

# Learning Bayesian Networks Under Sparsity Constraints: A Parameterized Complexity Analysis

**Niels Grüttemeier**  
**Christian Komusiewicz**  
*Hans-Meerwein-Straße 6*  
*35032 Marburg, Germany*

NIEGRU@INFORMATIK.UNI-MARBURG.DE  
 KOMUSIEWICZ@INFORMATIK.UNI-MARBURG.DE

## Abstract

We study the problem of learning the structure of an optimal Bayesian network when additional constraints are posed on the network or on its moralized graph. More precisely, we consider the constraint that the network or its moralized graph are close, in terms of vertex or edge deletions, to a sparse graph class  $\Pi$ . For example, we show that learning an optimal network whose moralized graph has vertex deletion distance at most  $k$  from a graph with maximum degree 1 can be computed in polynomial time when  $k$  is constant. This extends previous work that gave an algorithm with such a running time for the vertex deletion distance to edgeless graphs. We then show that further extensions or improvements are presumably impossible. For example, we show that learning optimal networks where the network or its moralized graph have maximum degree 2 or connected components of size at most  $c$ ,  $c \geq 3$ , is NP-hard. Finally, we show that learning an optimal network with at most  $k$  edges in the moralized graph presumably has no  $f(k) \cdot |I|^{\mathcal{O}(1)}$ -time algorithm and that, in contrast, an optimal network with at most  $k$  arcs can be computed in  $2^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$  time where  $|I|$  is the total input size.

## 1. Introduction

Bayesian networks are graphical models for probability distributions in which the presence of statistical dependencies between a set of random variables are represented via a directed acyclic graph (DAG)  $D = (N, A)$  over a set  $N$  of  $n$  random variables (Darwiche, 2009). An arc from a vertex  $u$  to a vertex  $v$  in a Bayesian network means that the distribution of  $v$  might depend on the value of  $u$ , while the absence of an arc means that  $u$  and  $v$  are conditionally independent given some set of variables. Once we have obtained a Bayesian network, one may infer the distribution of some random variables given the values of other random variables.

First, however, one needs to learn the network from observed data. An important step herein is to learn the *structure* of the network, that is, the arc set of the corresponding DAG. This problem is known as BAYESIAN NETWORK STRUCTURE LEARNING (BNSL). In BNSL, one is given for each network vertex  $v$  and each set of possible parents of  $v$  a parent score and the goal is to learn an acyclic network with a maximal sum of parent scores. To represent the observed data as closely as possible, it may seem appropriate to learn a tournament, that is, a DAG in which every pair of vertices  $u$  and  $v$  is connected either by the arc  $(u, v)$  or by the arc  $(v, u)$ . There are, however, several reasons why learning a tournament-like DAG should be avoided (For a detailed discussion we refer to the book by Darwiche (2009)): First, such a network gives no information about which variables are

conditionally independent. Second, including too many dependencies in the model makes the model vulnerable to overfitting. Finally, the problem of inferring distributions on a given Bayesian network is intractable when the DAG is tournament-like. More precisely, the inference problem on Bayesian networks is NP-hard (Cooper, 1990). When the network is tree-like, however, efficient inference algorithms are possible: If the moralized graph has small treewidth, the inference task can be solved more efficiently (Darwiche, 2009); the moralized graph of a network  $D$  is the undirected graph on the same vertex set that is obtained by adding an edge between each pair of vertices that is adjacent or has a common child in  $D$ .

Motivated by these reasons for avoiding tournament-like networks and instead aiming for tree-like networks, it has been proposed to learn optimal networks under structural constraints that guarantee that the network or its moralized graph is tree-like (Elidan & Gould, 2008; Korhonen & Parviainen, 2013, 2015; Chow & Liu, 1968; Dasgupta, 1999; Gaspers, Koivisto, Liedloff, Ordyniak, & Szeider, 2015). We continue this line of research, focusing on exact algorithms with worst-case running time guarantees. In other words, we want to find out for which structural constraints there are fast algorithms for learning optimal Bayesian networks under these constraints and for which constraints this is presumably impossible.

**Known Results.** The problem of learning a Bayesian network without structural constraints, which we call VANILLA-BNSL, is NP-hard (Chickering, 1995) and can be solved in  $2^n n^{\mathcal{O}(1)}$  time by dynamic programming over all subsets of  $N$  (Ott & Miyano, 2003; Silander & Myllymäki, 2006).

When the network is restricted to be a branching, that is, a directed tree in which every vertex has indegree at most one, then an optimal network can be computed in polynomial time (Chow & Liu, 1968; Gaspers et al., 2015). Note that learning a more restricted Bayesian network is not necessarily easier: While learning a branching is solvable in polynomial time, the problem becomes NP-hard if we aim to learn a directed path (Meek, 2001).

