, the property of being connected in graphs with three vertices is characterized by $\{2, 3\}$. Therefore, NP-hardness for graph properties characterized by $\{2, 3\}$ is implied by Theorem 3.5, the NP-hardness of Connectivity NWS.

For each of the remaining characterizations $\{1\}, \{2\}$, and $\{1, 3\}$, we give reduction from a different variant of 3-SAT. In our constructions below, $X$ denotes the set of variables of the input formula $\phi$, $L := \{x, \overline{x} \mid x \in X\}$ denotes the set of literals, and $C$ is the set of clauses of $\phi$ such that each clause $c \in C$ is a size-three subset of $L$. Furthermore, $E_X$ denotes the edges of the variable gadget, $E_C$ denotes the edges of the clause gadget, and $E_+$ denotes the edges to connect both gadgets. Analogous, $\mathcal{C}_X$ denotes the communities of the variable gadget, $\mathcal{C}_C$ denotes the communities of the clause gadget, and $\mathcal{C}_+$ denotes the communities to connect both gadgets.

Before we present the reductions, we give some intuition of what all three constructions have in common. For each variable $x \in X$, there is an edge $e_x$ representing the positive literal and an edge $e_{\overline{x}}$ representing the negative literal. The intuition is that only one of the edges $e_x$ and $e_{\overline{x}}$ can be contained in a sparsified graph. If the edge $e_x$ is contained in the sparsified graph, then the variable $x$ is set to `true`. Otherwise the edge $e_{\overline{x}}$ is contained in the sparsified graph and the variable $x$ is set to `false`. In other words, these edges encode the assignment. Moreover, for each clause $c = \{p, q, r\} \in C$, there are three edges $f_p^c$, $f_q^c$ and $f_r^c$ each representing a literal of clause $c$. The variable gadget and the clause gadget are connected in a way that the edge $f_p^c$ is contained in the sparsified graph if and only if the edge $e_p$ is contained in the sparsified graph. In other words, the construction transfers the assignment encoded in the variable gadget to the clause gadget.

**Theorem 3.6.** Π-NWS *restricted to communities of size at most* 3 *is NP-*

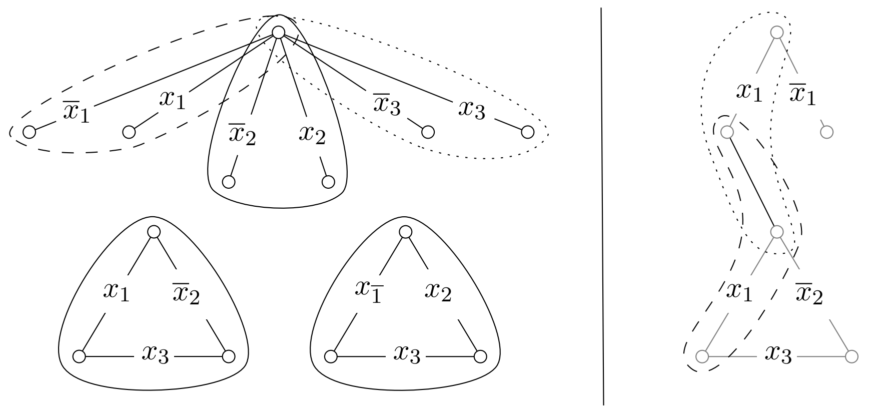


Figure 3.4: An example of the construction. The left side shows the variable gadget and the clause gadget for the formula $\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3)$. The right side shows how the variable gadget and clause gadget are connected for the literal $x_1$ of the clause $(x_1 \vee \overline{x}_2 \vee x_3)$.

*hard for graph properties characterized by* $\{1\}$.

*Proof.* We give a reduction from 1-IN-3-SAT, a variant of 3-SAT which is also NP-hard [15] and asks whether there exists an assignment for an input formula $\phi$ setting for each clause exactly one literal to `true`. In the following we describe the variable gadget, then we describe the clause gadget and finally we describe how both gadgets are connected.

**Variable Gadget** We add a vertex $z$ and for each variable $x \in X$, we add two vertices $u_x, u_{\overline{x}}$, the edges $e_x = \{u_x, z\}, e_{\overline{x}} = \{z, u_{\overline{x}}\}$, and a community $C_x = \{u_x, z, u_{\overline{x}}\}$.

**Clause Gadget** For each clause $c = \{p, q, r\} \in C$, we add three vertices $v_p^c, v_q^c, v_r^c$, the edges $f_p^c = \{v_p^c, v_q^c\}, f_q^c = \{v_q^c, v_r^c\}, f_r^c = \{v_r^c, v_p^c\}$, and a community $C_c = \{c_p, c_q, c_r\}$.

**Connecting the Gadgets** For each $c = \{p, q, r\} \in C$, we add the edges $g_p^c = \{v_p^c, u_p\}$, $g_q^c = \{v_q^c, u_q\}$, $g_r^c = \{v_r^c, u_r\}$ and the communities $\{z, u_p, v_p^c\}$, $\{u_p, v_p^c, v_q^c\}$, $\{z, u_q, v_q^c\}$, $\{u_q, v_q^c, v_r^c\}$, $\{z, u_r, v_r^c\}$, $\{u_r, v_r^c, v_p^c\}$.

Finally, let $G = (V, E)$ be the graph and let $\mathcal{C}$ be the set of communities resulting from the above construction. We set $\ell := |X| + 3|C|$. Let $I = (G, \mathcal{C}, \ell)$ be the final instance of $\Pi$-NWS. An example of the construction is shown in Figure 3.4.

**Correctness** We show that there exists an assignment setting exactly one literal per clause of $\phi$ to `true` if and only if $I$ is a yes-instance of $\Pi$-NWS.

($\Rightarrow$) Let $A : X \to \{0,1\}$ be an assignment setting exactly one literal to `true` for each clause of $\phi$. We describe how to obtain a sparsified graph $G' = (V, E')$ with $|E'| \leq |X| + 3|C|$ using the assignment $A$. The edges in $G'$ of the variable gadget are $E'_X := \{e_x \mid x \in X, A(x) = 1\} \cup \{e_{\overline{x}} \mid x \in X, A(x) = 0\}$. Note that $|E'_X| = |X|$. The edges in $G'$ of the clause gadget are $E'_C := \{f^c_p \mid \text{literal } p \text{ of clause } c \text{ is } \texttt{true} \text{ for assignment } A\}$. Observe that $|E'_C| = |C|$ because for each clause exactly one literal is set to `true`. The edges in $G'$ of the connection of both gadgets are $E'_+ := \{g^c_p \mid \text{literal } p \text{ of clause } c \text{ is } \texttt{false} \text{ for assignment } A\}$. Observe that $|E'_+| = 2|C|$ because for each clause exactly two literals are set to `false`. Finally, the edges of the sparsified graph are $E' := E'_X \cup E'_C \cup E'_+$, thus $|E'| = |X| + 3|C|$.

It remains to show that $|E(G'[C_i])| = 1$ for each community $C_i \in \mathcal{C}$. For each community $C_i \in \mathcal{C}_X$ of the variable gadget, we have $|E(G'[C_i])| = 1$ because either $e_x \in E'_X$ or $e_{\overline{x}} \in E'_X$ for a variable $x \in X$. For each community $C_i \in \mathcal{C}_C$ of the clause gadget, we have $|E(G'[C_i])| = 1$ because either $f^c_p \in E'_C$ or $f^c_q \in E'_C$ or $f^c_r \in E'_C$ for a clause $c = \{p, q, r\} \in C$. For each community $C_i \in \mathcal{C}_+$ of the connection of both gadgets, we have $|E(G'[C_i])| = 1$ because $g^c_p \in E_+$ if and only if $f^c_p \notin E'_C$ and $e_p \notin E'_X$. Hence, $I$ is a yes-instance of $\Pi$-NWS.

($\Leftarrow$) Let $I$ be a yes-instance of $\Pi$-NWS and let $G' = (V, E')$ be a sparsified graph. First, we specify the properties of the sparsified graph $G'$.

**Claim 3.7.** *Let $x \in X$ be a variable, let $c \in C$ be a clause, and let $p \in c$ be a literal of clause $c$. The sparsified graph $G'$ has the following properties:*

1. *either $e_x \in E'$ or $e_{\overline{x}} \in E'$*
2. *either $f^c_p \in E'$ or $f^c_q \in E'$ or $f^c_r \in E'$*
3. *$e_p \in E'$ if and only if $g^c_p \notin E'$ if and only if $f^c_p \in E'$*

*Proof.* For the first property, recall that all communities $C_X$ of the variable gadget are edge-disjoint. Since $|E(G'[C_i])| = 1$ for each community $C_i \in C_X$ of the variable gadget, this implies that either $e_x \in E'$ or $e_{\overline{x}} \in E'$.

For the second property, recall that all communities $C_C$ of the clause gadget are edge-disjoint. Since $|E(G'[C_i])| = 1$ for each community $C_i \in C_C$ of the clause gadget, this implies that either either $f^c_p \in E'$ or $f^c_q \in E'$ or $f^c_r \in E'$.

For the third property, recall that there is a length-three path consisting of the edges $e_p, g^c_p, f^c_p$. Furthermore, there exists two communities $C'$ and $C''$ with $E(G[C']) = \{g^c_p, e_p\}$ and $E(G[C'']) = \{g^c_p, f^c_p\}$. Since $|E(G'[C'])| = 1$, $e_p \in E'$ is equivalent to $g^c_p \notin E'$ which is equivalent to $f^c_p \in E'$ because also $|E(G'[C''])| = 1$. Thus, $e_p \in E'$ if and only if $g^c_p \notin E'$ if and only if $f^c_p \in E'$. $\diamondsuit$

Next, we define an assignment $A$ based on $G'$. By Claim 3.7, we know that either $e_x \in E'$ or $e_{\overline{x}} \in E'$.

We set $A : X \to \{0,1\}$ with $x \mapsto \begin{cases} 0 & e_{\overline{x}} \in E' \\ 1 & e_x \in E' \end{cases}$.

Note that the assignment $A$ assigns each variable $x \in X$ a unique value. Finally, we show that the assigment $A$ sets exactly one literal to `true` for each clause $c \in C$. By Claim 3.7, we know that either $f_p^c \in E'$ or $f_q^c \in E'$ or $f_r^c \in E'$ for each clause $c = \{p, q, r\} \in C$. Without loss of generality, we assume $f_p^c \in E'$ which implies $e_p \in E'$ by Claim 3.7. Since $A$ is defined that the literal $p$ is `true`, only the literal $p$ of clause $c$ is set to `true`. Hence, the assignment $A$ sets for each clause $c \in C$ exactly one literal to `true`. □

**Theorem 3.8.** Π-NWS *restricted to communities of size at most* 3 *is NP-hard for graph properties characterized by* $\{2\}$.

*Proof.* We give a reduction from 2-IN-3-SAT, a variant of 3-SAT which is also NP-hard [15] and asks whether there exists an assignment such that for each clause exactly two literals are set to `true`.

First, we introduce the notion of a permanent edge. An edge $e \in E$ is *permanent*, if every sparsified graph $G'$ of a yes-instance of Π-NWS contains this edge $e$. In our construction, any edge $\{u, v\}$ can be made permanent by adding a new vertex $w$, the edge $\{u, w\}$ and the community consisting of $u, v, w$. Since a graph property characterized by $\{2\}$ enforces exactly two edges in a subgraph induced by a community of size 3, every sparsified graph $G'$ of a yes-instance of Π-NWS has to contain the edges $\{u, v\}$ and $\{v, w\}$. Thus, for a graph property characterized by $\{2\}$, Π-NWS with permanent edges is polynomial time reducible to Π-NWS. Hence, in this case it is sufficient to show NP-hardness for Π-NWS with permanent edges. In the following we describe the variable gadget, then we describe the clause gadget and finally we describe how both gadgets are connected.

**Variable Gadget** We add a vertex $z$ and for each variable $x \in X$, we add two vertices $u_x, u_{\overline{x}}$, the edges $e_x = \{u_x, z\}, e_{\overline{x}} = \{z, u_{\overline{x}}\}$, the permanent edge $\{u_x, u_{\overline{x}}\}$, and a community $C_x = \{u_x, z, u_{\overline{x}}\}$.

**Clause Gadget** For each clause $c = \{p, q, r\} \in C$, we add three vertices $v_p^c, v_q^c, v_r^c$, the edges $f_p^c = \{v_p^c, v_q^c\}, f_q^c = \{v_q^c, v_r^c\}, f_r^c = \{v_r^c, v_p^c\}$, and a community $C_c = \{c_p, c_q, c_r\}$.

**Connecting the Gadgets** For each $c = \{p, q, r\} \in C$, we add the edges $g_p^c = \{v_p^c, u_p\}$, $g_q^c = \{v_q^c, u_q\}$, $g_r^c = \{v_r^c, u_r\}$, the permanent edges $g_p'^c = \{z, v_p^c\}$, $g_p''^c = \{u_p, v_q^c\}$, $g_q'^c = \{z, v_q^c\}$, $g_q''^c = \{u_q, v_r^c\}$, $g_r'^c = \{z, v_r^c\}$, $g_r''^c = \{u_r, v_p^c\}$, and the communities $\{z, u_p, v_p^c\}$, $\{u_p, v_p^c, v_q^c\}$, $\{z, u_q, v_q^c\}$, $\{u_q, v_q^c, v_r^c\}$, $\{z, u_r, v_r^c\}$, $\{u_r, v_r^c, v_p^c\}$.

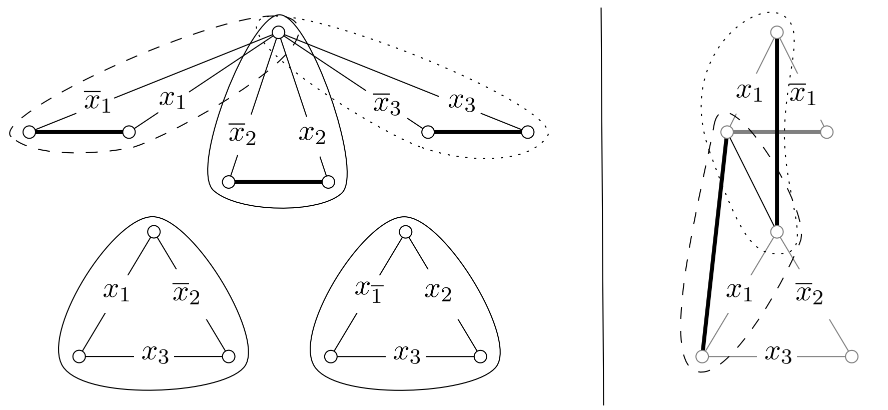


Figure 3.5: An example of the construction. The left side shows the variable gadget and the clause gadget for the formula $\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3)$. The right side shows how the variable gadget and clause gadget are connected for the literal $x_1$ of the clause $(x_1 \vee \overline{x}_2 \vee x_3)$. The bold edges indicates the permanent edges which are enforced to be contained in each sparsified graph.

Finally, let $G = (V, E)$ be the graph and let $\mathcal{C}$ be the set of communities resulting from the above construction. We set $\ell := 2|X| + 9|C|$. Let $I = (G, \mathcal{C}, \ell)$ be the final instance of Π-NWS. An example of the construction is shown in Figure 3.5.

**Correctness** We show that there exists an assignment setting exactly two literals per clause of $\phi$ to `true` if and only if $I$ is a yes-instance of Π-NWS.

($\Rightarrow$) Let $A : X \to \{0, 1\}$ be an assignment setting exactly two literals to `true` for each clause of $\phi$. We describe how to obtain a sparsified graph $G' = (V, E')$ with $|E'| \leq 2|X| + 9|C|$ using the assignment $A$. The edges in $G'$ of the variable gadget are $E'_X := \{e_x \mid x \in X, A(x) = 1\} \cup \{e_{\overline{x}} \mid x \in X, A(x) = 0\} \cup \{\text{the permanent edges of } E_X\}$. Note that $|E'_X| = 2|X|$. The edges in $G'$ of the clause gadget are $E'_C := \{f^c_p \mid$ literal $p$ of clause $c$ is `true` for assignment $A\}$. Observe that $|E'_C| = 2|C|$ because for each clause exactly two literals are set to `true`. The edges in $G'$ of the connection of both gadgets are $E'_+ := \{g^c_p \mid$ literal $p$ of clause $c$ is `false` for assignment $A\} \cup \{\text{the permanent edges of } E_+\}$. Observe that $|E'_+| = |C| + 6|C|$ because for each clause exactly one literal is set to `false` and the connection of both gadgets has $6|C|$ permanent edges. Finally, the edges of the sparsified graph are $E' := E'_X \cup E'_C \cup E'_+$, thus $|E'| = 2|X| + 9|C|$.

It remains to show that $|E(G'[C_i])| = 2$ for each community $C_i \in \mathcal{C}$. For each community $C_i \in \mathcal{C}_X$ of the variable gadget, we have $|E(G'[C_i])| = 2$ because $G'[C_i]$ contains the permanent edge $\{u_x, u_{\overline{x}}\}$ and either $e_x \in E'_X$ or $e_{\overline{x}} \in E'_X$ for a variable $x \in X$. For each community $C_i \in \mathcal{C}_C$ of the clause gadget, we have $|E(G'[C_i])| = 2$ because either $f_p^c, f_q^c \in E'_C$ or $f_q^c, f_r^c \in E'_C$ or $f_r^c, f_p^c \in E'_C$ for a clause $c = \{p, q, r\} \in C$. For each community $C_i \in \mathcal{C}_+$ of the connection of both gadgets, we have $|E(G'[C_i])| = 2$ because $G'[C_i]$ contains either the permanent edge $g_p'^c$ or the permanent edge $g_p''^c$ and $g_p^c \in E_+$ if and only if $f_p^c \notin E'_C$ and $e_p \notin E'_X$. Hence, $I$ is a yes-instance of Π-NWS.

(⇐) Let $I$ be a yes-instance of Π-NWS and let $G' = (V, E')$ be the sparsified graph. First, we specify the properties of the sparsified graph $G'$.

**Claim 3.9.** *Let $x \in X$ be a variable, let $c \in C$ be a clause, and let $p \in c$ be a literal of clause $c$. The sparsified graph $G'$ has the following properties:*

1. *either $e_x \in E'$ or $e_{\overline{x}} \in E'$*
2. *either $f_p^c, f_q^c \in E'$ or $f_q^c, f_r^c \in E'$ or $f_r^c, f_p^c \in E'$*
3. *$e_p \in E'$ if and only if $g_p^c \notin E'$ if and only if $f_p^c \in E'$*

*Proof.* For the first property, recall that all communities $C_X$ of the variable gadget are edge-disjoint. Since $|E(G'[C_i])| = 2$ for each community $C_i \in C_X$ of the variable gadget, this implies that either $e_x \in E'$ or $e_{\overline{x}} \in E'$ because the edge $\{u_x, u_{\overline{x}}\}$ is permanent.

For the second property, recall that all communities $C_C$ of the clause gadget are edge-disjoint. Since $|E(G'[C_i])| = 2$ for each community $C_i \in C_C$ of the clause gadget, this implies that either $f_p^c, f_q^c \in E'$ or $f_q^c, f_r^c \in E'$ or $f_r^c, f_p^c \in E'$.

For the third property, recall that there is a length-three path consisting of the edges $e_p$, $g_p^c, f_p^c$. Furthermore, there exists two communities $C'$ and $C''$ with $E(G[C']) = \{g_p^c, e_p, g_p'^c\}$ and $E(G[C'']) = \{g_p^c, f_p^c, g_p''^c\}$. Since $|E(G'[C'])| = 2$ and the edge $g_p'^c$ is permanent, $e_p \in E'$ is equivalent to $\hat{e}_p^c \notin E'$ which is equivalent to $e_p^c \in E'$ because also $|E(G'[C''])| = 2$ and the $g_p''^c$ is also permanent. Thus, $e_p \in E'$ if and only if $g_p^c \notin E'$ if and only if $f_p^c \in E'$. ♢

Next, we define an assignment $A$ based on $G'$. By Claim 3.9, we know that either $e_x \in E'$ or $e_{\overline{x}} \in E'$.

We set $A : X \to \{0, 1\}$ with $x \mapsto \begin{cases} 0 & e_{\overline{x}} \in E' \\ 1 & e_x \in E' \end{cases}$.

Note that the assignment $A$ assigns each variable $x \in X$ a unique value. Finally, we show that the assigment $A$ sets exactly two literals to `true` for each clause $c \in C$. By Claim 3.9, we know that either $f_p^c, f_q^c \in E'$ or $f_q^c, f_r^c \in E'$ or $f_r^c, f_p^c \in E'$ for each clause $c = \{p, q, r\} \in C$. Without loss of generality, we assume $f_p^c, f_q^c \in E'$ which implies $e_p, e_q \in E'$ by Claim 3.9. Since $A$ is defined that the literal $p$ and $q$ are `true`, exactly two literals are set to `true` for clause $c$. Hence, the assignment $A$ sets exactly two literals to `true` for each clause $c \in C$. □

**Theorem 3.10.** Π-NWS *restricted to communities of size at most* 3 *is NP-hard for graph properties characterized by* $\{1, 3\}$.

*Proof.* We give a reduction from NAE3SAT, a variant of 3-SAT which is also NP-hard [15] and asks whether there exists an assignment such that for each clause not all literals have the same truth value. In the following we describe the variable gadget, then we describe the clause gadget and finally we describe how both gadgets are connected.

**Variable Gadget** We add a vertex $z$ and for each variable $x \in X$, we add two vertices $u_x, u_{\overline{x}}$, the edges $e_x = \{u_x, z\}, e_{\overline{x}} = \{z, u_{\overline{x}}\}$, and a community $C_x = \{u_x, z, u_{\overline{x}}\}$.

**Clause Gadget** For each clause $c = \{p, q, r\} \in C$, we add four vertices $v_p^c, v_q^c, v_r^c, v_z^c$, all six possible edges $f_p^c = \{v_p^c, v_z^c\}$, $f_q^c = \{v_q^c, v_z^c\}$, $f_r^c = \{v_r^c, v_z^c\}$, $f_p'^c = \{v_q^c, v_r^c\}$, $f_q'^c = \{v_p^c, v_r^c\}$, $f_r'^c = \{v_p^c, v_q^c\}$ and three communities $C_1^c = \{v_p^c, v_q^c, v_z^c\}$, $C_2^c = \{v_q^c, v_r^c, v_z^c\}$, and $C_3^c = \{v_p^c, v_r^c, v_z^c\}$.

**Connecting the Gadgets** For each $c = \{p, q, r\} \in C$, we add the edges $g_p^c = \{v_p^c, u_p\}$, $g_q^c = \{v_q^c, u_q\}$, $g_r^c = \{v_r^c, u_r\}$ and the communities $\{z, u_p, v_p^c\}$, $\{u_p, v_p^c, v_q^c\}$, $\{z, u_q, v_q^c\}$, $\{u_q, v_q^c, v_r^c\}$, $\{z, u_r, v_r^c\}$, $\{u_r, v_r^c, v_p^c\}$.

Finally, let $G = (V, E)$ be the graph and let $\mathcal{C}$ be the set of communities resulting from the above construction. We set $\ell := |X| + 4|C|$. Let $I = (G, \mathcal{C}, \ell)$ be the final instance of Π-NWS. An example of the construction is shown in Figure 3.6.

**Correctness** We show that there is an assignment setting one or two literals per clause of $\phi$ to `true` if and only if $I$ is a yes-instance of Π-NWS.

($\Rightarrow$) Let $A : X \to \{0, 1\}$ be an assignment setting exactly one or two literals to `true` for each clause of $\phi$. Let $a$ be the number of clauses for which exactly one literal is set to `true` and let $b$ be the number of clauses for which exactly two literals are set to `true`. We describe how to obtain the sparsified graph $G' = (V, E')$ with $|E'| \leq |X| + 4|C|$ using the assignment $A$. The edges in $G'$ of the variable gadget are $E'_X := \{e_x \mid x \in X, A(x) = 1\} \cup \{e_{\overline{x}} \mid x \in X, A(x) = 0\}$. Note that $|E'_X| = |X|$. Recall that for each clause exactly two literals are set to `true`. The edges in $G'$ of the clause gadget are $E'_C := \{f_p^c, f_p'^c \mid \text{ in clause } c \text{ only literal } p \text{ is } \texttt{true} \text{ for assignment } A\} \cup \{f_p^c, f_q^c, f_r'^c \mid \text{ in clause } c \text{ the literals } p, q \text{ are } \texttt{true} \text{ for assignment } A\}$. Note that $|E'_C| = 2a + 3b$ because for $a$ clauses exactly one literal is set to `true` and for $b$ clauses exactly two literals are set to `true`. The edges in $G'$ of the connection of both gadgets are $E'_+ := \{g_p^c \mid \text{literal } p \text{ of clause } c \text{ is } \texttt{false} \text{ for assignment } A\}$. Observe that $|E'_+| = b + 2a$ because for $b$ clauses exactly one literal is

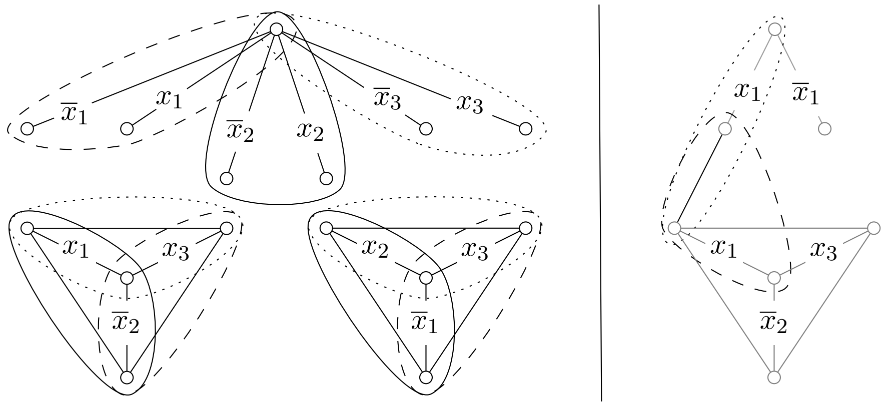


Figure 3.6: An example of the construction. The left side shows the variable gadget and the clause gadget for the formula $\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3)$. The right side shows how the variable gadget and clause gadget are connected for the literal $x_1$ of the clause $(x_1 \vee \overline{x}_2 \vee x_3)$.

set to `false` and for $a$ clauses exactly two literals are set to `false`. Finally, the edges of the sparsified graph are $E' := E'_X \cup E'_C \cup E'_+$, thus $|E'| = |X| + 4a + 4b = |X| + 4|C|$.

It remains to show that $|E(G'[C_i])| = 1$ or $|E(G'[C_i])| = 3$ for each community $C_i \in \mathcal{C}$. For each community $C_i \in \mathcal{C}_X$ of the variable gadget, we have $|E(G'[C_i])| = 1$ because either $e_x \in E'_X$ or $e_{\overline{x}} \in E'_X$ for a variable $x \in X$. For each community $C_i \in \mathcal{C}_+$ of the connection of both gadgets, we have $|E(G'[C_i])| = 1$ because $g^c_p \in E_+$ if and only if $f^c_p \notin E'_C$ and $e_p \notin E'_X$. Recall that there are three communities $C^c_1$, $C^c_2$, $C^c_3$ for each clause $c \in C$. The clause $c$ has either exactly one `true` literal or exactly two `true` literals. In the first case, we observe that $|E(G'[C^c_1])| = 1$, $|E(G'[C^c_2])| = 1$, and $|E(G'[C^c_3])| = 1$. In the second case, we observe that either $|E(G'[C^c_1])| = 3$ or $|E(G'[C^c_2])| = 3$ or $|E(G'[C^c_3])| = 3$ and that each subgraph induced by one of the other two communities contains exactly one edge. Thus, for each community $C_i \in \mathcal{C}_X$ of the clause gadget we have $|E(G'[C_i])| = 1$ or $|E(G'[C_i])| = 3$. Hence, $I$ is a yes-instance of Π-NWS.

($\Leftarrow$) Let $I$ be a yes-instance of Π-NWS and let $G' = (V, E')$ be a sparsified graph. First, we specify the properties of the sparsified graph $G'$.

**Claim 3.11.** *Let $x \in X$ be a variable, let $c \in C$ be a clause, and let $p \in c$ be a literal of clause $c$. The sparsified graph $G'$ has the following properties:*

1. *either $e_x \in E'$ or $e_{\overline{x}} \in E'$*
2. *neither $f^c_p, f^c_q, f^c_r \in E'$ nor $f^c_p, f^c_q, f^c_r \notin E'$*
3. *$e_p \in E'$ if and only if $g^c_p \notin E'$ if and only if $f^c_p \in E'$*

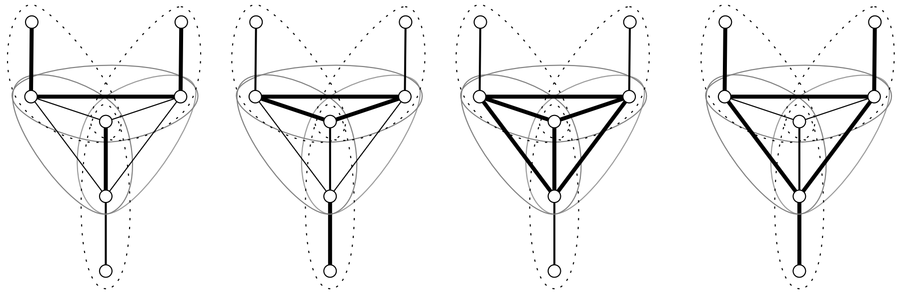

Figure 3.7: All four variants, how the graph property characterized by $\{1, 3\}$ can be satisfied in the communities of the clause gadget. (Note that the two possibilities which are isomorphic to the second one are omitted) The grey communities are the communities of the clause gadget. The dotted communities are a part of connection to the variable gadget. The bold edges represent the edges of a sparsified graph fulfilled the graph property characterized by $\{1, 3\}$ for all shown communities.

*Proof.* For the first property, recall that all communities $C_X$ of the variable gadget are edge-disjoint. Since $|E(G'[C_i])| = 1$ for each community $C_i \in C_X$ of the variable gadget, this implies that either $e_x \in E'$ or $e_{\overline{x}} \in E'$.

For the second property, recall that there are three communities $C_1^c$, $C_2^c$, $C_3^c$ for a clause $c \in C$. In Figure 3.7, all variants are shown, how a graph property characterized by $\{1, 3\}$ can be fulfilled in these three communities. Observe that the first and the second variant only need four edges. In contrast to this, the third and the fourth variant need six edges. Since $|X|$ edges in the sparsified graph are needed for the variable gadget, there are at most $4|C|$ edges of the clause gadget and the connection of both gadgets in $G'$. Hence, only the first and the second variant are possible in $G'$ because otherwise $G'$ has more than $|X| + 4|C|$ edges. This implies that neither $f_p^c, f_q^c, f_r^c \in E'$ nor $f_p^c, f_q^c, f_r^c \notin E'$.

For the third property, recall that there is a length-three path consisting of the edges $e_p$, $g_p^c, f_p^c$. Furthermore, there exists two communities $C'$ and $C''$ with $E(G[C']) = \{g_p^c, e_p\}$ and $E(G[C'']) = \{g_p^c, f_p^c\}$. Since $|E(G'[C'])| = 1$, $e_p \in E'$ is equivalent to $g_p^c \notin E'$ which is equivalent to $f_p^c \in E'$ because also $|E(G'[C''])| = 1$. Thus, $e_p \in E'$ if and only if $g_p^c \notin E'$ if and only if $f_p^c \in E'$. ♢

Next, We define an assignment $A$ based on $G'$. By Claim 3.11, we know that either $e_x \in E'$ or $e_{\overline{x}} \in E'$.

We set $A : X \to \{0, 1\}$ with $x \mapsto \begin{cases} 0 & e_{\overline{x}} \in E' \\ 1 & e_x \in E' \end{cases}$.

Note that the assignment $A$ assigns each variable $x \in X$ a unique value. Finally, we show that the assigment $A$ sets one or two literals to `true` for each clause $c \in C$. By Claim 3.11, we know that either one of edges $f_p^c, f_q^c, f_r^c$ is contained in $E'$ or two of these edges are contained in $E'$. First, we consider the case that exactly one of the edges $f_p^c, f_q^c, f_r^c$ is contained in $E'$. Without loss of generality, we assume $f_p^c \in E'$ which implies $e_p \in E'$ by Claim 3.11. Since $A$ is defined that the literal $p$ is `true`, the clause $c$ is satisfied by literal $p$. Second, we consider the case that exactly two of the edges $f_p^c, f_q^c, f_r^c$ are contained in $E'$. Without loss of generality, we assume $f_p^c, f_q^c \in E'$ which implies $e_p, e_q \in E'$ by Claim 3.11. Since $A$ is defined that the literal $p$ and $q$ are `true`, two literals of the clause $c$ are set to `true`. Hence, the assignment $A$ sets for each clause exactly one or exactly two literals to `true`. □

This completes our study of the NP-hardness of Density NWS, Stars NWS, Connectivity NWS, and Π-NWS restricted to communities of size at most 3.

## 3.2 Lower bounds

The naive brute force approach for each Π-NWS problem performs an exhaustive search over all subsets of edge set $E$ in the input graph $G = (V, E)$. This are $|\mathcal{P}(E)| = 2^m$ sets. With the restriction to graph properties Π which are verifiable in $O(\text{poly}(n + m))$ time, we are able to obtain the following general statement for Π-NWS problems.

**Theorem 3.12.** *Every* Π-NWS *problem is solvable in* $O(2^m \cdot |\mathcal{C}| \cdot \text{poly}(n + m))$ *time.*

*Proof.* Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of a Π-NWS problem. The algorithm is as follows. Check for each graph $G' := (V, E')$ with $E' \subseteq E$ and $|E'| \leq \ell$ whether each subgraph induced by a community $C_i \in \mathcal{C}$ fulfills the property Π. If such a graph $G'$ exists, then $I$ is a yes-instance of Π-NWS. If no such graph $G'$ exists, then $I$ is a no-instance of Π-NWS. Note that we assumed that the graph property Π is verifiable in $\text{poly}(n + m)$ time for each subgraph induced by a community. Since there are at most $2^m$ different subsets of $E$, the described algorithm has a running time of $O(2^m \cdot |\mathcal{C}| \cdot \text{poly}(n + m)))$. □

In undirected graphs the number of edges $m$ is at most $\binom{n}{2}$. Therefore, the exponential factor in the running time $O(2^m \cdot |\mathcal{C}| \cdot \text{poly}(n+m))$ stated in Theorem 3.12 is also expressible by parameter $n$ as $O(2^{n^2} \cdot |\mathcal{C}| \cdot \text{poly}(n+m))$. Since many real-world graphs in the field of social network analysis are sparse, it would be tempting to find an algorithm with a running time such that the exponent is for example only linear in the parameter $n$ instead of quadratic. This is not possible for Density NWS, Stars NWS, and Connectivity NWS assuming the Exponential Time Hypothesis. Even if the community size is restricted to at most 4, an algorithm with such a running time does not exist for Density NWS, Stars NWS, and Connectivity NWS when the ETH is true as shown in the following theorem.

**Theorem 3.13.** *If the ETH is true, then* Density NWS*,* Stars NWS*, and* Connectivity NWS *cannot be solved in* $2^{o(n^2)} \cdot \text{poly}(n+|\mathcal{C}|)$ *time even if restricted to instances with community size at most 4.*

*Proof.* We start with a reduction from 3-SAT to 4-Density NWS such that the resulting graph has $O(\sqrt{|\phi|})$ vertices and $O(\text{poly}(|\phi|))$ communities, where $\phi$ denotes the input formula and $|\phi|$ the sum of the number of variables and clauses. Then, the existence of an $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$-time algorithm for 4-Density NWS implies the existence of an $2^{o(|\phi|)}$-time algorithm for 3-SAT defeating the ETH [19, 20]. Then, we adapt the reduction to 4-Stars NWS and 4-Connectivity NWS. Without loss of generality we assume that each clause consists of exactly three literals. In the construction below $X$ denotes the set of variables, $L \coloneqq \{x, \overline{x} \mid x \in X\}$ the set of literals containing a positive and negative literal for each variable and $C \subseteq \{C_i \subseteq L \mid |C_i| = 3\}$ the set of clauses. For $\ell \in L$ we say that $\overline{\ell}$ denotes the complement literal of the same variable.

**Variable Gadget** We start by describing the construction of the variable gadget $G_X = (V_X, E_X)$. The idea is to create for each variable a community containing a $P_3$ with a density requirement of $\frac{1}{3}$. The first edge in such a $P_3$ represents the positive literal, and the second one the negative literal. The density requirement of $\frac{1}{3}$ is used to model that one literal must be set to `true`. These $P_3$s are arranged compactly, to achieve a variable gadget with $|V_X| \in O\left(\sqrt{|\phi|}\right)$. Let $\theta : L \to E_X$ be the mapping of the literals to the edges in the variable gadget. This mapping will be populated throughout the following construction.

Let $G_X$ be a complete balanced bipartite graph where both disjoint independent sets $U = \{u_1, \ldots, u_{n_x}\}$ and $V = \{v_1, \ldots, v_{n_x}\}$ consist of $n_x = 2\lceil\sqrt{|X|}\rceil$ vertices each. Observe that $n_x$ is dividable by 2. Then, we define a partition $V' \coloneqq \{\{v_{i \cdot 2-1}, v_{i \cdot 2}\} \mid i \in \{1, \ldots, \frac{n_x}{2}\}\}$ of $V$ into sets of size 2. The induced subgraph used for representing each variable is a $P_3$. Such a $P_3$ consists of one vertex $u \in U$ being its center and two vertices $v_i, v_j \in V$.

**Claim 3.14.** *There exists an injection between $X$ and $U \times V'$.*

*Proof.* There exists an injection because $|U \times V'| = |U| \cdot |V'| = n_x \cdot \frac{n_x}{2} = 2\lceil\sqrt{|X|}\rceil \cdot \frac{2\lceil\sqrt{|X|}\rceil}{2} \geq 2\sqrt{|X|} \cdot \sqrt{|X|} = 2|X| \geq |X|$. ♢

By Claim 3.14 let $\eta : X \to U \times V'$ be such an injection. We use this injection to assign each variable a $P_3$. Then, we add for each $x \in X$ with $\eta(x) = (u_i, \{v_i, v_j\})$ a community $C_x = \{u_i, v_i, v_j\}$ with $\alpha(C_x) = \frac{1}{3}$. We denote these communities with $\mathcal{C}^X$. Finally, we set $\theta(x) \coloneqq \{u_i, v_i\}$ and

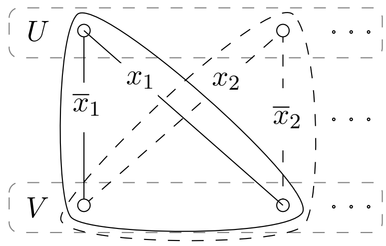


Figure 3.8: An example of the construction showing how the variables are represented and how the $P_3$s are arranged

$\theta(\overline{x}) := \{u_i, v_j\}$ to assign the positive and negative literal of $x$ an edge of the variable gadget. An example of a variable gadget is shown in Figure 3.8.

**Claim 3.15.** *Each edge of $E_X$ is contained in only one subgraph induced by a community in $\mathcal{C}_X$.*

*Proof.* Each community $C_x$ consists of one vertex $u \in U$ and two vertices $v_i, v_j \in V$ forming a $P_3$ with $u$ as center. Suppose there are two different communities $C_{x_1}$ and $C_{x_2}$ such that they have one edge in common. This implies that $C_{x_1}$ and $C_{x_2}$ contain the same vertex of $U$ and have one vertex of $V$ in common. This leads to a contradiction because $V'$ is a partition. ♢

**Clause Gadget** We continue by describing the construction of the clause gadget $G_C = (V_C, E_C)$. The idea is to create a size-three star for each clause in the formula. Each edge represents one literal of the clause. For each such star, we create three communities each consisting of a different $P_3$ in the star with a density requirement of $\frac{1}{3}$. Again, these induced subgraphs are arranged compactly, to achieve a clause gadget with $|V_C| \in O\left(\sqrt{|C|}\right)$. Let $\nu : \{(c, \ell_1), (c, \ell_2), (c, \ell_3) \mid c = \{\ell_1, \ell_2, \ell_3\} \in C\} \to E_C$ be the mapping of the literal occurrences in clauses to the edges of the clause gadget. This mapping will be populated throughout the following construction.

Let $G_C$ be also a complete balanced bipartite graph where both disjoint independent sets $Y = \{y_1, \ldots, y_{n_c}\}$ and $Z = \{z_1, \ldots, z_{n_c}\}$ consists of $n_c = 3\lceil\sqrt{|C|}\rceil$ vertices each. Note that $n_c$ is dividable by 3. Then, we define a partition $Z' := \{z_{i \cdot 3-2}, z_{i \cdot 3-1}, z_{i \cdot 3} \mid i \in \{1, \ldots, \frac{n_c}{3}\}\}$ of $Z$ into sets of size 3.

**Claim 3.16.** *There exists an injection between $C$ and $Y \times Z'$.*

*Proof.* There exists an injection because $|Y \times Z'| = |Y| \cdot |Z'| = n_c \cdot \frac{n_c}{3} = 3\lceil\sqrt{|C|}\rceil \cdot \frac{3\lceil\sqrt{|C|}\rceil}{3} \geq 3\sqrt{|C|} \cdot \sqrt{|C|} = 3|C| \geq |C|$. ♢

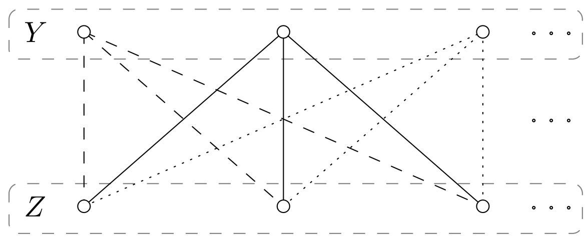


Figure 3.9: An example of the construction showing how the stars representing the clauses are arranged.

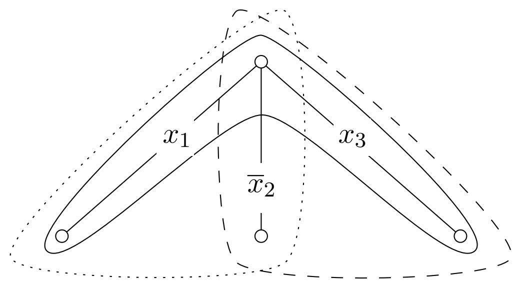


Figure 3.10: An example of the clause construction showing how the clause $x_1 \vee \overline{x}_2 \vee x_3$ is represented.

By Claim 3.16, let $\mu : C \to Y \times Z'$ be an injection. We use this injection to assign each clause a star of size three. Then, we create for each clause $c = \{\ell_1, \ell_2, \ell_3\} \in C$ with $\mu(c) = (y_i, \{z_i, z_j, z_k\})$ three communities $C_c^1 = \{y_i, z_i, z_j\}$, $C_c^2 = \{y_i, z_i, z_k\}$ and $C_c^3 = \{y_i, z_j, z_k\}$ with $\alpha(C_c^1) = \alpha(C_c^2) = \alpha(C_c^3) = \frac{1}{3}$. We denote these communities with $\mathcal{C}^C$. Finally, we set $\nu(c, \ell_1) := \{y_i, z_i\}$, $\nu(c, \ell_2) := \{y_i, z_j\}$, and $\nu(c, \ell_3) := \{y_i, z_k\}$ to assign each literal in clause $c$ an edge of the clause gadget. An example of a clause gadget is shown in Figure 3.9. In Figure 3.10 an example is shown how a clause is represented.

**Claim 3.17.** *All subgraphs induced by the vertex sets in* $W = \{C_c^1 \cup C_c^2 \cup C_c^3 \mid c \in C\}$ *are pairwise edge-disjoint.*

*Proof.* Each vertex set $W_i \in W$ consists of one vertex $y \in Y$ and three vertices $z_i, z_j, z_k \in Z$ forming a star with $y$ as center. Suppose there are two different vertex sets $W_1, W_2 \in W$ such that they have at least one edge in common. This implies that $W_1$ and $W_2$ contain the same vertex of $Y$ and have one vertex of $Z$ in common. This leads to a contradiction because $Z'$ is a partition. ♢

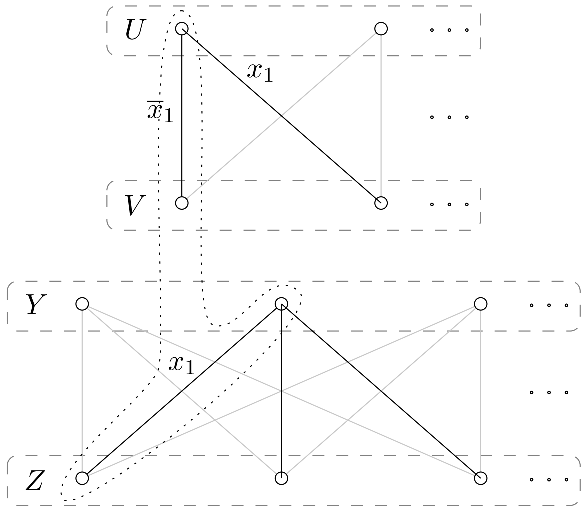


Figure 3.11: An example of the construction showing how a literal in a clause is connected with the variable gadget

**Connecting the Gadgets** We complete the construction by describing how the variable and clause gadget are connected using communities. The idea is to put the endpoints of an edge describing a literal in the clause together with the endpoints of the edge describing the opposite literal in the variable gadget in a community. These communities are used to model occurrences of variables in the clauses. We create for each clause $c = \{\ell_1, \ell_2, \ell_3\} \in C$ three communities $C_c^{\ell_1} = \nu(c, \ell_1) \cup \theta(\overline{\ell_1})$, $C_c^{\ell_2} = \nu(c, l_2) \cup \theta(\overline{\ell_2})$ and $C_c^{\ell_3} = \nu(c, \ell_3) \cup \theta(\overline{\ell_3})$ with $\alpha(C_c^{\ell_1}) = \alpha(C_c^{\ell_2}) = \alpha(C_c^{\ell_3}) = \frac{1}{6}$. We denote these communities with $\mathcal{C}_C^X$. An example of the connection for one literal is shown in Figure 3.11.

Finally, let $G_{4\text{DNS}} = G_X \cup G_C$ be the graph, let $\mathcal{C}_{4\text{DNS}} = \mathcal{C}_X \cup \mathcal{C}_C \cup \mathcal{C}_C^X$ be the set of communities and let $\alpha$ be the mapping resulting from the above construction.

**Correctness** We show that the formula $\phi$ is satisfiable if and only if $I_{4\text{DNS}} = (G_{4\text{DNS}}, C_{4\text{DNS}}, \alpha, |X| + 2|C|)$ is a yes-instance of 4-Density NWS.

Before we proof this statement, we make an observation about the parameter $l$.

**Claim 3.18.** *For $l < |X| + 2|C|$, the instance $I = (G_{4\text{DNS}}, C_{4\text{DNS}}, \alpha, l)$ is a no-instance of* 4-Density NWS.

*Proof.* At least one edge of each community in the variable gadget is contained in the sparsified graph to fulfill their density requirement. Concluding from Claim 3.15, these are at least $|X|$ edges. At least two edges of each triple of communities forming a star of size four in the clause gadget are contained in the sparsified graph to fulfill their density requirement. Concluding from Claim 3.17, these are at least $2|C|$ edges. This implies that $I = (G_{4\text{DNS}}, C_{4\text{DNS}}, \alpha, l)$ is a no-instance of 4-Density NWS for $l < |X| + 2|C|$. $\diamondsuit$

($\Rightarrow$) Let $A : X \to \{0, 1\}$ be an assignment satisfying $\phi$. First, we describe how to obtain the sparsified graph $G' = (V, E')$ with $|E'| = |X| + 2|C|$ using $A$. For each variable, we select the edge of the variable gadget representing the literal that is not fulfilled by $A$, formally described by $E'_X := \{\theta(x) \mid x \in X, A(x) = 0\} \cup \{\theta(\overline{x}) \mid x \in X, A(x) = 1\}$. The set $E'_X$ fulfills the density requirement of the communities in the variable gadget. Let $E'_C$ denote a set of edges. For each clause $c = \{\ell_1, \ell_2, \ell_t\}$, there is at least one literal that is fulfilled by $A$ which is denoted by $l_t$. We add the edges $\nu(c, l_1)$ and $\nu(c, l_2)$ to $E'_C$. In the construction of the clause gadget, we created three communities $C^1_c = \nu(c, \ell_1) \cup \nu(c, \ell_2)$ , $C^2_c = \nu(c, \ell_1) \cup \nu(c, \ell_t)$, and $C^3_c = \nu(c, \ell_2) \cup \nu(c, \ell_t)$. Each of them having a density requirement of $\frac{1}{3}$ which is fulfilled by the edges $\nu(c, l_1)$ and $\nu(c, l_2)$. While connecting both gadgets, we also created three communities $C^{\ell_1}_c = \nu(c, \ell_1) \cup \theta(\ell_1)$, $C^{\ell_2}_c = \nu(c, \ell_2) \cup \theta(\ell_2)$, and $C^{\ell_3}_c = \nu(c, \ell_t) \cup \theta(\ell_t)$, each of them having a density requirement of $\frac{1}{3}$. This density requirement is fulfilled for $C^{\ell_1}_c$ and $C^{\ell_2}_c$ by the edges $\nu(c, \ell_1)$ and $\nu(c, \ell_2)$. For $C^{\ell_t}_c$ the density requirement is also fulfilled but this time by an edge in $E'_X$. Finally, we set $E' := E'_X \cup E'_C$ and observe that $|E'| = |X| + 2|C|$. Therefore, $I_{4\text{DNS}}$ is a yes-instance of 4-Density NWS.

($\Leftarrow$) First, we define an assignment $A$ based on the sparsified graph $G = (V, E')$ where $E'_X$ denotes the edges in $G'$ of the variable gadget and $E'_C$ the edges in $G'$ of the clause gadget. Then, we show that the assignment $A$ satisfies $\phi$.

We set $A : X \to \{0, 1\}$ with $x \mapsto \begin{cases} 0 & \theta(\overline{x}) \notin E'_X \\ 1 & \theta(x) \notin E'_X \end{cases}$.

By the proof of Claim 3.18, we conclude that $|E'_X| = k$ and not both edges $\theta(x)$ and $\theta(\overline{x})$ of a variable $x$ are contained in $E'_X$. Therefore, the assignment $A$ does assign each variable $x$ a unique value. Recall that we created for each clause $c = \{\ell_1, \ell_2, \ell_t\}$ three communities $C^{\ell_1}_c$, $C^{\ell_2}_c$, and $C^{\ell_t}_c$ connecting the variable and the clause gadget. Due to the solution size of $|X| + 2|C|$, we have $E' \cap \{\nu(c, \ell_1), \nu(c, \ell_2), \nu(c, \ell_t)\} = 2$. Hence, the density requirement of one of the communities $C^{\ell_1}_c$, $C^{\ell_2}_c$ and $C^{\ell_t}_c$ has to be fulfilled by an edge contained in $E'_X$. Without loss of generality we assume that this is the case for $C^{\ell_t}_c$. This implies that $\theta(\overline{\ell_t}) \in E'_X$ and therefore $A(\ell_t) = 1$ which satisfies the clause $c$. This applies to all clauses, wherefore $\phi$ is satisfied by $A$.

**Adapting the construction for Stars NWS** Now, we modify the construction to be able to replace the density requirement for each community with the star requirement. The idea is to add edges to each community such that these edges form together with the sparsified graph $G'_{4\text{DNS}}$ of the 4-Density NWS instance a sparsified graph $G'_{4\text{SNS}}$ for 4-Stars NWS.

**Variable Gadget** We make the set $V$ a clique instead of an independent set. This transforms every $P_3$ in the variable gadget to a triangle. Then, we create for each edge $\{v_i, v_j\}$ in the clique $V$ a community consisting only of $v_i$ and $v_j$.

**Clause Gadget** We make the set $Z$ a clique instead of an independent set. This transforms every $P_3$ in the clause gadget to a triangle. Then, we create for each edge $\{z_i, z_j\}$ in the clique $Z$ a community consisting only of $z_i$ and $z_j$.

**Connecting the Gadgets** For each community $\{u, v, y, z\} \in C_C^X$ with $u \in U$, $v \in V$, $y \in Y$ and $z \in Z$ that has been created to connect both gadgets, we add the edges $\{u, z\}$, $\{v, y\}$ and $\{v, z\}$. Then, we create for each added edge a community consisting of the endpoints of the edge.

Finally, let $G_{4\text{SNS}}$ denote the adapted graph and let $\mathcal{C}_{4\text{SNS}}$ denote the adapted set of communities resulting from the modified construction. Furthermore, let $I_{4\text{SNS}} = (G_{4\text{SNS}}, \mathcal{C}_{4\text{SNS}}, |X| + 2|C| + 3|C| + \binom{|V|}{2} + \binom{|Z|}{2})$ denote the instance of Stars NWS resulting from the modified construction.

**Claim 3.19.** *$I_{4\text{SNS}}$ is a yes-instance of* 4-Stars NWS *if and only if $I_{4\text{DNS}}$ is a yes-instance of* 4-Density NWS*.*

*Proof.* Throughout the proof we call an edge $e \in E(G_{4\text{SNS}})$ *old* if $e \in E(G_{4\text{DNS}})$ and *new* if $e \notin E(G_{4\text{DNS}})$.

($\Rightarrow$) Let $I_{4\text{SNS}}$ be a yes-instance of 4-Stars NWS and let $G'_{4\text{SNS}}$ be a sparsified graph. We show how to construct a sparsified graph for the instance $I_{4\text{DNS}}$ of 4-Density NWS with $|X| + 2|C|$ edges. Observe that all new edges are contained in $G'_{4\text{SNS}}$ due to the size-two communities. These are $\binom{|V|}{2}$ edges for the variable gadget, $\binom{|Z|}{2}$ edges for the clause gadget, and $3|C|$ edges for the connection between both gadgets. Next, observe that the new edges almost form a complete star in each subgraphs induced by the size-three and size-four communities. To form a complete star at least one old edge in each subgraphs induced by these communities is contained in $G'_{4\text{SNS}}$. These are $|X| + 2|C|$ old edges. We remove the new edges from $G'_{4\text{SNS}}$. The remaining graph contains $|X|+2|C|$ old edges and each subgraph induced by a size-three or size-four community contains at least one edge. It satisfies the density requirement of each community in the instance $I_{4\text{DNS}}$. Therefore, the resulting graph is a sparsified graph for the instance $I_{4\text{DNS}}$. Thus, $I_{4\text{DNS}}$ is a yes-instance of 4-Density NWS.

(⇐) Let $I_{\text{4DNS}}$ be a yes-instance of 4-Density NWS and let $G'_{\text{4DNS}}$ be a sparsified graph. We show how to construct a sparsified graph for the instance $I_{\text{4SNS}}$ of 4-Stars NWS with $|X| + 2|C| + 3|C| + \binom{|V|}{2} + \binom{|Z|}{2}$ edges. Observe that each subgraph induced by a size-three or size-four community contains at least one old edge. We add all new edges to $G'_{\text{4DNS}}$ which are $3|C| + \binom{|V|}{2} + \binom{|Z|}{2})$ edges. By the definition of the modified construction we know that these new edges form together with the old edges in each subgraph induced by a size-three and size-four community a star. The star requirement of the subgraphs induced by the additional size-two communities in $G'_{\text{4SNS}}$ is also satisfied by the new edges. Therefore, the extended graph is a sparsified graph for the instance $I_{\text{4SNS}}$ of 4-Stars NWS and has $|X| + 2|C| + 3|C| + \binom{|V|}{2} + \binom{|Z|}{2}$ edges. Thus, $I_{\text{4SNS}}$ is a yes-instance of 4-Stars NWS. ♢

**Correctness** By the correctness of the reduction to 4-Density NWS and Claim 3.19, we conclude that the formula $\phi$ is satisfiable if and only if $I_{\text{4SNS}}$ is a yes-instance of 4-Stars NWS.

**Adapting the construction for Connectivity NWS** Now, we modify the construction to be able to replace the density requirement for each community with the requirement of being connected. Again, the idea is to add edges to each community such that these edges form together with the sparsified graph $G'_{\text{4DNS}}$ of the 4-Density NWS instance a sparsified graph $G'_{\text{4CNS}}$ for 4-Connectivity NWS.

**Variable Gadget** We make the set $V$ a clique instead of an independent set. This transforms every $P_3$ in the variable gadget to a triangle. Then, we create for each edge $\{v_i, v_j\}$ in the clique $V$ a community consisting only of $v_i$ and $v_j$.

**Clause Gadget** We make the set $Z$ a clique instead of an independent set. This transforms every $P_3$ in the clause gadget to a triangle. Then, we create for each edge $\{z_i, z_j\}$ in the clique $Z$ a community consisting only of $z_i$ and $z_j$.

**Connecting the Gadgets** For each community $\{u, v, y, z\} \in C_C^X$ with $u \in U$, $v \in V$, $y \in Y$ and $z \in Z$ that has been created to connect both gadgets, we add the edges $\{u, z\}$ and $\{v, y\}$. Then, we create for each added edge a community consisting of the endpoints of the edge.

Finally, let $G_{\text{4CNS}}$ denote the adapted graph and let $\mathcal{C}_{\text{4CNS}}$ denote the adapted set of communities resulting from the modified construction. Furthermore, let $I_{\text{4CNS}} = (G_{\text{4CNS}}, \mathcal{C}_{\text{4CNS}}, |X| + 2|C| + 2|C| + \binom{|V|}{2} + \binom{|Z|}{2})$ denote the instance of CONNECTIVITY NWS resulting from the modified construction.

**Claim 3.20.** *$I_{4\text{CNS}}$ is a yes-instance if and only if $I_{4\text{DNS}}$ is a yes-instance.*

*Proof.* The proof is analogous to the proof of Claim 3.19. Again, we call an edge $e \in E(G_{4\text{CNS}})$ *old* if $e \in E(G_{4-SD})$ and *new* if $e \notin E(G_{4-SD})$.

($\Rightarrow$) Let $I_{4\text{CNS}}$ be a yes-instance of 4-CONNECTIVITY NWS and let $G'_{4\text{CNS}}$ be a sparsified graph. We show how to construct a sparsified graph for the instance $I_{4\text{DNS}}$ of 4-DENSITY NWS with $|X| + 2|C|$ edges. Observe that all new edges are contained in $G'_{4\text{CNS}}$ due to the size-two communities. These are $\binom{|V|}{2}$ edges for the variable gadget, $\binom{|Z|}{2}$ edges for the clause gadget and $3|C|$ edges for the connection between both gadgets. Next, observe that each subgraph induced by a size-three or size-four community is almost connected by the new edges. At least one old edge in each subgraphs induced by these communities is contained in $G'_{4\text{CNS}}$ to connect them completely. These are $|X|+2|C|$ old edges. We remove the new edges from $G'_{4\text{CNS}}$. The remaining graph contains $|X| + 2|C|$ old edges and each subgraph induced by a size-three or size-four community contains at least one edge. It satisfies the density requirement of each community the instance $I_{4\text{DNS}}$. Therefore, the resulting graph is a sparsified graph for the instance $I_{4\text{DNS}}$. Thus, $I_{4\text{DNS}}$ is a yes-instance of 4-DENSITY NWS.

($\Leftarrow$) Let $I_{4\text{DNS}}$ be a yes-instance and let $G'_{4\text{DNS}}$ be a sparsified graph. We show how to construct a sparsified graph for the instance $I_{4\text{CNS}}$ with $|X| + 2|C| + 2|C| + \binom{|V|}{2} + \binom{|Z|}{2}$ edges implying $I_{4\text{CNS}}$ is a yes-instance. Observe that each subgraph induced by a size-three or size-four community contains at least one old edge. We add all new edges to $G'_{4\text{DNS}}$ which are $2|C| + \binom{|V|}{2} + \binom{|Z|}{2})$ edges. By the definition of the modified construction we know that these new edges connects together with the old edges each subgraph induced by a size-three or size-four community. The subgraphs induced by the additional size-two communities in $G'_{4\text{CNS}}$ are also connected. Therefore, the extended graph is a sparsified graph for the instance $I_{4\text{CNS}}$ of 4-CONNECTIVITY NWS and has $|X| + 2|C| + 2|C| + \binom{|V|}{2} + \binom{|Z|}{2}$ edges. Thus, $I_{4\text{CNS}}$ is a yes-instance of 4-CONNECTIVITY NWS. ◇

**Correctness** By the correctness of the reduction to 4-DENSITY NWS and Claim 3.19, we conclude that the formula $\phi$ is satisfiable if and only if $I_{4\text{CNS}}$ is a yes-instance of 4-CONNECTIVITY NWS. □

This bound also answers the question, whether it is possible to improve the exponent to be only sub-linear in parameter $m$ to get an algorithm with a running time of $2^{o(m)} \cdot |\mathcal{C}| \cdot \text{poly}(n + m)$. Under the Exponential Time Hypothesis such an algorithm does not exist for DENSITY NWS, STARS NWS, and CONNECTIVITY NWS which directly follows from Theorem 3.13.

**Corollary 3.21.** *If the ETH is true, then* DENSITY NWS*,* STARS NWS *and* CONNECTIVITY NWS *cannot be solved in* $2^{o(m)} \cdot |\mathcal{C}| \cdot \text{poly}(n+m)$ *time.*

*Proof.* Since $m \leq n^2$, the existence of an algorithm with a running time of $2^{o(m)} \cdot |\mathcal{C}| \cdot \text{poly}(n+m)$ implies the existence of an algorithm with a running time of $2^{o(n^2)} \cdot |\mathcal{C}| \cdot \text{poly}(n+m)$. By Theorem 3.13 we conclude that this would defeat the ETH. □

# 4. Parameterized Complexity

In the following, we study DENSITY NWS, STARS NWS, and CONNECTIVITY NWS using the framework of parameterized complexity.

## 4.1 The Parameter $\ell$

In this section, we analyze DENSITY NWS, STARS NWS and CONNECTIVITY NWS parameterized by $\ell$, the number of edges of the sparsified graph. In Section 3.1, we saw that DENSITY NWS, STARS NWS, and CONNECTIVITY NWS are NP-hard even if the instances are restricted to those with having communities of size at most 3. Because of this, we start our analysis for parameter $\ell$ with instances having communities of size at most 3. We start with 3-DENSITY NWS followed by 3-STARS NWS and 3-CONNECTIVITY NWS. Then, we study the more general case, where the community size is bounded by an arbitrary integer $d$. Finally, we analyze the problems with unbounded community size parameterized by $\ell$.

A maximum community size of at most 3 implies that each subgraph induced by a community has at most three edges. This restricts the search tree size of an exhaustive search on the edge set to at most $O(3^\ell)$. This allows us to obtain fixed-parameter tractability for 3-DENSITY NWS as shown in Theorem 4.1.

**Theorem 4.1.** 3-DENSITY NWS *is solvable in* $O(3^\ell \cdot |\mathcal{C}| \cdot (n+m))$ *time.*

*Proof.* We start by giving an algorithm. Let $I = (G, \mathcal{C}, \alpha, \ell)$ be an instance of 3-DENSITY NWS. The call *Solve3DNWS*$(G, \mathcal{C}, \alpha, \ell, \emptyset)$ of Algorithm 1 solves the instance $I$ by performing an exhaustive search on the edge set $E$ to find the edge set $E'$ of a sparsified graph. In each recursive call, a community $C_i$ is selected whose density requirement is not already satisfied by the edges in $E'$. If no such community exists, then $I$ is a yes-instance. If such a community exists, but $\ell = 0$, then $I$ is a no-instance. Then, the algorithm branches on the edges in the subgraph induced by $C_i$ that are not already contained in $E'$. For each branch, such an edge is added to $E'$ and the parameter $\ell$ is decreased by one accordingly.

**Algorithm 1:** Algorithm for 3-Density NWS: *Solve3DNWS*

**Input** : $G = (V, E), \mathcal{C}, \alpha, \ell, E'$
**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no
1 $\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid \frac{1}{3} \cdot |E' \cap E(G[C_i])| < \alpha(C_i)\}$
2 **if** $\mathcal{C} = \emptyset$ **then**
3 | **return** $G' = (V, E')$
4 **if** $\ell = 0$ **then**
5 | **return** *no*
6 $C_i \leftarrow$ pick element from $\mathcal{C}$
7 **forall** $e \in E(G[C_i]) \setminus E'$ **do**
8 | **if** *Solve3DNWS*$(G, \mathcal{C}, \alpha, \ell - 1, E' \cup \{e\})$ *returns a graph* $G'$ **then**
9 | | **return** $G'$
10 **return** *no*

**Running time** We continue by analyzing the running time of the described algorithm. The loop in Lines 7–9 iterates over the edges in a subgraph induced by a community which are at most 3 due to the community size of 3. Therefore, the recursive call in Line 8 is made at most three times, each call decreasing the parameter $\ell$ by one. This leads to a branching vector of $[1, 1, 1]$ resulting in a search tree size of at most $O(3^\ell)$. Checking the termination condition and selecting the next community $C_i$ in Lines 1-6 takes at most $O(|\mathcal{C}| \cdot (n+m))$ time. Hence, the algorithm has a total running time of $O(3^\ell \cdot |\mathcal{C}| \cdot (n + m))$. □

Theorem 2 introduces an algorithm for solving 3-Density NWS. In Lemma 3.4, we saw that 3-Stars NWS and 3-Connectivity NWS are essentially special cases of 3-Density NWS. This allows us to transfer the fixed-parameter tractability of 3-Density NWS to the other two problems.

**Corollary 4.2.** 3-Stars NWS *and* 3-Connectivity NWS *are solvable in* $O(3^\ell \cdot |\mathcal{C}| \cdot (n + m))$ *time.*

*Proof.* Let $I = (G, \mathcal{C}, \ell)$ be an instance of 3-Stars NWS or an instance of 3-Connectivity NWS. Let $\alpha : \mathcal{C} \to \{\frac{2}{3}\}$ be the constant mapping of $\mathcal{C}$ to $\frac{2}{3}$. According to Lemma 3.4, the instance $I_{\text{3DNS}} = (G, \mathcal{C}, \alpha, \ell)$ is an equivalent instance of 3-Density NWS which is solvable in $O(3^\ell \cdot |\mathcal{C}| \cdot (n + m))$ time according to Theorem 4.1. □

Algorithm 2 performs a naive branching without considering the structure of an instance like different density requirements of different communities. Next, we introduce two reduction rules and one branching rule leading to a better algorithm. We define the reduction rules based on the branching algorithm. We omit the straightforward correctness proofs. The observation made by Lemma 3.2 leads directly to Reduction Rule 1. Lemma 3.2

describes edges which have definitely to be kept in a sparsified graph in order to satisfy the density requirement of a community. This are edges of the subgraphs induced by a community which have already at least edges as possible to still satisfy its density requirement.

**Reduction Rule 1.** *Let $G, \mathcal{C}, \alpha, \ell$, and $E'$ be the current parameters of a recursive step of Algorithm 1. If there exists a community $C_i \in \mathcal{C}$ with $|E(G[C_i])| = \lceil \binom{|C_i|}{2} \cdot \alpha(C_i) \rceil$, then add the edges in $E(G[C_i])$ to $E'$, remove $C_i$ from $\mathcal{C}$ and decrease $\ell$ accordingly.*

Next, we consider communities not sharing an edge with any other community. Let $G = (V, E)$ be a graph and let $\mathcal{C}$ be a set of communities over $V$. We say a community $C_i \in \mathcal{C}$ is *independent*, if there does not exist a community $C_j \in \mathcal{C} \setminus \{C_i\}$ with $E(G[C_i]) \cap E(G[C_j]) \neq \emptyset$. An independent community does not affect any other communities and therefore can be solved independently. This leads to the following reduction rule.

**Reduction Rule 2.** *Let $G, \mathcal{C}, \alpha, \ell$, and $E'$ be the current parameters of a recursive step of the branching algorithm. If there exists an independent community $C_i \in \mathcal{C}$, then add the minimum number of edges in $E(G[C_i])$ to $E'$ such that $|E(G[C_i]) \cap E'| \geq \alpha(C_i) \cdot \binom{|C_i|}{2}$, remove $C_i$ from $\mathcal{C}$, and decrease $\ell$ accordingly.*

The following branching rule exploits the existence of a size-three community $C_i$ with a density requirement $\alpha(C_i) \in (\frac{1}{3}, \frac{2}{3}]$.

**Branching Rule 1.** *Let $G, \mathcal{C}, \alpha, \ell$ and $E'$ be the current parameters of a recursive step of the branching algorithm. If there exists a community $C_i = \{u, v, w\} \in \mathcal{C}$ with $|E(G[C_i])| = 3$, $E(G[C_i]) \cap E' = \emptyset$ and $\alpha(C_i) \in (\frac{1}{3}, \frac{2}{3}]$, then branch into the following cases:*

1. *Add $\{u, v\}$ and $\{v, w\}$ to $E'$ and decrease $\ell$ by 2.*
2. *Add $\{v, w\}$ and $\{u, w\}$ to $E'$ and decrease $\ell$ by 2.*
3. *Add $\{u, v\}$ and $\{u, w\}$ to $E'$ and decrease $\ell$ by 2.*

Branching Rule 1 has a branching vector of $[2, 2, 2]$ which results a branching number of approximately 1.733. Using Reduction Rules 1–2 and Branching Rule 1, we are able to improve the algorithm for 3-Density NWS.

**Theorem 4.3.** 3-Density NWS *is solvable in $O(2.076^{\ell} \cdot |\mathcal{C}| \cdot (n+m))$ time.*

*Proof.* We adapt the branching of Algorithm 1. The resulting algorithm is shown in Algorithm 2. First, we apply Reduction Rules 1–2 exhaustively. In the branching, we apply the Branching Rule 1 exhaustively.

**Claim 4.4.** *If neither the Reduction Rules 1–2 nor the Branching Rule 1 apply anymore, then one of the following statements applies for each remaining community $C_i \in \mathcal{C}$:*

1. *If $\alpha(C_i) \in (0, \frac{1}{3}]$ then $|E(G[C_i]) \cap E'| = 0$*
2. *If $\alpha(C_i) \in (\frac{1}{3}, \frac{2}{3}]$ then $|E(G[C_i]) \cap E'| = 1$*

*Proof.* Let $C_i \in \mathcal{C}$ be a community and neither Reduction Rules 1 and 2 nor Branching Rule 1 are applicable.. In case of $\alpha(C_i) \in (\frac{2}{3}, 1]$ Reduction Rule 1 applies. In case of $\alpha(C_i) = 0$ the community $C_i$ has been removed from $\mathcal{C}$ in Line 2. Hence, we observe that $\alpha(C_i) \in (0, \frac{2}{3}]$. First, we consider the case when $\alpha(C_i) \in (0, \frac{1}{3}]$. Then, we observe that $|E(G[C_i]) \cap E'| = 0$ because otherwise the density requirement $\alpha(C_i)$ is already satisfied by the edges in $E'$. Next, we consider the case when $\alpha(C_i) \in (\frac{1}{3}, \frac{2}{3}]$. In case of $|E(G[C_i]) \cap E'| \geq 2$ the density requirement $\alpha(C_i)$ is already satisfied by the edges in $E'$ and the community $C_i$ is removed from $\mathcal{C}$ in Line 2. In case of $|E(G[C_i]) \cap E'| = 0$, we observe that $|E(G[C_i])| = 3$ because in case of $|E(G[C_i])| = 2$ Reduction Rule 1 applies and in case of $|E(G[C_i])| < 2$ the density requirement $a(C_i)$ is not even satisfied in $G$. Therefore, in case of $|E(G[C_i]) \cap E'| = 0$ Branching Rule 1 is applicable. Thus, we have $|E(G[C_i]) \cap E'| = 1$. ♢

By Claim 4.4 we know that for each remaining community $C_i \in \mathcal{C}$ the density requirement $\alpha(C_i)$ is satisfied by $E'$ except one edge. Thus, the remaining task is to select a subset of $E^+ \subseteq E \setminus E'$ such that $E^+ \cap C_i \neq \emptyset$ for each community $C_i \in \mathcal{C}$. Since each community has at most three edges, the remaining task can be expressed as an instance of 3-Hitting Set. We define the equivalent 3-Hitting Set instance $I_{3-\mathrm{HS}} = (U, \mathcal{S}, \ell)$, where the universe is defined as $U := E \setminus E'$ and the family of subset over $U$ is defined as $\mathcal{S} := \{E(G[C_i]) \setminus E' \mid C_i \in \mathcal{C}\}$. If $\mathcal{S}$ has a hitting set $X$ of size at most $\ell$, then $G' = (V, E' \cup X)$ is a sparsified graph with at most $\ell$ edges. Otherwise, $I$ is a no-instance.

**Running time** The branching number of Branching Rule 1 is 1.733. Let $\ell'$ be the number of edges in $E'$ that originated from Branching Rule 1. Hence, the search tree created by this rule has a size of at most $O(1.733^{\ell'})$. The Hitting Set instance created in Line 10 is solvable in $O(2.076^{\ell-\ell'} \cdot |\mathcal{C}|)$time [27]. This leads to an overall running time of $O(2.076^{\ell} \cdot |\mathcal{C}| \cdot (n+m))$ for solving an instance of 3-Density NWS. □

We could apply Algorithm 2 also to 3-Stars NWS and 3-Connectivity NWS as we did in Corollary 4.2 with Algorithm 1. But this algorithm does not exploit that in the sparsified graph each subgraph induced by a community has at least two edges. Using this property we are able to obtain

**Algorithm 2:** Refined Algorithm for 3-Density NWS

**Input :** $G = (V, E), \mathcal{C}, \alpha, \ell, E'$
**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no

*Apply Reduction Rules 1-2 exhaustively*
$\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid \frac{1}{3} \cdot |E' \cap E(G[C_i])| < \alpha(C_i)\}$
**if** $\mathcal{C} = \emptyset$ **then**
  **return** $G' = (V, E')$
**if** $\ell = 0$ **then**
  **return** *no*
**if** *Branching Rule 1 is applicable* **then**
  *Apply Branching Rule 1*
**else**
  $I_{3-\text{HS}} \leftarrow (E \setminus E', \{C_i \setminus E' \mid C_i \in \mathcal{C}\}, \ell)$
  **if** $I_{3-\text{HS}}$ *is a yes-instance with hitting set* $X$ **then**
    **return** $G' = (V, E' \cup X)$
**return** *no*

an even better algorithm. We start by refining the branching by taking advantage of intersections between communities.

**Branching Rule 2.** *Let* $G, \mathcal{C}, \ell$, *and* $E'$ *be the current parameters of a recursive step. If there exist two communities* $C_i = \{u, v, x\}, C_j = \{u, v, y\} \in \mathcal{C}$ *with* $|E(G[C_i])| = 3$, $|E(G[C_j])| = 3$, $E(G[C_i]) \cap E' = \emptyset$ *and* $E(G[C_j]) \cap E' = \emptyset$, *then branch into the following cases:*

1. *Add* $\{u, v\}$ *to* $E'$ *and decrease* $\ell$ *by* 1.

2. *Add* $\{u, x\}$, $\{v, x\}$, $\{u, y\}$ *and* $\{v, y\}$ *to* $E'$ *and decrease* $\ell$ *by* 4.

Branching Rule 2 has a branching vector of $[1, 4]$ which results in a branching number of approximately 1.381. An example where the rule is applicable is shown in Figure 4.1.

**Branching Rule 3.** *Let* $G, \mathcal{C}, \ell$, *and* $E'$ *be the current parameters of a recursive step. If there exist two different communities* $C_i = \{u, v, x\}, C_j = \{u, v, y\} \in \mathcal{C}$ *with* $|E(G[C_i])| = 3$, $|E(G[C_j])| = 3$, $E(G[C_i]) \cap E' = \emptyset$ *and* $E(G[C_j]) \cap E' = \{\{v, y\}\}$, *then branch into the following cases:*

1. *Add* $\{u, v\}$ *to* $E'$ *and decrease* $\ell$ *by* 1.

2. *Add* $\{u, x\}$, $\{v, x\}$, *and* $\{u, y\}$ *to* $E'$ *and decrease* $\ell$ *by* 3.

Branching Rule 3 has a branching vector of $[1, 3]$ which results in a branching number of approximately 1.466. An example where the rule is applicable is shown in Figure 4.2. Before we use these rules to improve the

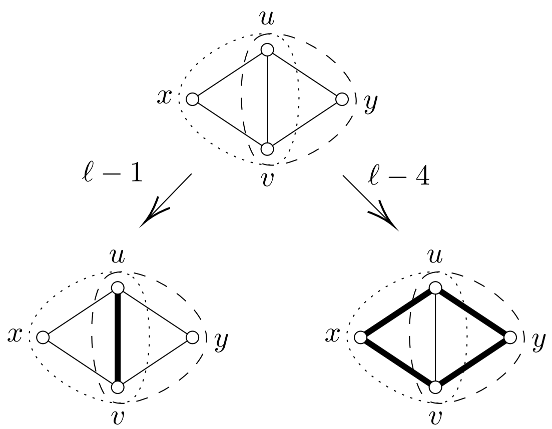


Figure 4.1: An example where Branching Rule 2 is applicable.

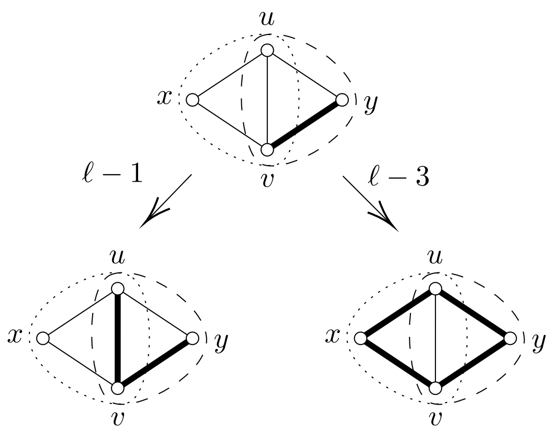


Figure 4.2: An example where Branching Rule 3 is applicable.

search tree size, we reformulate Reduction Rules 1–2 so that they also apply to instances of Stars NWS and Connectivity NWS. Again, we omit the straightforward correctness proofs.

**Reduction Rule 3.** *Let* $G, \mathcal{C}, \ell$, *and* $E'$ *be the current parameters of a recursive step. If there exists a community* $C_i \in \mathcal{C}$ *with* $|E(G[C_i])| = |C_i| - 1$, *then add the edges of* $E(G[C_i])$ *to* $E'$, *remove* $C_i$ *from* $\mathcal{C}$, *and decrease* $\ell$ *by the number of new edges in* $E'$.

**Reduction Rule 4.** *Let* $G, \mathcal{C}, \ell$, *and* $E'$ *be the current parameters of a recursive step. If there exists an independent community* $C_i \in \mathcal{C}$, *then add the minimum number of edges in* $E(G[C_i])$ *to* $E'$ *such that* $|E(G[C_i]) \cap E'| = |C_i| - 1$, *remove* $C_i$ *from* $\mathcal{C}$ *and decrease* $\ell$ *accordingly.*

**Theorem 4.5.** 3-Stars NWS *and* 3-Connectivity NWS *are solvable in* $O(1.466^\ell \cdot |\mathcal{C}| \cdot (n + m))$ *time.*

*Proof.* Again, we adapt the branching of Algorithm 1. The resulting algorithm is shown in Algorithm 3. Frist, we apply Reduction Rules 3–4 exhaustively. In the branching we apply the Branching Rules 2–3 exhaustively.

**Claim 4.6.** *If neither the Reduction Rules 3–4 nor the Branching Rules 2–3 apply anymore, then $|E(G[C_i])| = 3$ and $|E(G[C_i]) \cap E'| = 1$ for each community $C_i \in C$.*

*Proof.* Let $C_i \in \mathcal{C}$ a community and neither Reduction Rules 3 and 4 nor the Branching Rules 2 and 3 are applicable. In case of $|E(G[C_i])| = 2$ Reduction Rule 3 applies. In case of $|E(G[C_i])| < 2$ even the induced subgraph $G[C_i]$ is not connected. Therefore, we observe that $|E(G[C_i])| = 3$. Now, assume that $E(G[C_i]) \cap E' \neq 1$ First, we observe, that $E(G[C_i]) \cap E' \neq \emptyset$ and $C_i$ cannot be independent. Thus, there exists a community $C_j \in \mathcal{C} \setminus \{C_i\}$ with $|C_i \cap C_j| = 2$ and $|E(G[C_j]) \cap E'| < 2$. Then, the following cases are possible:

1. $|E(G[C_j]) \cap E'| = 0$
2. $|E(G[C_j]) \cap E'| = 1$

The first case allows the application of Branching Rule 2 and the second case the application of Branching Rule 3. This contradicts the assumption that Branching Rules 2 and 3 are not applicable. ♢

By Claim 4.6 we know that $|E(G[C_i]) \cap E'| = 1$ for each community $C_i\mathcal{C}$, but we need $|E(G[C_i]) \cap E'| \geq 2$ such that each community is connected. Thus, the remaining task is to select a subset of $E^+ \subseteq E \setminus E'$ such that $E^+ \cap C_i \neq \emptyset$ for each community $C_i \in \mathcal{C}$. Since for each community $C_i \in \mathcal{C}$ the subgraph induced by $C_i$ has exactly two edges not being contained in $E'$, the remaining task can be expressed as an instance of Vertex Cover. We define the equivalent Vertex Cover instance $I_{\text{VC}} = (G_{\text{VC}} = (V_{\text{VC}}, E_{\text{VC}}), \ell)$ where the vertices are defined as $V_{\text{VC}} := E \setminus E'$ and the edge set is defined as $E_{\text{VC}} := \{C_i \setminus E' \mid C_i \in \mathcal{C}\}$. If $G_{\text{VC}}$ has a vertex cover $X$ of size at most $\ell$, then $G' = (V, E' \cup X)$ is a sparsified graph with at most $\ell$ edges. Otherwise, $I$ is a no-instance.

**Running time** The branching numbers of the Branching Rules 2–3 are 1.381 and 1.466. Let $\ell'$ be the number of edges in $E'$ that originated from the Branching Rules 2–3. Hence, the search tree created by these rules has a size of at most $O(1.466^{\ell'})$. The Vertex Cover instance created in Line 12 is solvable in $O(1.286^{\ell-\ell'} \cdot (\ell - \ell') \cdot m)$ time [7]. This leads to an overall running time of $O(1.466^{\ell} \cdot |\mathcal{C}| \cdot (n + m))$ for solving an instance of 3-Stars NWS or 3-Connectivity NWS. □

**Algorithm 3:** Refined Algorithm for 3-Stars NWS and 3-Connectivity NWS

**Input** : $G = (V, E), \mathcal{C}, \ell, E'$
**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no

1 *Apply Reduction Rules 3–4 exhaustively*
2 $\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid |E' \cap E(G[C_i])| < 2\}$
3 **if** $\mathcal{C} = \emptyset$ **then**
4   **return** $G' = (V, E')$
5 **if** $\ell = 0$ **then**
6   **return** *no*
7 **if** *Branching Rule 2 is applicable* **then**
8   *apply Branching Rule 2*
9 **if** *Branching Rule 3 is applicable* **then**
10   *apply Branching Rule 3*
11 **else**
12   $I_{\mathrm{VC}} \leftarrow (E \setminus E', \{C_i \setminus E' \mid C_i \in \mathcal{C}\}, \ell)$
13   **if** $I_{\mathrm{VC}}$ *is a yes-instance with a vertex cover $X$ of size at most $\ell$* **then**
14     **return** $G' = (V, E' \cup X)$
15 **return** *no*

Now we take a closer look at instances where the community size is bounded by an arbitrary integer $d$. This implies that the number of edges in a subgraph induced by a community does not exceed $d^2$. This allows us to adapt Algorithm 1 leading to fixed-parameter tractability for $d$-Density NWS.

**Theorem 4.7.** *$d$-Density NWS is solvable in $O(d^{2\ell} \cdot |\mathcal{C}| \cdot (n + m))$.*

*Proof.* We use a branching strategy trying out all possible subsets of edges that would satisfy the density requirement in the sparsified graph. The algorithm is shown in Algorithm 4. In each recursive step, we select a community $C_i \in \mathcal{C}$ such that $E(G[C_i]) \cap E'| < \lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil|$. If no such community is left and $|E'| \leq \ell$, then the instance is a yes-instance of $d$-Density NWS. Let $x_i = \lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil| - E(G[C_i]) \cap E'|$ be the minimum number of edges which needs to be added to $E'$ such that the density requirement of $C_i$ is satisfied by $E'$. Then, we make a recursive call for each size-$x_i$ subset of $E(G[C_i]) \setminus E'$ and decrease $\ell$ by $x_i$.

**Running Time** Let $C_i \in \mathcal{C}$ be the community selected for branching in Line 6. The branching vector of the branching in the loop in Lines 8-10 has $\binom{|E(G[C_i]) \setminus E'|}{x}$ entries of value $x_i$. The branching number of this vector is bounded by $\sqrt[x]{\binom{\binom{d}{2}}{x}} \leq \sqrt[x]{\binom{d^2}{x}} \leq \sqrt[x]{d^{2x}} \leq d^2$. This leads to a

**Algorithm 4:** Algorithm for $d$-Density NWS: *SolveDNWS*

**Input** : $G = (V, E), \mathcal{C}, \alpha, \ell, E'$

**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no

1 $\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid \frac{1}{3} \cdot |E' \cap E(G[C_i])| < \alpha(C_i)\}$
2 **if** $\mathcal{C} = \emptyset$ **then**
3 | **return** $G' = (V, E')$
4 **if** $\ell = 0$ **then**
5 | **return** *no*
6 $C_i \leftarrow$ pick element from $\mathcal{C}$
7 $x_i \leftarrow \lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil | - E(G[C_i]) \cap E'|$
8 **forall** $E'_i \subseteq (E(G[C_i]) \setminus E')$ *with* $|E_i| = x_i$ **do**
9 | **if** *SolveDNWS*$(G, \mathcal{C}, \alpha, \ell - x_i, E' \cup E'_i)$ *returns a graph* $G'$ **then**
10 | | **return** $G'$
11 **return** *no*

search tree size of at most $O(d^{2\ell})$. This results in an overall running time of $O(d^{2\ell} \cdot |\mathcal{C}| \cdot (n + m))$. □

Next, we state the fixed-parameter tractability of $d$-Stars NWS by giving an algorithm running in $O(d^{\ell} \cdot |\mathcal{C}| \cdot (n + m))$ time. The algorithm exploits that a community size of at most $d$ restricts the number of different possible spanning stars in each subgraph induced by a community to $d$.

**Theorem 4.8.** $d$-Stars NWS *is solvable in* $O(d^{\ell} \cdot |\mathcal{C}| \cdot (n + m))$ *time.*

*Proof.* We use a branching strategy trying out all potential centers of each community in the sparsified graph. The algorithm is shown in Algorithm 5. In each recursive step, we select a community $C_i \in \mathcal{C}$ for which the requirement of containing a spanning star is not satisfied by the current edge set $E'$. If no such community is left and $|E'| \leq \ell$, then the instance is a yes-instance of $d$-Stars NWS. If there exists such a community and $\ell$ is already 0, then $I$ is a no-instance of $d$-Stars NWS. If none of the conditions mentioned above applies, we branch on each potential center vertex $c_i$ of community $C_i$, add the necessary edges to $E'$ and decrease $\ell$ accordingly.

**Running time** Let $C_i \in \mathcal{C}$ be the community selected for branching in Line 6. The branching vector of the branching in the loop in Lines 7–10 has at most $d$ entries of value at least 1 because a community has at most $d$ and $\ell$ is decreased each time at least by 1. The branching number of this branching vector is $d$. This leads to a search tree size of at most $O(d^{\ell})$. This results in a overall running time of $O(d^{\ell} \cdot |\mathcal{C}| \cdot (n + m))$. □

Finally, we give an algorithm solving $d$-Connectivity NWS which has

**Algorithm 5:** Algorithm for $d$-Stars NWS: *SolveSNWS*

**Input** : $G = (V, E), \mathcal{C}, \ell, E'$
**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no
1 $\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid G^*[C_i]$ does not contain a star for $G^* = (V, E')\}$
2 **if** $\mathcal{C} = \emptyset$ **then**
3 | **return** $G' = (V, E')$
4 **if** $\ell = 0$ **then**
5 | **return** *no*
6 $C_i \leftarrow$ pick element from $\mathcal{C}$
7 **forall** $c_i \in C_i$ *with* $C_i \subseteq N[c_i]$ **do**
8 | $E'_i \leftarrow \{\{c_i, v \mid v \in C \setminus \{c_i\}\} \setminus E'|$
9 | **if** *SolveSNWS*$(G, \mathcal{C}, \ell - |E'_i|, E' \cup E'_i)$ *returns a graph* $G'$ **then**
10 | | **return** $G'$
11 **return** *no*

a running time of $O(d^{2\ell} \cdot |\mathcal{C}| \cdot (n+m))$. Thus, $d$-Connectivity NWS is also fixed-parameter tractable with respect to parameter $\ell$.

**Theorem 4.9.** *$d$-Connectivity NWS is solvable in $O(d^{2\ell} \cdot |\mathcal{C}| \cdot (n+m))$ time.*

*Proof.* The algorithm is shown in Algorithm 5. In each recursive step, we select a community $C_i \in \mathcal{C}$ for which the induced subgraph $G^*[C_i]$ is not already connected in $G^* = (V, E')$. If no such community is left and $|E'| \leq \ell$, then the instance is a yes-instance of $d$-Connectivity NWS. If there exists such a community and $\ell = 0$, then $I$ is a no-instance of $d$-Connectivity NWS. If none of the conditions mentioned above applies, we branch on the edges in the subgraph induced by $C_i$ that are not already contained in $E'$ and decrease $\ell$ by 1.

**Running Time** Let $C_i \in \mathcal{C}$ be the community selected for branching in Line 6. The branching vector of the branching in the loop in Lines 7-9 has at most $|E(G[C_i])|$ entries of value 1. Thus, the branching number of this vector is bounded by $|E(G[C_i])| \leq \binom{d}{2} \leq d^2$. This leads to a search tree size of at most $O(d^{2\ell})$. This results in an overall running time of $O(d^{2\ell} \cdot |\mathcal{C}| \cdot (n+m))$. □

In the previous part, we saw FPT-algorithms for $d$-Density NWS, $d$-Stars NWS, and $d$-Connectivity NWS parameterized by $\ell$. Therefore, the next question that arises is whether polynomial kernels exists. Let $\mathcal{F}$ be a family of sets of equal size over a universe $U$. A set $\mathcal{S} \subseteq \mathcal{F}$ is called a sunflower with $|\mathcal{S}|$ petals and a core $Y$, if $S_i \cap S_j = Y$ for each different $S_i, S_j \in \mathcal{S}$. First, we state the Sunflower Lemma by Erdős and Rado [13] helping us to develop polynomial kernels for all three problems.

**Algorithm 6:** Algorithm for $d$-Connectivity NWS: *SolveCNWS*

**Input** : $G = (V, E), \mathcal{C}, \ell, E'$
**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no
$\mathcal{C} \leftarrow \{C_i \in \mathcal{C} \mid G^*[C_i] \text{ is not connected for } G^* = (V, E')\}$
**if** $\mathcal{C} = \emptyset$ **then**
  **return** $G' = (V, E')$
**if** $\ell = 0$ **then**
  **return** *no*
$C_i \leftarrow$ pick element from $\mathcal{C}$
**forall** $e \in E(G[C_i]) \setminus E'$ **do**
  **if** *SolveCNWS*$(G, \mathcal{C}, \ell - 1, E' \cup \{e\})$ *returns a graph* $G'$ **then**
    **return** $G'$
**return** *no*

**Lemma 4.10** (**Sunflower Lemma [13][9]**). *Let $U$ be a set of elements and $\mathcal{F}$ a family of subsets of $U$ each having size exactly $d$. If $|\mathcal{A}| > d!(k-1)^d$, then $\mathcal{F}$ contains a sunflower with $k$ petals which can be computed in time polynomial in $|\mathcal{F}| + |U| + d$.*

Next, we formulate a reduction rule for Density NWS based on the existence of a sunflower in the set of communities of an instance.

**Reduction Rule 5.** *Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of $d$-*Density *NWS such that $\mathcal{C}$ contains a sunflower $\mathcal{S}$ with $\ell + 1$ petals. Let $Y$ denote the core of the sunflower. Let $x$ be the smallest number such that $x \geq \lceil \alpha(C_i) \cdot \binom{d}{2} \rceil$ and $x \geq \lceil \alpha(C_j) \cdot \binom{d}{2} \rceil$ for two different communities $C_i, C_j \in \mathcal{S}$. If $|E(G[Y])| < x$, then $I$ is a no-instance. Otherwise, add a new community $C_y = Y \cup Z$ where $Z$ contains $d - |Y|$ new vertices to $\mathcal{C}$ and set $\alpha(C_y) = x / \binom{d}{2}$. Remove all communities $C_i \in \mathcal{S}$ with $\lceil \alpha(C_j) \cdot \binom{d}{2} \rceil \leq x$ from $\mathcal{C}$ and add the vertices in $Z$ to $V$. This are at least two communities because $x$ has been picked such that $x$ edges are sufficient to satisfy the density requirement of at least two communities. Since only one new community is added, the described rule reduces the number of communities by at least* 1.

**Lemma 4.11.** *Rule 5 is correct.*

*Proof.* Let $I = (G = (V, E), \mathcal{C}, \alpha, \ell)$ be an instance of $d$-Density NWS such that $\mathcal{C}$ contains a sunflower $\mathcal{S}$ with $\ell + 1$ petals. Let $Y$ denote the core of the sunflower. Without loss of generality we assume $\alpha(C_i) > 0$ for each community $C_i \in \mathcal{C}$. Let $x$ be the smallest number such that $x \geq \lceil \alpha(C_i) \cdot \binom{d}{2} \rceil$ and $x \geq \lceil \alpha(C_j) \cdot \binom{d}{2} \rceil$ for $C_i, C_j \in \mathcal{S}$ with $C_i \neq C_j$. We begin with the case when $|E(G[Y])| < x$. If $|E(G[Y])| = 0$, then there are at least $\ell + 1$ edges necessary to satisfy the density requirement $\alpha(C_i)$ of each community $C_i \in \mathcal{S}$. This implies that $I$ is a no-instance of $d$-Density NWS. If $|E(G[Y])| > 0$, then there are at least $\ell$ communities whose density requirement cannot be satisfied by edges in $E(G[Y])$ solely. Recall that $x$ is the smallest number of edges necessary to satisfy the density requirement of at least two different communities. Hence, there are at least $\ell$ communities which need at least $x$ edges to satisfy their density requirement. Since $|E(G[Y])| < x$, it is not sufficient to only select edges of $E(G[Y])$ to satisfy the density requirement of these $\ell$ communities. Therefore, at least an additional edge has to be contained in the sparsified for each of these $\ell$ communities. Note that this additional edge is different for each of the $\ell$ communities because the petals of a sunflower are pairwise disjoint. To satisfy the density requirement of the remaining community, at least one edge in $E(G[Y])$ is needed. This are at least $\ell + 1$ edges which again implies that $I$ is a no-instance. We now consider the case when $|E(G[Y])| \geq x$. First, we observe that in case of $I$ being a yes-instance $d$-Density NWS a sparsified graph $G'$ has to contain $x$ edges of $E(G[Y])$ because otherwise the same argumentation as in the case $|E(G[Y])| < x$ applies. This is ensured by adding the community $C_y$ and setting its density requirement to $\alpha(C_y) = x/\binom{d}{2}$. Observe that satisfying the density requirement $\alpha(C_y)$ implies satisfying the density requirement of all petals $P_i \in \mathcal{S}$ with $x \geq \lceil \alpha(P_i) \cdot \binom{d}{2} \rceil$. Therefore, the removal of these petals from $\mathcal{C}$ is correct. □

With Reduction Rule 5 and the Sunflower Lemma we are able to obtain a kernel for $d$-Density NWS.

**Theorem 4.12.** $d$-Density NWS *has a kernel consisting of at most* $d! \cdot \ell^d \cdot d$ *communities and at most* $d! \cdot \ell^d \cdot d^2$ *vertices.*

*Proof.* Let $I = (G, \mathcal{C}, \alpha, \ell)$ be an instance of $d$-Density NWS with $|C| > d! \cdot \ell^d \cdot d$. The assumption $|C| > d! \cdot \ell^d \cdot d$ implies the existence of a $d' \leq d$ such that the number of communities of size exactly $d'$ is greater than $d! \cdot \ell^d$. This allows us to apply the Sunflower Lemma to obtain a sunflower with $\ell + 1$ petals. Therefore Reduction Rule 5 is applicable. The application of Reduction Rule 5 is done exhaustively. This is possible in polynomial time because according to the Sunflower Lemma finding such a sunflower takes only polynomial time. The rule can be applied at most $|\mathcal{C}|$ times because each application reduces the number of communities at least by 1. Let $I' = (G', \mathcal{C}', \alpha, \ell)$ denote the reduced instance. We observe that $|\mathcal{C}'| \leq d! \cdot \ell^d \cdot d$ and therefore $|V(G')| \leq d! \cdot \ell^d \cdot d^2$. □

The existence of a sunflower in the set of communities leads also to reduction rules for Stars NWS and Connectivity NWS. In the following, these reduction rules are presented.

**Reduction Rule 6.** *Let* $I = (G = (V, E), \mathcal{C}, \ell)$ *be an instance of* $d$-Stars NWS. *If* $\mathcal{C}$ *contains a sunflower with* $\ell + 1$ *petals, then* $I$ *is a no-instance of* $d$-Stars NWS.

**Lemma 4.13.** *Rule 6 is correct.*

*Proof.* Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of $d$-Stars NWS such that $\mathcal{C}$ contains a sunflower $\mathcal{S}$ with $\ell + 1$ petals. Let $Y$ denote the core of the sunflower. For each community $C_i \in \mathcal{S}$, there exists at least one vertex $u_i \in C_i$ with $u_i \notin Y$ and $u_i \notin C_j$ for any other petal $C_j \in \mathcal{S} \setminus \{C_i\}$. Let $G' = (V, E')$ be an arbitrary sparsified graph of $G$. Let $c_{G'} : \mathcal{C} \to V$ denote the mapping of communities to their center vertex in $G'$. We distinguish two cases $c_{G'}(C_i) = u_i$ and $c_{G'}(C_i) \neq u_i$ for each community $C_i \in \mathcal{S}$. If $c_{G'}(C_i) = u_i$, then there exists another vertex $v_i \in C_i$ with $v \neq u$. This implies $\{u_i, v_i\} \in E'$. If $c_{G'}(C_i) \neq u_i$, then we have $\{u_i, c_{G'}(C_i)\} \in E'$. This implies $|E'| \geq \ell + 1$. Thus, $I$ is a no-instance of $d$-Stars NWS. □

**Reduction Rule 7.** *Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of $d$-*Connectivity *NWS. If $\mathcal{C}$ contains a sunflower with $\ell + 1$ petals, then $I$ is a no-instance of $d$-*Connectivity *NWS.*

**Lemma 4.14.** *Rule 7 is correct.*

*Proof.* Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of $d$-Connectivity NWS such that $\mathcal{C}$ contains a sunflower $\mathcal{S}$ with $\ell + 1$ petals. Let $Y$ denote the core of the sunflower. For each community $C_i \in \mathcal{S}$ there exists at least one vertex $u_i \in C_i$ with $u_i \notin Y$ and $u_i \notin C_j$ for any other community $C_j \in \mathcal{X} \setminus \{C_i\}$. The vertex $u_i$ has to be connected to the core of the sunflower. This requires at least $|X| = \ell + 1$ edges, implying that $I$ is a no-instance of $d$-Connectivity NWS. □

Together with the Sunflower Lemma, both rules state an upper bound for the number of communities in yes-instances of Stars NWS and Connectivity NWS. This leads to the following problem kernels.

**Theorem 4.15.** *$d$-*Stars *NWS and $d$-*Connectivity *NWS have a kernel consisting of at most $d! \cdot \ell^d \cdot d$ communities and at most $d! \cdot \ell^d \cdot d^2$ vertices.*

*Proof.* Let $I = (G, \mathcal{C}, \ell)$ be an instance of $d$-Stars NWS or an instance of $d$-Connectivity NWS with $|C| > d! \cdot \ell^d$. The assumption $|C| > d! \cdot \ell^d \cdot d$ implies the existence of a $d' \leq d$ such that the number of communities of size exactly $d'$ is greater than $d! \cdot \ell^d$. This allows us to apply the Sunflower Lemma to obtain a sunflower with $\ell + 1$ petals and a core $Y$. In case of $d$-Stars NWS, Reduction Rule 6 is applicable stating that $I$ is a no-instance of $d$-Stars NWS. In case of $d$-Connectivity NWS, Reduction Rule 6 is applicable also stating that $I$ is a no-instance $d$-Connectivity NWS. This implies that a yes-instance of $d$-Stars NWS or $d$-Connectivity NWS consists of at most $d! \cdot \ell^d \cdot d$ communities and at most $d! \cdot \ell^d \cdot d^2$ vertices. □

The next step in the analysis of parameter $\ell$ is to investigate the fixed-parameter tractability of instance where the community size is unbounded.

Unfortunately, Density NWS is $W[2]$-hard with respect to parameter $\ell$ and therefore not fixed-parameter tractable assuming FPT $\neq W[2]$. The $W[2]$-hardness is directly implied by the reduction from Hitting Set by Gionis et al. [16] which was used there to prove NP-hardness. For the sake of completeness, we recall their construction in the proof of Theorem 4.16 to reduce from Hitting Set parameterized by the size of the hitting set which is known to be $W[2]$-complete [9].

**Theorem 4.16.** Density NWS *is W[2]-hard with respect to parameter* $\ell$*.*

*Proof.* We give a simplified version of the reduction from Hitting Set given by Gionis et al. [16]. Let $I_{\text{HS}} = (U, \mathcal{S}, k)$ be an instance of Hitting Set where $U$ is the universum, $\mathcal{S}$ a collection of subsets over $U$ and $k$ the maximum size of the hitting set. We start by defining the graph $G = (V, E)$. The set $V$ contains each vertex of $V$ and one additional vertex $z$. The edge set is $E \coloneqq \{\{z, u\} \mid u \in U\}$. Thus, $G$ is a star with center $z$. We define the set of communities $\mathcal{C} \coloneqq \{S_i \cup \{z\} \mid S_i \in S\}$. For each community $C_i \in \mathcal{C}$, we set the density requirement $\alpha(C_i) \coloneqq 1/\binom{|U|}{2}$. Finally, we set the parameter $\ell \coloneqq k$. Let $I_{\text{DNS}} = (G, \mathcal{C}, \alpha, \ell)$ denote the resulting instance of Density NWS.

**Correctness** We show that $I_{\text{HS}}$ is a yes-instance of Hitting Set if and only if $I_{\text{DNS}}$ is a yes-instance of Density NWS.

($\Rightarrow$) Let $X$ be a hitting set of size at most $k$. We show how to obtain a sparsified graph $G' = (V, E')$ with $|E'| \leq \ell = k$. We set $E' \coloneqq \{\{z, x\} \mid x \in X\}$ observing that $|E'| \leq k = \ell$. Since $X$ is a hitting set, there exists for each community $C_i = \{S_i\} \cup \{z\}$ an element $x_i \in X$ with $x_i \in S_i$ implying $\{x_i, z\} \in E'$. Therefore, for each community $C_i \in \mathcal{C}$, the density requirement of $1/\binom{|V|-1}{2}$ is satisfied. Therefore, $G'$ is a sparsified graph which implies that $I_{\text{DNS}}$ is a yes-instance of Density NWS.

($\Leftarrow$) Let $I_{\text{DNS}}$ be a yes-instance of Density NWS and let $G' = (V, E')$ be a sparsified graph. We show how to obtain a hitting set $X$ with $|X| \leq \ell = k$. We set $X \coloneqq \{x \mid \{x, z\} \in E'\}$ observing that $|X| \leq \ell = k$. For each $S_i \in \mathcal{S}$, there exists a community $C_i$ such that $E(G'[C_i]) \cap E' \neq \emptyset$. This is the case because for every community $C_i \in \mathcal{C}$, there exists an edge $\{x_i, z\} \in E(G'[C_i])$ due to the density requirement of $\alpha(C_i) > 0$. This implies that there exists for each $S_i \in S$ an element $x_i \in X$ with $x_i \in S_i$. Therefore, $X$ is a hitting set with $|X| \leq k$ and $I_{\text{HS}}$ is a yes-instance of Hitting Set. $\square$

Next, we transfer the fixed-parameter tractability of $d$-Stars NWS and $d$-Connectivity NWS to instances with communities of unbounded size by making an observation about the relation between parameter $\ell$ and the maximum size $d$ of a community in yes-instances of Stars NWS and Connectivity NWS.

**Lemma 4.17.** *Let $I = (G, \mathcal{C}, \ell)$ be an instance of* Stars NWS *or* Connectivity NWS*. If $\max_{C_i \in \mathcal{C}} |C_i| > \ell + 1$, then $I$ is a no-instance.*

*Proof.* Let $C_i \in \mathcal{C}$ be a community with $|C_i| > \ell + 1$. Then, $C_i$ contains at least $\ell + 2$ vertices. A connected graph with $\ell + 2$ vertices has at least $\ell + 1$ edges. The same holds for a graph which is a star of size $\ell + 1$. Therefore, a sparsified graph contains at least $\ell+1$ edges, implying that $I$ is a no-instance Stars NWS and Connectivity NWS. □

Lemma 4.17 limits the maximum size $d$ of a community in yes-instances of Stars NWS or Connectivity NWS to $\ell + 1$. Hence, we can express the running times in Theorem 4.8 and Theorem 4.9 depending on parameter $\ell$ solely. This leads us to the two following corollaries.

**Corollary 4.18.** Stars NWS *is solvable in $O((\ell+1)^{\ell} \cdot |\mathcal{C}| \cdot (n+m))$ time.*

*Proof.* Let $I$ be an instance of Stars NWS. If Lemma 4.17 applies, then $I$ is a no-instance. Otherwise we know that $d \leq \ell + 1$ allowing us to use Algorithm 5 presented in the proof of Theorem 4.8. □

**Corollary 4.19.** Connectivity NWS *is solvable in $O((\ell+1)^{2\ell} \cdot |\mathcal{C}| \cdot (n+m))$ time.*

*Proof.* Let $I$ be an instance of Connectivity NWS. If Lemma 4.17 applies, then $I$ is a no-instance. Otherwise we know that $d \leq \ell + 1$ allowing us to use Algorithm 6 presented in the proof of Theorem 4.9. □

After knowing that Stars NWS and Connectivity NWS parameterized by $\ell$ are fixed-parameter tractable, the next question that arises is the existence of polynomial kernels. Under the assumption NP $\not\subseteq$ coNP/poly this question can be answered negatively, as shown in the following theorem.

**Theorem 4.20.** Stars NWS *and* Connectivity NWS *do not admit a polynomial kernel with respect to parameter $\ell$ unless NP $\subseteq$ coNP/poly.*

*Proof.* We give a polynomial parameter transformation from Hitting Set parameterized by the size of the universe and the solution size which is known to not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [9]. Let $I_{\text{HS}} = (U, \mathcal{S}, k)$ be an instance of Hitting Set. The following construction works for both Stars NWS and Connectivity NWS. We start by defining the graph $G = (V, E)$. The set $V$ contains each element of $U$ and one additional vertex $z$. $G$ is a clique. We define the set of communities $\mathcal{C} := \{S_i \cup \{z\} \mid S_i \in S\} \cup \{\{u, v\} \mid u, v \in U, u \neq v\}$. Finally, we set the parameter $\ell := \binom{|U|}{2} + k$. Let $I = (G, \mathcal{C}, \ell)$ denote the resulting instance of Stars NWS or Connectivity NWS.

**Correctness** We show that $I_{\text{HS}}$ is a yes-instance of Hitting Set if and only if $I$ is a yes-instance of Stars NWS and Connectivity NWS.

($\Rightarrow$) Let $X$ be a hitting set of size at most $k$. We show how to obtain a sparsified graph $G' = (V, E')$ with $|E'| \leq \binom{|U|}{2} + \ell$. We set $E' = \{\{z, x\} \mid x \in X\} \cup \{\{u, v\} \mid u, v \in U, u \neq v\}$ observing that $E' \leq \binom{|U|}{2} + k = \ell$. Since $X$ is a hitting set, there exists for each community $C_i = \{S_i\} \cup \{z\}$ an element $x_i \in X$ with $x_i \in S_i$ implying $\{x_i, z\} \in E'$. Therefore, for each community $C_i \in \mathcal{C}$, the element $x_i$ is a center of a spanning star in $G'[C_i]$ because $x_i$ is connected to $z$ and to all other vertices in $U$. Hence, for each community $C_i \in \mathcal{C}$ the induced subgraph $G'[C_i]$ contains a spanning star and is therefore also connected. This implies that $I$ is a yes-instance of Stars NWS and Connectivity NWS.

($\Leftarrow$) Let $I$ be a yes-instance of Stars NWS or Connectivity NWS and let $G' = (V, E')$ be a sparsified graph. We show how to obtain a hitting set $X$ with $|X| \leq \ell - \binom{|U|}{2}$. First, we observe, that $\{\{u, v\} \mid u, v \in U, u \neq v\} \subseteq E'$ due to the size-two communities in $\mathcal{C}$. We set $X \coloneqq \{u \mid \{u, z\} \in E'\}$ observing that $|X| \leq \ell - \binom{|U|}{2} = k$. For each $S_i \in \mathcal{S}$, there exists a community $C_i$ and a vertex $u_i \in U$ such that $C_i \setminus \{z\} = S_i$ and $\{u_i, z\} \in E'$. For Stars NWS this is the case because for every community $C_i \in \mathcal{C}$ the induced subgraph $G'[C_i]$ contains a spanning star. For Connectivity NWS this is the case because for every community $C_i \in \mathcal{C}$ the induced subgraph $G'[C_i]$ is connected. This implies that there exists for each $S_i \in S$ an element $u_i \in X$ with $u_i \in S_i$. Therefore, $X$ is a hitting set with $|X| \leq \ell$ and $I_{\text{HS}}$ is a yes-instance of Hitting Setz. ☐

This completes our analysis of the parameterized complexity of Density NWS, Stars NWS and Connectivity NWS parameterized by $\ell$.

## 4.2 The Parameter $k$

In the previous section we considered the natural parameter $\ell$ defined as the number of edges kept in a sparsified graph. We now analyze the dual parameter $k \coloneqq m - \ell$. Informally, $k$ is the number of edges that are removed from the initial graph. Recall that we reduced in Theorem 3.1 from Vertex Cover to show NP-hardness. Since an undirected graph $G = (V, E)$ contains a vertex cover of size at most $k$ if and only if $G$ contains an independent set of size at least $|V| - k$, we are able to reuse the construction to reduce from Independent Set which is known to be $W[1]$-complete [9]. This reduction preserves the parameter $|V| - k$ and therefore allowing us to obtain $W[1]$-hardness for Density NWS, Stars NWS, and Connectivity NWS.

**Theorem 4.21.** Density NWS, Stars NWS*, and* Connectivity NWS *are* $W[1]$*-hard for parameter* $k$ *even if restricted to communities of size at* *most* 3.

*Proof.* We give a parameterized reduction from Independent Set. Let $I_{\text{IS}} = (G = (V, E), k)$ be an instance of Independent Set.

**Density NWS** We start with the proof for 3-Density NWS. We use the same construction as in the reduction from Vertex Cover in Theorem 3.1. Let $G_{\text{3DNS}} = (V_{\text{3DNS}}, E_{\text{3DNS}})$ be the graph, let $\mathcal{C}$ be the set of communities and let $\alpha$ be the density requirement resulting from the construction in Theorem 3.1. Recall that the vertex set is defined as $V_{\text{3DNS}} \coloneqq V \cup \{z\}$. The edge set is defined as $E_{\text{3DNS}} \coloneqq \{\{v, z\} \mid v \in V\}$. The set of communities is defined as $\mathcal{C} \coloneqq \{\{u, v, z\} \mid \{u, v\} \in E\}$ and the density requirement is defined as $\alpha : \mathcal{C} \to [0, 1],\ C_i \mapsto \frac{1}{3}$. Hence, we have $|E_{\text{3DNS}}| = |V|$ and $|\mathcal{C}| = |E|$. We set $\ell \coloneqq |E_{\text{3DNS}}| - k = |V| - k$. Let $I_{\text{3DNS}} = (G_{\text{3DNS}}, \mathcal{C}, \ell, \alpha)$ denote the resulting instance of 3-Density NWS.

**Correctness** We show that $I_{\text{IS}}$ is a yes-instance of Independent Set if and only if $I_{\text{3DNS}}$ is a yes-instance of 3-Density NWS.

($\Rightarrow$) Let $I_{\text{IS}}$ be a yes-instance of Independent Set and $S$ the independent set of size at least $k$. Then, $G$ contains a vertex cover $X$ of size at most $|V| - k$ implying that $I_{\text{3DNS}}$ is a yes-instance of 3-Density NWS.

($\Leftarrow$) Recall that if $I_{\text{3DNS}}$ is a yes-instance of 3-Density NWS, $G$ contains a vertex cover $X$ of size at most $\ell = |V| - k$. This implies that $G$ contains an independent set $S$ of size at least $|V| - \ell = k$. Hence, $I_{\text{IS}}$ is a yes-instance of Independent Set.

**Stars NWS and Connectivity NWS** According to Lemma 3.4, 3-Stars NWS and 3-Connectivity NWS are essentially the same. Therefore, we only give the proof for 3-Connectivity NWS.

Let $G_{\text{3CNS}} = (V_{\text{3CNS}}, E_{\text{3CNS}})$ be the graph and let $\mathcal{C}$ be the set of communities resulting from the construction in Theorem 3.5. Recall that the vertex set is defined as $V_{\text{3CNS}} \coloneqq V \cup \{z\}$. The edge set is defined as $E_{\text{3CNS}} \coloneqq \{\{v, z\} \mid v \in V\} \cup E$. The set of communities is defined as $\mathcal{C} \coloneqq \{\{u, v, z\}, \{u, v\} \mid \{u, v\} \in E\}$ Next, recall that $|E_{\text{3CNS}}| = |V| + |E|$ and $|\mathcal{C}| = 2 \cdot |E|$. We set $\ell \coloneqq |E_{\text{3CNS}}| - k = |V| - k + |E|$. Let $I_{\text{3CNS}} = (G_{\text{3CNS}}, \mathcal{C}, \ell)$ denote the resulting instance of 3-Connectivity NWS.

**Correctness** We show that $I_{\text{IS}}$ is a yes-instance of Independent Set if and only if $I_{\text{3CNS}}$ is a yes-instance of Connectivity NWS.

($\Rightarrow$) Let $I_{\text{IS}}$ be a yes-instance of Independent Set and let $S$ the independent set of size at least $k$. Then, $G$ contains a vertex cover $X$ of size at most $|V| - k$ implying that $I_{\text{3CNS}}$ is a yes-instance of Connectivity NWS.

($\Leftarrow$) Recall that if $I_{\text{3CNS}}$ is a yes-instance Connectivity NWS, $G$ contains a vertex cover $X$ of size at most $\ell - |E| = |V| - k$. This implies that

$G$ contains an independent set $S$ of size at least $|V| - (\ell - |E|) = k$. Hence, $I_{\text{IS}}$ is a yes-instance of Independent Set. □

To complete the study of parameter $k$, it is left to mention that all three problems are in XP for $k$. This can be seen by a brute force algorithm performing an exhaustive search over all subsets $E^*$ of the edges in the input graph $G = (V, E)$ with a size of $k$. If for any such set $E^*$ the resulting graph $G' = (V, E \setminus E^*)$ satisfies the respective graph property for each community, then $I$ is a yes-instance. The number of such edge sets is bounded by $m^k$ leading to a overall running time of $O(m^k \cdot |\mathcal{C}| \cdot \text{poly}(n + m))$.

## 4.3 The Parameter $t$

Even though parameter $\ell$, the number of edges in the sparsified graph, depends only on the size of the sparsified graph, it is not completely independent from the size of the corresponding Π-NWS instance. Since in sparsified graphs of yes-instances of Stars NWS and Connectivity NWS each subgraph induced by a community is connected, a graph $G$ with more vertices leads to a greater parameter $\ell$. This is the case because each subgraph induced by a community $C_i$ has at least $|C_i| - 1$ edges. Hence, a sparsified graph $G'$ has at least $n - 1$ edges assuming $G'$ is connected and each vertex is contained in at least one community. In other words, $n - 1$ is a lower bound for parameter $\ell$ in this case. Next, we study Stars NWS and Connectivity NWS parameterized above this lower bound. The parameter $t$ is defined as the size of a minimum feedback edge set of the sparsified graph of an instance of Stars NWS and Connectivity NWS. Thus, the parameter $t$ measures how close the sparsified graph is to a tree. Formally, the definition is $t := \ell - n + x$ where $x$ denotes the number of connected components of $G'$.

For Stars NWS in the case of $t = 0$, we give a polynomial-time algorithm. This answers an open question of Korach and Stern [21]. It is open whether Stars NWS parameterized by $t$ is fixed-parameter tractable. Moreover, it is open whether Stars NWS parameterized by $t$ is in XP.

**Theorem 4.22.** *Let $I = (G = (V, E), \mathcal{C}, \ell)$ where $\ell := n - 1$ be an instance of* Stars NWS *where the hypergraph $\mathcal{H} = (V, \mathcal{C})$ is connected. Such an instance $I$ is solvable in $O(|\mathcal{C}|^2 \cdot n^2)$ time.*

*Proof.* Before we present our algorithm, we start by making several observations about a yes-instance of Stars NWS for $t = 0$. Next, we observe that a sparsified graph of a yes-instance of Stars NWS for $t = 0$ is acyclic and therefore a tree.

**Claim 4.23.** *If $I$ is a yes-instance of* Stars NWS, *then every sparsified graph $G' = (V, E')$ is a tree.*

*Proof.* The star requirement implies that for each community $C_i \in \mathcal{C}$ the induced subgraph $G'[C_i]$ is connected. Since $\mathcal{H}$ is connected, $G'$ is also connected. The graph $G'$ is a tree because $|E'| = n - 1$ and therefore acyclic. $\diamondsuit$

Next, we make two observations about the relation of the center vertices of two communities whose intersection is of size at least 2 or 3.

**Claim 4.24.** *Let $I$ be a yes-instance of* Stars NWS *for $t = 0$ with a sparsified graph $G'$. Let $c_{G'} : \mathcal{C} \to V$ denotes the mapping of communities to their center vertex in $G'$ and let $C_i, C_j \in \mathcal{C}$ be two communities.*

1. *If $|C_i \cap C_j| \geq 2$, then $c_{G'}(C_i) \in C_i \cap C_j$ and $c_{G'}(C_j) \in C_i \cap C_j$.*

2. *If $|C_i \cap C_j| \geq 3$, then $c_{G'}(C_i) = c_{G'}(C_j)$ and $c_{G'}(C_i) \in C_i \cap C_j$ and $c_{G'}(C_j) \in C_i \cap C_j$.*

*Proof.* Let $S_i$ denote the spanning star contained in $G'[C_i]$ and let $S_j$ denote the spanning star contained in $G'[C_j]$. We start with the proof of the first part. Without loss of generality we assume $c_{G'}(C_i) \notin C_j$ towards a contradiction. First, we observe that $S_i$ and $S_j$ do not have an edge in common because each edge in $S_i$ has at most one endpoint in $C_i \cap C_j$. This implies that $|E(G'[C_i \cup C_j])| \geq |C_i| + |C_j| - 2$. By Claim 4.23, we know $|E(G'[C_i \cup C_j])| = |C_i \cup C_j| - 1$ which leads to a contradiction: $|E(G'[C_i \cup C_j])| > |E(G'[C_i \cup C_j])| - 1 \geq |C_i| + |C_j| - 2 - 1 \geq |C_i| + |C_j| - |C_i \cap C_j| - 1 = |C_i \cup C_j| - 1 = |E(G'[C_i \cup C_j])|$.

Now, we prove the second part. We assume $c_{G'}(C_i) \neq c_{G'}(C_j)$ towards a contradiction. Due to the first part, we have $c_{G'}(C_i) \in C_i \cap C_j$ and $c_{G'}(C_j) \in C_i \cap C_j$ because $|C_i \cap C_j| \geq 3$. Observe that $S_i$ and $S_j$ have exactly the edge $\{c_{G'}(C_i), c_{G'}(C_j)\}$ in common. Since $|C_i \cap C_j| \geq 3$, there exists $u \in C_i \cap C_j$ with $u \neq c_{G'}(C_i)$ and $u \neq c_{G'}(C_j)$. Because $S_i$ and $S_j$ are stars, the edges $\{u, c_{G'}(C_i)\}$ and $\{u, c_{G'}(C_j)\}$ are contained in $G'$. Hence, there is a triangle in $G'[C_i \cap C_j]$ which is a contradiction to Claim 4.23 that $G'$ is a tree. $\diamondsuit$

**Algorithm** We begin by giving an intuition how the algorithm works. For each community, the set of vertices which could be potential centers of a spanning star in the induced subgraph $G'[C_i]$ are computed with respect to Claim 4.24. Then, for each group of communities, having the same center vertex according to Claim 4.24, a candidate is selected from the previously computed sets. If no such vertex exists, then $I$ is a no-instance of Stars NWS. Next, for each community in such a group the edges are computed which form a spanning star with the already selected center vertex. If at most $n - 1$ edges are selected, then a sparsified graph with at most $n - 1$ edges is found and $I$ is a yes-instance of Stars NWS. Otherwise, $I$ is a no-instance of Stars NWS.

We continue by defining the relation $R$ based on the second statement of Claim 4.24 which states when two communities have the same center vertex.

$$R \subseteq \mathcal{C} \times \mathcal{C},\ R(C_i, C_j) :\Leftrightarrow |C_i \cap C_j| \geq 3\ \forall C_i, C_j \in \mathcal{C}$$

To obtain an equivalence relation, we define $\tilde{R}$ as the reflexive, symmetric and transitive closure of $R$. The equivalence classes $\mathcal{C}/\tilde{R}$ of the equivalence relation $\tilde{R}$ are the groups of communities which have the same center vertex in $G'$ according to the second statement of Claim 4.24. Node that does not mean that two communities contained in different equivalence classes cannot have the same center vertex.

Next, we define several mappings helping us to describe which vertices are candidates for being the center of a spanning star in a subgraph induced by a community. The mapping $\nu$ describes which vertices of a community $C_i$ could be potential centers of a spanning star in $G'[C_i]$.

$$\nu : \mathcal{C} \to \mathcal{P}(V),\ C_i \mapsto \{v \in C_i \mid C_i \subseteq N[v]\}$$

The mapping $\mu$ describes which vertices of a community $C_i$ can be the center of a spanning star in $G'[C_i]$ with respected to the first statement of Claim 4.24. The first statement of Claim 4.24 states that the set of potential centers of the stars of two communities is their intersection if it has a size of at least two.

$$\mu : \mathcal{C} \to \mathcal{P}(V),\ C_i \mapsto \{v \in C_i \mid \forall C_j \in \mathcal{C} :\ |C_j \cap C_i| \geq 2 \Rightarrow v \in C_j\}$$

The relation $\tilde{R}$ and the mappings $\nu$ and $\mu$ describe restrictions which vertices are potential centers. The mapping $\varphi$ combines these three restrictions.

$$\varphi : \mathcal{C}/\tilde{R}\ \to \mathcal{P}(V),\ [C_i]_{\tilde{R}} \mapsto \bigcap_{C_k \in [C_i]_{\tilde{R}}} (\nu(C_k) \cap \mu(C_k))$$

The decision algorithm is shown in Algorithm 7. The idea is to select for each equivalence class $[C_i]_{\tilde{R}}$ a vertex forming the center of the spanning stars of the communities belonging to $[C_i]_{\tilde{R}}$. If this is not possible, that is if there is an equivalence class $[C_i]_{\tilde{R}}$ with $\varphi([C_i]_{\tilde{R}}) = \emptyset$, then the instance $I$ is a no-instance of Stars NWS. Finally, it is checked whether $|E| = n - 1$ to ensure that the sparsified graph $G'$ is a tree as observed in Claim 4.23.

**Correctness** We show that $I$ is a yes-instance of Stars NWS for $t = 0$ if and only if the algorithm returns a sparsified graph with $n - 1$ edges.

($\Leftarrow$) By the definition of $\varphi$, the vertex $u$ selected in Line 6 is a potential center of a spanning star for each community $C_j \in [C_i]_{\tilde{R}}$. In addition with the edge selection in Lines 7–9, this implies that the graph returned in Line 13 satisfies the star condition for each subgraph induced by a community $C_i \in \mathcal{C}$. Because of the conditional statement in Lines 11–12, the

**Algorithm 7:** Algorithm for Stars NWS with $\ell := n-1$ ($t = 0$)

**Input** : $I = (G = (V, E), \mathcal{C}, n-1)$

**Output:** A sparsified graph $G'$ with at most $n-1$ edges or no

1 $E' \leftarrow \emptyset$

2 **forall** $[C_i]_{\tilde{R}} \in \mathcal{C}/\tilde{R}$ **do**

3     **if** $|\varphi([C_i]_{\tilde{R}})| = 0$ **then**

4         // no center candidate available for $[C_i]_{\tilde{R}}$

5         **return** *no*

6     $u \leftarrow$ pick element of $\varphi([C_i]_{\tilde{R}})$

7     **forall** $C_j \in [C_i]_{\tilde{R}}$ **do**

8         // select the edges for the star with $u$ as center for $C_j$

9         $E' \leftarrow E' \cup \{\{u, v\} \mid v \in C_j, v \neq u\}$

10 $G' \leftarrow (V, E')$

11 **if** $|E'| > n-1$ **then**

12     **return** *no*

13 **return** $G'$

sparsified graph returned in Line 13 has at most $n-1$ edges. Hence, $I$ is a yes-instance of Stars NWS for $t = 0$.

($\Rightarrow$) Let $I$ be a yes-instance and let $G''$ be a sparsified graph with $n-1$ edges. Let $c_{G''} : \mathcal{C} \to V$ denotes the mapping of communities to their center vertex in $G''$.

By the definition of a spanning star, that the center vertex is adjacent to each other vertex in the community, we know that $c_{G''}(C_i) \in \nu(C_i)$ for each community $C_i \in \mathcal{C}$. By the first statement of Claim 4.24, we know that $c_{G''}(C_i) \in \mu(C_i)$ for each community $C_i \in \mathcal{C}$. Due to the second statement of Claim 4.24, we know that $c_{G''}(C_i) = c_{G''}(C_j)$ if $\tilde{R}(C_i, C_j)$ for two communities $C_i, C_j \in \mathcal{C}$. Hence, we have $c_{G''}(C_i) \in \varphi([C_i]_{\tilde{R}})$ for each community $C_i \in \mathcal{C}$. This implies that $\varphi([C_i]_{\tilde{R}}) \neq \emptyset$ for each equivalence class $[C_i]_{\tilde{R}} \in \mathcal{C}/\tilde{R}$. Therefore, the return statement in Line 5 is never reached.

By the definition of $\varphi$, the vertex $u$ selected in Line 6 is a potential center of a spanning star for each community $C_j \in [C_i]_{\tilde{R}}$. In addition with the edge selection in Lines 7–9, this implies that the graph $G'$ in Line 10 satisfies the star condition for each subgraph induced by a community $C_i \in \mathcal{C}$.

Next, we make two observations about the graph $G' = (V, E')$ in Line 10 regarding cycles. We differentiate two kinds of cycles in a sparsified graph $G'$ of a yes-instance $I = (G, \mathcal{C}, \ell)$ of Stars NWS. Let $c_{G'} : \mathcal{C} \to V$ denotes the mapping of communities to their center vertex in $G'$. Since there might be multiple center vertices in $G'$, the mapping may not be unique. Let $S_i$ denote for a community $C_i \in \mathcal{C}$ the spanning star of $G'[C_i]$ having the center vertex $c_{G'}(C_i)$. We say a cycle $c$ in $G'$ is *local* if there exists two communities $C_i, C_j \in \mathcal{C}$ such that $c$ is contained in $S_i \cup S_j$. Otherwise, we say a cycle $c$

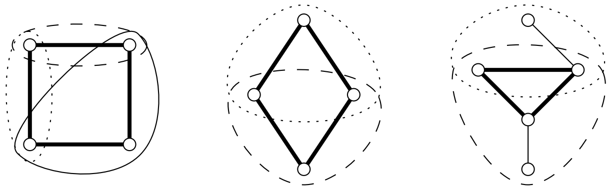

Figure 4.3: Examples for the different kinds of cycles in a sparsified graph. The bold edges mark the edges which form the cycles. On the left side a global cycle is shown. In the middle and on the right side a local cycle is shown.

in $G'$ is *global*, that is if $c$ is splittable into at least three paths $p_1, \ldots, p_r$ for $r \geq 3$ such that for each path $p_i$ there exist a community $C_i \in \mathcal{C}$ such that $p_i$ is contained in $S_i$. Note that each path $p_i$ has length one or length two because each path in a star has only length one or length-two. An example of both cycle kinds is shown in Figure 4.3.

**Claim 4.25.** *Let $G' = (V, E')$ be the graph in Line 10 and let $c_{G'} : \mathcal{C} \to V$ be the mapping of communities to their center vertex in $G'$. The graph $G'$ does not contain a local cycle.*

*Proof.* We assume towards a contradiction that $c$ is a local cycle in $G'$. By definition, there exist two communities $C_i, C_j \in \mathcal{C}$ such that the local cycle $c$ is contained in $S_i \cup S_j$. Since $S_i$ and $S_j$ are acyclic because they are stars, we have $|C_i \cap C_j| \geq 2$. Next, we distinct the two cases $|C_i \cap C_j| = 2$ and $|C_i \cap C_j| \geq 3$ each leading to a contradiction that $c$ is a local cycle in $S_i \cup S_j$.

**Case 1**: We assume $|C_i \cap C_j| = 2$. Then, there exist exactly two vertices $u, v \in C_i \cap C_j$. Since $\varphi$ is defined with respect to Claim 4.24, this implies that $c_{G'}(C_i) \in C_i \cap C_j$ and $c_{G'}(C_j) \in C_i \cap C_j$. This implies that the edge $\{u, v\}$ is contained in $S_i$ and $S_j$. Hence, we have $E(S_i \cup S_j) = |E(S_i)| + |E(S_j)| - 1 = |C_i| + |C_j| - 3 = |C_i \cup C_j| - 1$. Because $S_i$ and $S_j$ are stars, the graph $S_i \cup S_j$ is connected and a tree. This is a contradiction to the assumption that $c$ is a local cycle in $S_i \cup S_j$.

**Case 2**: We assume $|C_i \cap C_j| \geq 3$. Since $\varphi$ is defined with respect to Claim 4.24, we have $c_{G'}(C_i) = c_{G'}(C_j)$. This implies that $S_i$ and $S_j$ have $|C_i \cap C_j| - 1$ edges in common. Hence, we have $E(S_i \cup S_j) = |E(S_i)| + |E(S_j)| - (|C_i \cap C_j| - 1) = |C_i| - 1 + |C_j| - 1 - (|C_i \cap C_j| - 1) = |C_i| + |C_j| - |C_i \cap C_j| - 1 = |C_i \cup C_j| - 1$. Since $S_i$ and $S_j$ are stars and have the same center, the graph $S_i \cup S_j$ is connected and a star. This is a contradiction to the assumption that $c$ is a local cycle in $S_i \cup S_j$. ♢

**Claim 4.26.** *Let $G' = (V, E')$ be the graph in Line 10. The graph $G'$ does not contain a global cycle.*

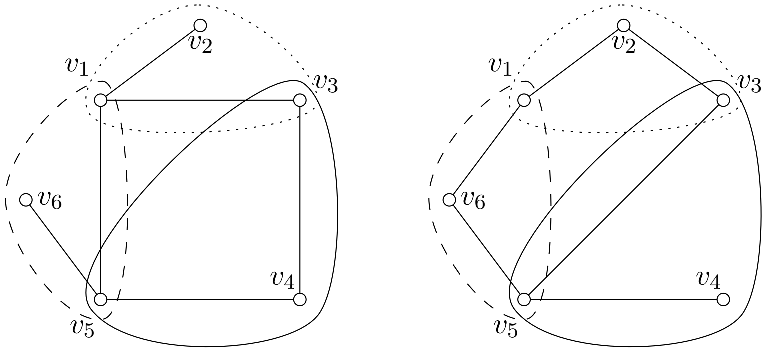


Figure 4.4: An example of two sparsified graphs of the same instance with different star centers. The left graph contains the cycle $(v_1, v_3, v_4, v_5)$ which can be decomposed into the paths $p_1 = (v_1, v_3), p_2 = (v_3, v_4, v_5), p_3 = (v_5, v_1)$. The right graph contains the cycle $(v_1, v_2, v_3, v_5, v_6)$ which can be decomposed into the paths $p'_1 = (v_1, v_2, v_3), p'_2 = (v_3, v_5), p'_3 = (v_5, v_6, v_1)$. Observe that the endpoints of the paths $p_i$ and $p'_i$ are the same.

*Proof.* We assume towards a contradiction that $c$ is a global cycle in $G'$. By definition, the global cycle is splittable into at least three paths $p_1, \ldots, p_r$ of length one or two such that each path $p_i$ is contained in $S_i$ for a community $C_i \in \mathcal{C}$. Let $s_i$ denote the start and $t_i$ the end of the path $p_i$. Thus, we have $s_i = t_{i-1}$ for $1 < i \leq r$ and $s_1 = t_r$. Due to the star requirement, the vertices $s_i$ and $t_i$ of each path $p_i$ are connected in every sparsified graph for instance $I$ of Stars NWS. An example of such a situation is shown in Figure 4.4. This implies that the subgraph consisting of the $(s_i, t_i)$-paths of every sparsified graph for instance $I$ of Stars NWS contains a cycle. Because of this every sparsified graph for $I$ contains a cycle. This implies that the graph $G''$ also contains a cycle. This is a contradiction to the assumption that $I$ is a yes-instance. $\diamondsuit$

By Claim 4.25 and Claim 4.26 the graph $G'$ in Line 10 contains neither local cycles nor global cycles. Moreover, $G'[C_i]$ contains a spanning star for each community $C_i \in \mathcal{C}$. Hence, the graph $G'$ is acyclic and connected. Therefore, Algorithm 7 finds a sparsified graph $G'$ with $n-1$ edges for the yes-instance $I$ of Stars NWS.

**Running Time** The equivalence classes $\mathcal{C}/\tilde{R}$ are computable in $O(|\mathcal{C}|^2 \cdot n)$ time. The mapping $\nu$ is computable in $O(|C| \cdot n)$ time. The mapping $\mu$ is computable in $O(|\mathcal{C}|^2 \cdot n^2)$ time. The mapping $\varphi$ is computable in $O(|C| \cdot n)$ time. The statement in Line 9 is executed at most $\sum_{[C_i]_{\tilde{R}} \in C/\tilde{R}} |[C_i]_{\tilde{R}}| = |\mathcal{C}|$ times. This leads to a running time of $O(|\mathcal{C}| \cdot n)$ for the nested loops. Hence, Algorithm 7 has a running time of $O(|\mathcal{C}|^2 \cdot n^2)$. $\square$

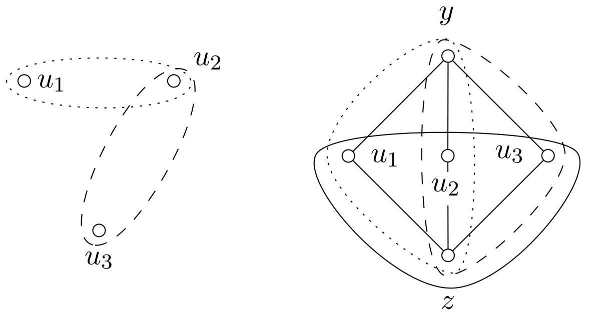


Figure 4.5: An example of the construction. The left side shows the Hitting Set instance $I_{\mathrm{HS}}$, the right side shows the Connectivity NWS instance $I_{\mathrm{CNS}}$.

In Theorem 4.22 the instances of Stars NWS are restricted to connected hypergraphs. Next, we generalize the algorithm to hypergraphs with any number of connected components.

**Corollary 4.27.** *Let $I = (G = (V, E), \mathcal{C}, |V| - x)$ be an instance of* Stars NWS *where $x$ is the number of connected components of the hypergraph $\mathcal{H} = (V, \mathcal{C})$. Such an instance $I$ is solvable in $O(\mathrm{poly}(n + |\mathcal{C}|))$ time.*

*Proof.* We split the hypergraph $\mathcal{H} = (V, \mathcal{C})$ into its connected components $\mathcal{H}_1 = (V_1, \mathcal{C}_1), \ldots, \mathcal{H}_x = (V_x, \mathcal{C}_x)$. The instance $I$ is a yes-instance of Stars NWS if and only if the instances $I_1 = (G_1 = (V_1, E(G[V_1])), \mathcal{C}_1, |V_1| - 1), \ldots, I_x = (G_x = (V_x, E(G[V_x])), \mathcal{C}_x, |V_x| - 1)$ are all yes-instances of Stars NWS. Each instance $I_i$ is solvable in $O(\mathrm{poly}(n_i + |\mathcal{C}_i|))$ time using the algorithm presented in Theorem 4.22. Overall the instance $I$ of Stars NWS is solvable in $O(\mathrm{poly}(n + |\mathcal{C}|))$ time. □

For Connectivity NWS parameterized by $t$, we get $W[2]$-hardness by a parameterized reduction from Hitting Set parameterized by the size of the hitting set.

**Theorem 4.28.** Connectivity NWS *is W[2]-hard for parameter $t$.*

*Proof.* We reduce from Hitting Set. Let $I_{\mathrm{HS}} = (U, \mathcal{S}, k)$ be an instance of Hitting Set. We start by defining the graph $G = (V, E)$. The vertex set $V$ contains every vertex of $U$ and two additional vertices $y$ and $z$. The edge set is $E := \{\{y, u\}, \{z, u\} \mid u \in U\}$ such that $G[U \cup \{y\}]$ is a star with center $y$ and $G[U \cup \{z\}]$ is a star with center $z$. Let $C_z = U \cup \{z\}$ be a community. We define the set of communities $\mathcal{C} := \{S_i \cup \{y, z\} \mid S_i \in S\} \cup \{C_z\}$. Finally, we set the parameter $t := k - 1$. Let $I_{\mathrm{CNS}} = (G, \mathcal{C}, t)$ denote the resulting instance of Connectivity NWS. An example of the construction is shown in Figure 4.5.

**Correctness** We show that $I_{\text{HS}}$ is a yes-instance of Hitting Set if and only if $I_{\text{CNS}}$ is a yes-instance of Connectivity NWS.

($\Rightarrow$) Let $X$ be the hitting set of size at most $k$. We show how to obtain a sparsified graph $G' = (V, E')$ with $|E'| = n - 1 + t$. We set $E' := \{\{z, u\} \mid u \in U\} \cup \{\{y, x\} \mid x \in X\}$ and observe that $E' = |U| + k = n - 2 + k = n - 2 + t + 1 = n - 1 + t$. For each community $C_i = \{S_i\} \cup \{y, z\} \in \mathcal{C}$, we observe that $G[S_i \cup \{z\}]$ is connected. Since $X$ is a hitting set, for each community $C_i = \{S_i\} \cup \{y, z\}$ there exists an element $x_i \in X$ with $x_i \in S_i$ which implies $\{x_i, y\} \in E'$. Hence, for each community $C_i \in \mathcal{C}$, the induced subgraph $G'[C_i]$ is connected. Therefore, $G'$ is a sparsified graph implying $I_{\text{CNS}}$ is a yes-instance of Connectivity NWS.

($\Leftarrow$) Let $I_{\text{CNS}}$ be a yes-instance and let $G' = (V, E')$ be the sparsified graph. Let $E'_z := \{\{z, u\} \mid u \in U\}$ and $E'_y := E' \setminus E'_z$. We observe that $E'_z \subseteq E'$ because the subgraph induced by the community $C_z$ is connected. Now, we show how to obtain a hitting set $X$ with $|X| = t + 1$. We set $X := \{x \mid \{x, y\} \in E'_y\}$ and observe that $|X| = |E'_y| = |E'| - |E'_z| = |V| - 1 + t - |U| = t + 1$. For each $S_i \in \mathcal{S}$, there exists a community $C_i$ such that $E(G'[C_i]) \cap E'_y \neq \emptyset$. This is the case because for every community $C_i \in \mathcal{C}$, the induced subgraph $G'[C_i]$ is connected, $y \in C_i$ and $C_i$ contains at least one element of $U$. This implies that there exists for each $S_i \in S$ an element $x_i \in X$ with $x_i \in S_i$. Therefore, $X$ is a hitting set with $|X| = t + 1$ and $I_{\text{HS}}$ is a yes-instance of Hitting Set. □

## 4.4 The Parameter $|\mathcal{C}|$

In this section, we study parameter $|\mathcal{C}|$, the number of communities. For Density NWS parameterized by $|\mathcal{C}|$, we formulate an ILP with $2^{|\mathcal{C}|}$ variables. An integer linear program with a fixed number of variables implies fixed-parameter tractability [9, 12] for the number of variables. Thus, Density NWS is fixed-parameter tractable with respect to parameter $|\mathcal{C}|$.

**Theorem 4.29.** Density NWS *parameterized by $|\mathcal{C}|$ is fixed-parameter tractable.*

*Proof.* We formulate an ILP with $2^{|\mathcal{C}|}$ variables for Density NWS. This implies fixed-parameter tractability for parameter $|\mathcal{C}|$ [9, 12]. For a better understanding what the variables of the ILP represent, we introduce the notion of edge twins.

Let $I = (G = (V, E), \mathcal{C}, \alpha, \ell)$ be an instance of Density NWS. We say two edges $e_1, e_2 \in E$ are *edge twins*, if they are contained in the subgraphs of the same communities which is formally expressed by $\forall C_i \in \mathcal{C} :\; e_1 \in E(G[C_i]) \Leftrightarrow e_2 \in E(G[C_i])$. Let $\tilde{R}$ denote the relation built upon the definition of edge twins. Note that $\tilde{R}$ is an equivalence relation. Let $E/\tilde{R}$ denote the equivalence classes of $\tilde{R}$. Since the power set of $\mathcal{C}$ has the cardinality $2^{|\mathcal{C}|}$, there are at most $2^{|\mathcal{C}|}$ different edge twin classes in the instance $I$. Hence, each edge twin class $[e]_{\tilde{R}} \in E/\tilde{R}$ is characterized by the maximal set of communities $\mathcal{X} \subseteq \mathcal{C}$ such that $[e]_{\tilde{R}} \subseteq E(G[C_i])$ for each $C_i \in \mathcal{X}$.

For each edge twin class, a variable denoted by $e_{\mathcal{X}}$ is created where $\mathcal{X}$ is the characterization of the edge twin classes mentioned above. In other words, the subscript $\mathcal{X}$ denotes for which communities $C_i \in \mathcal{C}$ the induced subgraph $G[C_i]$ contains the edges of the edge twin class $e_{\mathcal{X}}$. These variables describe how many edges of each edge twin class are retained in the sparsified graph $G'$. The goal is to minimize the sum of these variables which minimizes the number of edges in the sparsified graph.

Next, we describe the two groups of constraints. The first group of constraints enforces that the density requirement $\alpha(C_i)$ is satisfied for each community $C_i \in \mathcal{C}$. Note that $\lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil$ is a constant for each community $C_i \in \mathcal{C}$ and denotes the number of edges which are needed to satisfy the density requirement $\alpha(C_i)$. Also note that for a community $C_i$ all edge twin classes characterized by a subset $\mathcal{X}$ of $\mathcal{C}$ with $C_i \in \mathcal{C}$ contribute to the number of edges in a subgraph induced by $C_i$. This is expressed by the sum of the variables representing these edge twin classes.

The second group of constraints restricts the possible values for each variable $e_{\mathcal{X}}$. Recall that a variable $e_{\mathcal{X}}$ describes how many edges of the edge twin class characterized by $\mathcal{X}$ are retained in the sparsified graph $G'$. Thus, the value of a variable cannot be negative and also cannot exceed the number of edges in the associated edge twin class. In the following, the final ILP formulation is presented:

$$\begin{aligned} \min \quad & \sum_{\mathcal{X} \subseteq \mathcal{C}} e_{\mathcal{X}} & \\ \text{such that} \quad & \sum_{\mathcal{X} \subseteq \mathcal{C}, C_i \in \mathcal{X}} e_{\mathcal{X}} \geq \lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil & \forall C_i \in \mathcal{C} \\ & 0 \leq e_{\mathcal{X}} \leq |E(G[\bigcap_{C_i \in \mathcal{X}} C_i])| & \forall \mathcal{X} \subseteq \mathcal{C} \end{aligned}$$

□

For Stars NWS parameterized by $|\mathcal{C}|$, we are also able to obtain fixed-parameter tractability. This time we achieve fixed-parameter tractability by giving a direct algorithm instead of an integer linear program.

**Theorem 4.30.** Stars NWS *is solvable in* $O(4^{|\mathcal{C}|^2} \cdot (n+m) + n^2 \cdot |\mathcal{C}|)$ *time.*

*Proof.* Before we give an algorithm, we define *center twins* of vertices in an instance of Stars NWS. Let $I = (G = (V, E), \mathcal{C}, \ell)$ be an instance of Stars NWS. We define several mappings helping us to describe when two vertices $u, v \in V$ are center twins. The mapping $\nu$ describes which vertices of a community $C_i$ could be potential centers of a star in the sparsified graph.

$$\nu : \mathcal{C} \to \mathcal{P}(V),\ C_i \mapsto \{v \in C_i \mid C_i \subseteq N[v]\}$$

The mapping $\sigma$ describes in which communities a vertex is contained in.

$$\sigma : V \to \mathcal{P}(\mathcal{C}),\ v \mapsto \{C_i \in \mathcal{C} \mid v \in C_i\}$$

The mapping $\mu$ describes for which communities a vertex is a potential center.

$$\mu : V \to \mathcal{P}(\mathcal{C}),\ v \mapsto \{C_i \in \mathcal{C} \mid v \in \nu(C_i)\}$$

Next, we define the center twin relation $\tilde{T}$ based on the mappings $\sigma$ and $\mu$.

$$\tilde{T} \subseteq V \times V,\ \tilde{T}(u, v) :\Leftrightarrow \sigma(u) = \sigma(v) \wedge \mu(u) = \mu(v)$$

Two vertices $u, v \in V$ are center twins if they are contained in the same communities and are also potential centers of the same communities. Note that $\tilde{T}$ is an equivalence relation. Let $V/\tilde{T}$ denote the equivalence classes. Next, we make an observation about yes-instances of Stars NWS regarding center twins.

**Claim 4.31.** *Let $I$ be an yes-instance of* Stars NWS *with the sparsified graph $G' = (V, E')$ where $c_{G'} : \mathcal{C} \to V$ denotes the mapping of communities to their center vertex in $G'$. Let $u \in V$ be a vertex, let $[u]_{\tilde{T}}$ be its equivalence class of $\tilde{T}$ and let $\mathcal{C}^u_{\tilde{T}} \subseteq \mathcal{C}$ be the set of communities such that $c_{G'}(C_z) \in [u]_{\tilde{T}}$ for each $C_z \in \mathcal{C}^u_{\tilde{T}}$. There exists a sparsified graph $G'' = (V, E'')$ with $|E''| \leq |E'|$ such that $u$ is the center of all communities $C_z \in \mathcal{C}^u_{\tilde{T}}$ and the centers of the other communities $C_i \in \mathcal{C} \setminus \mathcal{C}^u_{\tilde{T}}$ are the same as in $G'$.*

*Proof.* An example of the statement is shown in Figure 4.6. We define $c_{G''}$ as the mapping of communities to their center vertex where $c_{G''}(C_z) = u$ for $C_z \in \mathcal{C}^u_{\tilde{T}}$.

$$c_{G''} : \mathcal{C} \to V,\ C_x \mapsto \begin{cases} u & C_x \in \mathcal{C}^u_{\tilde{T}} \\ c_{G'}(C_x) & \text{otherwise} \end{cases}$$

Then, we define based on this mapping the graph $G'' = (V, E'')$ with $E'' = \{\{c_{G''}(C_z), u\} \in E(G[C_z]) \mid C_z \in \mathcal{C}\}$. Observe that the induced subgraph $G''[C_z]$ contains a spanning star because $u$ is a center candidate of each $C_z \in \mathcal{C}^u_{\tilde{T}}$.

It remains to show that $|E''| \leq |E'|$. Recall that the spanning stars of two communities $C_i, C_j \in \mathcal{C}^u_{\tilde{T}}$ with $c_{G'}(C_i) \neq c_{G'}(C_j)$ have exactly one edge $e_{ij}$ in common due to the twin property of the centers. This edge $e_{ij}$ exists also in $G''$ where $c_{G'}(C_i) = c_{G'}(C_j)$. Therefore, we conclude $|\bigcup_{C_z \in \mathcal{C}^u_{\tilde{T}}} E(G''[C_z])| \leq |\bigcup_{C_z \in \mathcal{C}^u_{\tilde{T}}} E(G'[C_z])|$.

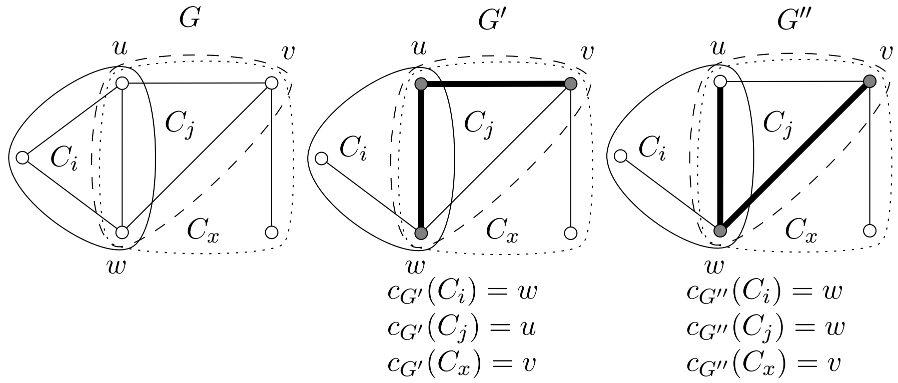


Figure 4.6: An example of the statement made by Claim 4.31. On the left side, the input graph $G$ is shown. In the middle a sparsified graph is shown, where the centers of the communities $C_i$ and $C_j$ are different while being twins. On the right side, a sparsified graph is shown, where the centers of the communities $C_i$ and $C_j$ are the same. The bold marked edges are the ones being involved in the spanning stars of more than one community.

Next, we consider the other communities affected by the new center mapping $c_{G''}$. Let $\mathcal{C}_u = \{C_i \in \mathcal{C} \mid c_{G'}(C_i) = u\}$ and $\mathcal{C}_{\hat{u}} = \mathcal{C}^u_{\tilde{T}} \setminus \mathcal{C}_u$. Changing the centers of the communities in $\mathcal{C}_{\hat{u}}$ also affects the relation of the communities $\mathcal{C}_{\hat{u}}$ and the communities $C_x \in \mathcal{C} \setminus \mathcal{C}^u_{\tilde{T}}$ with $c_{G'}(C_x) \in C_z$ and $c_{G'}(C_z) \in C_x$ for a community $C_z \in \mathcal{C}_{\hat{u}}$ because $c_{G'}(C_x)$ and $c_{G'}(C_z)$ are adjacent in $G'$. This implies that the edge $e'_{xz} = \{c_{G'}(C_x), c_{G'}(C_z)\}$ is involved in both spanning stars in $G'[C_x]$ and $G'[C_z]$. Let $E'_{XZ}$ denote these edges. In $G''$ the center of $C_z$ is changed to $u$, meaning $e'_{xz}$ is no longer involved in the spanning star of $G''[C_z]$. We observe that $u \in C_x$ because $u$ and $c_{G'}(C_z)$ are center twins. Therefore, $c_{G''}(C_x)$ and $u$ are adjacent in $G''$ meaning the edge $e''_{xz} = \{u, c_{G''}(C_x)\}$ is involved in both spanning stars in $G''[C_x]$ and $G''[C_z]$. Let $E''_{XZ}$ denote these edges. We observe $|E''_{XZ}| = |E'_{XZ}|$. Hence, we conclude $|E''| \leq |E'|$. $\diamondsuit$

**Algorithm** The idea is to branch for each community $C_i \in \mathcal{C}$ into each center twin class $[u]_{\tilde{T}} \in V/\tilde{T}$ with $u \in \mu(C_i)$. To get the actual center for the community, an arbitrary element of the center twin class is selected. Note that this element is selected such that it is always the same for the same center twin class. After a center has been selected for all communities, it is checked whether the resulting sparsified graph has at most $\ell$ edges. If this is the case, then $I$ is a yes-instance of STARS NWS. Otherwise the other branches are considered. If no branch leads to a sparsified graph with at most $\ell$ edges, then $I$ is a no-instance of STARS NWS. The decision algorithm is shown in Algorithm 8.

**Algorithm 8:** Algorithm for Stars NWS: *SolveSNWS*

**Input :** $G = (V, E), \mathcal{C}, \ell, E'$

**Output:** A sparsified graph $G'$ with at most $\ell$ edges or no

1 **if** $\mathcal{C} = \emptyset$ **then**
2   **if** $\ell < |E'|$ **then**
3     **return** *no*
4   **return** $G' = (V, E')$
5 $C_i \leftarrow$ pick element from $\mathcal{C}$
6 **forall** $[u]_{\tilde{T}} \in V/\tilde{T}$ *and* $[u]_{\tilde{T}} \subseteq \nu(C_i)$ **do**
7   $E'' \leftarrow E' \cup \{\{u, v\} \mid v \in C_i \setminus \{u\})$
8   **if** *SolveSNWS*$(G, \mathcal{C} \setminus \{C_i\}, \ell, E'')$ *returns a graph* $G'$ **then**
9     **return** $G'$
10 **return** *no*

**Correctness** We show that $I$ is a yes-instance of Stars NWS if and only if the algorithm returns a sparsified graph.

(⇒) Let $I$ be a yes-instance of Stars NWS and let $G' = (V, E')$ be a sparsified graph. By applying Claim 4.31, we are able to obtain a graph $G'' = (V, E'')$ such that $c_{G''}(C_i) = c_{G''}(C_j)$ for all pairs of communities $C_i, C_j \in \mathcal{C}$ where $c_{G'}(C_i)$ and $c_{G'}(C_i)$ are center twins. Such a graph $G''$ is definitely found by traversing the search tree built by Algorithm 8.

(⇐) Let $G'$ be the sparsified graph returned by the algorithm in Line 4. The conditional statement in Line 2 ensures that $G'$ has at most $\ell$ edges. The termination condition in Line 1 ensures together with the statement in Line 7 that for each community $C_i \in \mathcal{C}$ the induced subgraph $G'[C_i]$ contains a spanning star. Hence, a sparsified graph has been found by the algorithm which implies that $I$ is a yes-instance of Stars NWS.

**Running time** Let $C_i \in \mathcal{C}$ be the community selected for branching in Line 5. The branching vector of the branching in the loop in Lines 6-9 has at most $|V/\tilde{T}|$ entries of value 1, each decreasing the number of communities by 1. The branching number of this vector is bounded by $|V/\tilde{T}| \leq |\mathcal{P}(\mathcal{C})| \cdot |\mathcal{P}(\mathcal{C})| = 2^{|\mathcal{C}|} \cdot 2^{|\mathcal{C}|} = 4^{|\mathcal{C}|}$. Since the maximum depth of the recursion is bounded by $|\mathcal{C}|$, the search tree has a size of at most $O(4^{|C|^2})$. The equivalence classes $V/\tilde{T}$ are computable in $O(n^2 \cdot |\mathcal{C}|)$ time. The edge set in Line 7 is computable in $O(n + m)$ time. This leads to an overall running time of $O(4^{|\mathcal{C}|^2} \cdot (n + m) + n^2 \cdot |\mathcal{C}|)$. □

For Connectivity NWS parameterized by $|\mathcal{C}|$, we are neither able to prove fixed-parameter tractability nor able to prove $W[i]$-hardness for any $i \geq 1$. Next, we study problem kernels for Density NWS and Stars NWS. Under the assumption that NP $\not\subseteq$ coNP/poly, no polynomial kernels exist for

DENSITY NWS and STARS NWS when parameterized by $|\mathcal{C}|$. This is shown in the following two theorems by a polynomial parameter transformation from HITTING SET parameterized by the number of sets.

**Theorem 4.32.** DENSITY NWS *parameterized by* $|\mathcal{C}|$ *does not admit a polynomial kernel unless NP* $\subseteq$ *coNP/poly.*

*Proof.* The parameterized reduction from HITTING SET parameterized by the size of the hitting set to DENSITY NWS parameterized by $\ell$ in the proof of 4.16 is also a polynomial parameter transformation from HITTING SET parameterized by the number of sets to DENSITY NWS parameterized by $|\mathcal{C}|$. HITTING SET parameterized by the number of sets is known to not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [9]. In the following, we recall the construction briefly. Let $I_{\text{HS}} = (U, \mathcal{S}, k)$ be an instance of HITTING SET. We define the graph $G = (V, E)$. The set $V$ contains each vertex of $V$ and one additional vertex $z$. The edge set is $E := \{\{z, u\} \mid u \in U\}$. We define the set of communities $\mathcal{C} := \{S_i \cup \{z\} \mid S_i \in S\}$ and set $\alpha(C_i) := 1/\binom{|U|}{2}$ for each community $C_i \in \mathcal{C}$. Finally, we set the parameter $\ell := k$. Let $I_{\text{DNS}} = (G, \mathcal{C}, \alpha, \ell)$ denote the resulting instance of DENSITY NWS. Observe that $|S| = |\mathcal{C}|$. The correctness proof of the polynomial parameter transformation is omitted, since it is already proved in Theorem 4.16. □

**Theorem 4.33.** STARS NWS *parameterized by* $|\mathcal{C}|$ *does not admit a polynomial kernel unless NP* $\subseteq$ *coNP/poly.*

*Proof.* We give a polynomial parameter transformation from HITTING SET parameterized by the number of sets which is known to not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [9]. Let $I_{\text{HS}} = (U, \mathcal{S}, k)$ be an instance of HITTING SET. We assume that $|U| > 1$. Let $Z$ be a set of $|U|^3$ new vertices such that $U \cap Z = \emptyset$. We start by defining the graph $G = (V, E)$. The vertex set is $V := U \cup Z$ and the edge set is $E := \{\{u, v\} \mid u, v \in U, u \neq v\} \cup \{\{u, z\} \mid u \in U, z \in Z\}$. In other words, $G[U]$ is a clique and $G[G \cup \{z\}]$ is a star of size $|U|$ for each new vertex $z \in Z$. Next, we define the set of communities $\mathcal{C} := \{S_i \cup Z \mid S_i \in \mathcal{S}\}$. Finally, we set the parameter $\ell := k \cdot |U|^3 + |U|^2$. Let $I_{\text{SNS}} = (G, \mathcal{C}, \ell)$ denote the resulting instance of STARS NWS. Note that $|\mathcal{C}| = |\mathcal{S}|$, $|E| = \binom{|U|}{2} + |U|^4$ and $|V| = |U| + |U|^3$. An example of the construction is shown in Figure 4.7.

**Correctness** We show that $I_{\text{HS}}$ is a yes-instance of HITTING SET if and only if $I_{\text{SNS}}$ is a yes-instance of STARS NWS.

($\Rightarrow$) Let $X$ be a hitting set of size at most $k$. We show how to obtain a sparsified graph $G' = (V, E')$ with $|E'| \leq \ell$. We set $E' = \{\{z, x\} \mid x \in X, z \in Z\} \cup \{\{u, x\} \mid x \in X, u \in U \setminus \{x\}\}$ observing that $|E'| \leq k \cdot (|Z| + |U| - 1) = k \cdot |Z| + k \cdot |U| \leq k \cdot |U|^3 + |U|^2$. Recall that for each community $C_i \in \mathcal{C}$ a set $S_i \in \mathcal{C}$ exists, such that $C_i = S_i \cup Z$. Since $X$ is a hitting set, for each

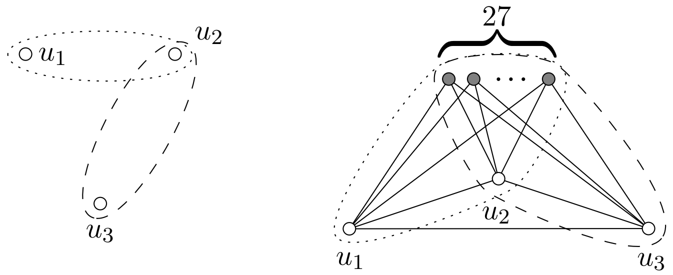


Figure 4.7: An example of the construction. The left side shows the Hitting Set instance $I_{\text{HS}}$, the right side shows the Stars NWS instance $I_{\text{SNS}}$. The grey vertices are the new vertices, the elements of the set $Z$. The different line styles show the mapping between the sets in the Hitting Set instance and the communities in the Stars NWS instance.

community $C_i$, there exists an element $x_i \in X$ with $x_i \in C_i$. Such an element $x_i$ is the center of a spanning star in $G'[C_i]$ because $\{\{z, x_i\} \mid z \in Z\} \subseteq E'$ and $\{\{z, u\} \mid u \in C_i \setminus Z\} \subseteq \{\{u, x_i\} \mid u \in U\} \subseteq E'$. Therefore, $G'$ is a sparsified graph which implies that $I_{\text{SNS}}$ is a yes-instance of Stars NWS.

($\Leftarrow$) Let $I_{\text{SNS}}$ be a yes-instance of Stars NWS and let $G' = (V, E')$ be the sparsified graph where $c_{G'} : \mathcal{C} \to V$ denotes the mapping of communities to their center vertex in $G'$. We show how to obtain a hitting set $X$ with $|X| \leq k$. We set $X = \{c_{G'}(C_i) \mid C_i \in \mathcal{C}\}$. Since $Z \subseteq C_i$ and $Z$ is an independent set in $G$, $c_{G'}(C_i) \notin Z$ holds for each community $C_i \in \mathcal{C}$. This implies that $X \subseteq U$. Since $|E'| \leq \ell = k \cdot |U|^3 + |U|^2$, we conclude that $|X| \leq k$ because $|U| > 1$. For each $S_i \in \mathcal{S}$, there exists a community $C_i$ such that $S_i = C_i \setminus Z$. This implies that for each set $S_i \in \mathcal{S}$, there exists the element $u = c_{G'}(C_i)$ with $u \in X$ and $u \in S_i$. Therefore, $X$ is a hitting set with $|X| \leq k$ and $I_{\text{HS}}$ is a yes-instance of Hitting Set. $\square$

# 5. Conclusion

In this work, we studied the complexity of Π-NWS and its three variants Density NWS, Stars NWS and Connectivity NWS. In the following, we first summarize our results and then we pose our open questions and point into directions for further work.

## 5.1 Summary

In Section 3.1, we studied NP-hardness for Density NWS, Stars NWS, Connectivity NWS, and Π-NWS restricted to communities of size at most 3. First, we recalled reductions from the literature which implies the NP-hardness of Density NWS, Stars NWS, and Connectivity NWS even if restricted to communities of size at most 3. Then, we gave a complexity dichotomy of Π-NWS restricted to communities of size at most 3 which implies that Π-NWS restricted to communities of size at most 3 is NP-hard for all graph properties Π which neither are fulfilled by an edgeless graph nor are only fulfilled by cliques.

In Section 3.2, we showed lower bounds based on the Exponential Time Hypothesis (ETH). First, we observed that the running time of a trivial brute force algorithm trying each possible spanning subgraph is $O(2^m \cdot \text{poly}(n + |\mathcal{C}|))$. Since $m \leq n^2$, the running time of this brute force algorithm is also expressible as $O(2^{n^2} \cdot \text{poly}(n + |\mathcal{C}|))$. Then, we showed that neither Density NWS, nor Stars NWS, nor Connectivity NWS is solvable in $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$ time or $2^{o(m)} \cdot \text{poly}(n + |\mathcal{C}|)$ time unless the ETH fails. These lower bounds even apply if the instances are restricted to communities of size at most 4.

In Section 4, we studied Density NWS, Stars NWS, and Connectivity NWS in the context of parameterized complexity. An overview of the results is shown in Table 5.1. With respect to parameter $\ell$, the number of edges of the sparsified graph, we obtained $W[2]$-hardness for Density NWS and fixed-parameter tractability for Stars NWS and Connectivity NWS. Furthermore, we showed that Stars NWS and Connectivity NWS do not admit a polynomial kernel for parameter $\ell$ unless NP $\subseteq$ coNP/poly. For parameter $k$, which is defined as the difference of $m$ and $\ell$, we obtained $W[1]$-hardness for Density NWS, Stars NWS, and Connectivity NWS. The $W[1]$-hardness even applies if these problems are restricted to instances with communities of size at most 3. For Stars

Table 5.1: An overview over the results regarding parameterized complexity. The entry for Density NWS parameterized by $t$ expresses that this parameterization was not studied for Density NWS.

| Parameter | Density NWS | Stars NWS | Connectivity NWS |
|---|---|---|---|
| $\ell$ | $W[2]$-hard | FPT / no polynomial kernel | |
| $k$ | $W[1]$-hard | | |
| $t$ | - | ? | $W[2]$-hard |
| $\lvert\mathcal{C}\rvert$ | FPT / no polynomial kernel | | ? |

NWS and Connectivity NWS, we studied the parameter $t$, the size of a minimal feedback edge set of the sparsified graph. We gave an polynomial-time algorithm for Stars NWS when $t = 0$ and showed $W[2]$-hardness for Connectivity NWS parameterized by $t$. For parameter $|\mathcal{C}|$, the number of communities, we obtained fixed-parameter tractability for Density NWS and Stars NWS. Moreover, we showed that no polynomial kernels exist for Density NWS and Stars NWS unless NP $\subseteq$ coNP/poly. For Connectivity NWS we were not able to show either fixed-parameter tractability or $W[1]$-hardness.

## 5.2 Future Work

Our study of Density NWS, Stars NWS and Connectivity NWS in the context of parameterized complexity leaves mainly two questions open. The first open question is whether Connectivity NWS parameterized by $|\mathcal{C}|$ is fixed-parameter tractable. The second open question is whether Stars NWS parameterized by $t$ is fixed-parameter tractable.

Since $\ell$ can be quadratic in $n$, the absence of a $2^{o(n^2)} \cdot \text{poly}(n + |\mathcal{C}|)$-time algorithm only implies the absence of a $2^{o(\ell)} \cdot \text{poly}(n + |\mathcal{C}|)$-time algorithm for Stars NWS and Connectivity NWS. Thus, there is a discrepancy between the lower bound and the running time of the given FPT-algorithms. Hence, another open question is whether there exist single-exponential time FPT-algorithms for Stars NWS and Connectivity NWS parameterized by $\ell$.

Another direction for further work might be studying additional parameters or combinations of the parameters. For example the parameter $d + t$ for Connectivity NWS might be interesting, since Connectivity NWS parameterized by $t$ solely is $W[2]$-hard.

A new parameter that might be interesting is the difference between an upper bound of parameter $\ell$ and the actual parameter $\ell$, the number of edges in the sparsified graph. The upper bound, we suggest, is the minimum number of edges a sparsified graph contains assuming that all communities are pairwise disjoint. For an instance $I = (G, \mathcal{C}, \alpha, \ell)$ of Density NWS this upperbound is $\sum_{C_i \in \mathcal{C}} (\lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil)$. For an instance $I = (G, \mathcal{C}, \ell)$ of Stars NWS or Connectivity NWS this upper bound is $\sum_{C_i \in \mathcal{C}} (|C_i| - 1)$. The final parameter is then defined as $s := \sum_{C_i \in \mathcal{C}} (\lceil \alpha(C_i) \cdot \binom{|C_i|}{2} \rceil) - \ell$ for Density NWS and as $s := \sum_{C_i \in \mathcal{C}} (|C_i| - 1) - \ell$ for Stars NWS and Connectivity NWS.

There are two kinds of edges in a subgraph induced by a community in a sparsified graph. First, there are edges which are essential in the sense that the subgraph without these edges does not fulfill the required graph property. Second, there are edges which can be removed such that the subgraph still satisfies the required graph property. Informally, the parameter $s$ is a measure how many edges are essential for more than one community. Since one edge can be essential for more than two communities, the parameter $s$ is an upper bound for the number of essential edges.

Moreover, it might also be interesting to study $\Pi$-NWS for other graph properties or to use the characterization of graph properties in Section 3.1 to obtain more general results for $\Pi$-NWS in the context of parameterized complexity.

# Bibliography

[1] D. Angluin, J. Aspnes, and L. Reyzin. Network construction with subgraph connectivity constraints. *Journal of Combinatorial Optimization*, 29(2):418–432, 2015.

[2] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[3] G. Bodwin. A note on distance-preserving graph sparsification. *Information Processing Letters*, 174:106205, 2022.

[4] É. Bonnet, D.-E. Fălămaş, and R. Watrigant. Constraint generation algorithm for the minimum connectivity inference problem. In *International Symposium on Experimental Algorithms*, pages 167–183. Springer, 2019.

[5] C. Chekuri and N. Korula. A graph reduction step preserving element-connectivity and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 254–265. Springer, 2009.

[6] C. Chekuri, T. Rukkanchanunt, and C. Xu. On element-connectivity preserving graph simplification. In *Algorithms-ESA 2015*, pages 313–324. Springer, 2015.

[7] J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010.

[8] J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchy, and M. Weller. Polynomial-time data reduction for the subset interconnection design problem. *SIAM Journal on Discrete Mathematics*, 29(1):1–25, 2015.

[9] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.

[10] M. A. Dar, A. Fischer, J. Martinovic, and G. Scheithauer. Integer linear programming formulations for the minimum connectivity infer-

ence problem and model reduction principles. *Discrete Optimization*, 40:100623, 2021.

[11] R. Diestel. *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition, 2017.

[12] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[13] P. Erdös and R. Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, 1960.

[14] H. Fan, C. Hundt, Y.-L. Wu, and J. Ernst. Algorithms and implementation for interconnection graph problem. In *International Conference on Combinatorial Optimization and Applications*, pages 201–210. Springer, 2008.

[15] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[16] A. Gionis, P. Rozenshtein, N. Tatti, and E. Terzi. Community-aware network sparsification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 426–434. SIAM, 2017.

[17] O. Goldreich. Computational complexity: a conceptual perspective. *ACM Sigact News*, 39(3):35–39, 2008.

[18] S. Gupta, P. Jain, A. Petety, and S. Singh. Parameterized complexity of d-hitting set with quotas. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 293–307. Springer, 2021.

[19] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[20] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[21] E. Korach and M. Stern. The complete optimal stars-clustering-tree problem. *Discrete Applied Mathematics*, 156(4):444–450, 2008.

[22] G. Lindner, C. L. Staudt, M. Hamann, H. Meyerhenke, and D. Wagner. Structure-preserving sparsification of social networks. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 448–454. IEEE, 2015.

[23] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 529–537, 2011.

[24] D. Mellor, E. Prieto, L. Mathieson, and P. Moscato. A kernelisation approach for multiple d-hitting set and its application in optimal multi-drug therapeutic combinations. *PLoS One*, 5(10):e13055, 2010.

[25] P. Parchas, N. Papailiou, D. Papadias, and F. Bonchi. Uncertain graph sparsification. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2435–2449, 2018.

[26] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4(1):1–27, 2011.

[27] M. Wahlström. *Algorithms, measures and upper bounds for satisfiability and related problems.* PhD thesis, Department of Computer and Information Science, Linköpings universitet, 2007.

[28] S. Wasserman, K. Faust, et al. Social network analysis: Methods and applications. 1994.

[29] H. Yang and M. G. H. Bell. Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3):257–278, 1998.

[30] F. Zhou, S. Malher, and H. Toivonen. Network simplification with minimal loss of connectivity. In *2010 IEEE international conference on data mining*, pages 659–668. IEEE, 2010.