On the negative side, BNSL where the moralized graph of the network is restricted to have treewidth at most  $\omega$  is NP-hard for every fixed  $\omega \geq 2$  and can be solved in  $3^n n^{\omega + \mathcal{O}(1)}$  time (Korhonen & Parviainen, 2013). Finally, Korhonen and Parviainen (2015) considered a structural constraint that restricts the treewidth of the moralized graph by restricting the size of its vertex cover. A *vertex cover* in a graph  $G$  is a vertex set  $S$  such that every edge of  $G$  has at least one endpoint in  $S$ . Korhonen and Parviainen (2015) showed that BNSL where the moralized graph is restricted to have a vertex cover of size at most  $k$  can be solved in  $4^k \cdot n^{2k + \mathcal{O}(1)}$  time. Since having a bounded vertex cover—we refer to graphs with this property as bounded-vc graphs—implies that the graph has bounded treewidth, the networks that are learned by BNSL with bounded-vc moralized graphs allow for fast inference algorithms. An algorithm with running time  $f(k) \cdot |I|^{\mathcal{O}(1)}$  is unlikely for this BNSL variant, since it is W[1]-hard with respect to the parameter  $k$  (Korhonen & Parviainen, 2015). Here,  $|I|$  denotes the total input size. In other words, it seems necessary that the degree of the running time polynomial depends on  $k$ .

**Our Results.** Extending previous work, we provide an almost complete picture of the parameterized and classical complexity of BNSL with respect to several constraints that guarantee tree-like networks or moralized graphs. Since the constraints are formulated in terms of undirected graphs, we will refer to the undirected underlying graph of a network

Table 1: An overview of the parameterized complexity of constrained BNSL problems for an input parameter  $k$  that upper-bounds structural parameters in the resulting skeleton or moralized graph. The *dissociation number* is the minimum size of a vertex set  $S$ , such that after removing  $S$ , the maximum degree is one. The *distance to degree 2* is the minimum size of a vertex set  $S$ , such that after removing  $S$ , the maximum degree is 2. The *c-component order connectivity (c-COC)* is the minimum size of a vertex set  $S$ , such that after removing  $S$ , every connected component has order at most  $c$ .

Bounded by $k$	Skeleton	Moralized Graph
Treewidth	NP-h for $k = 1$ (Dasgupta, 1999)	NP-h for $k = 2$ (Korhonen & Parviainen, 2013)
Vertex cover number	XP (Thm 6) W[2]-h (Thm 7)	XP (Korhonen & Parviainen, 2015) W[2]-h (Cor 8)
Dissociation number	W[2]-h (Thm 7)	XP (Thm 17) W[2]-h (Cor 8)
Distance to degree 2	NP-h for $k = 0$ (Thm 19)	NP-h for $k = 0$ (Thm 19)
$c$ -COC for $c \geq 3$	NP-h for $k = 0$ (Thm 20)	NP-h for $k = 0$ (Thm 20)
Number of edges	FPT (Cor 27) no $k^{\mathcal{O}(1)}$ kernel (Cor 33)	XP (Prop 28) W[1]-h (Thm 29)
Feedback edge set	NP-h for $k = 0$ (Dasgupta, 1999)	W[1]-h (Thm 31)

as its *skeleton*. An overview of our results and previous results for the considered problems is given in Table 1.

The results for BNSL with bounded-vc moralized graphs (Korhonen & Parviainen, 2015) form the starting point for our work. We show that BNSL with bounded-vc skeletons can be solved in polynomial time when the vertex cover size bound  $k$  is a constant. Moreover, we show that, as for bounded-vc moralized graphs, an algorithm with running time  $f(k) \cdot |I|^{\mathcal{O}(1)}$  is unlikely since BNSL with bounded-vc skeletons is W[2]-hard.

After complementing the results for BNSL with bounded-vc moralized graphs by results for its skeleton counter part, we consider further, related structural constraints. To this end, we take the following alternative view of vertex covers: A graph has a vertex cover of size  $k$  if and only if it can be transformed into an edgeless graph by  $k$  vertex deletions. Thus, in BNSL with bounded-vc moralized graphs we learn a network whose moralized graph is close, in terms of the number of vertex deletions, to a sparse graph class. We investigate whether there are further positive examples for such constrained network learning problems.

First, we consider the constraint that the skeleton or the moralized graph can be transformed into a graph with maximum degree 1 by at most  $k$  vertex deletions. This property is

also known as having *dissociation number* at most  $k$  and we refer to graphs with this property as bounded-diss-number graphs in the following. We show that under this constraint, BNSL with bounded-diss-number moralized graphs can be solved in  $n^{3k} \cdot k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$  time and thus in polynomial time for every constant value of  $k$ . This extends the result for BNSL with bounded-vc moralized graphs in the following sense: the value of  $k$  can be arbitrarily smaller than the vertex cover number of the resulting network and thus for fixed  $k$  our algorithm can learn an optimal network for a larger class of graphs than the algorithm for BNSL with bounded-vc moralized graphs. Observe that moralized graphs with bounded dissociation number still have bounded treewidth and thus inference on the learned networks will still be solvable efficiently. On the negative side, we show that an algorithm with running time  $f(k) \cdot |I|^{\mathcal{O}(1)}$  is unlikely since the problem is W[2]-hard. The latter hardness result also holds for BNSL with bounded-diss-number skeletons; we did not obtain a positive result for this case, however.

We then consider two further constraints that are related to the dissociation number: We show that learning an optimal network whose skeleton or moralized graph has maximum degree 2 is NP-hard and that learning an optimal network in which every component of the skeleton or the moralized graph has at most  $c$  vertices,  $c \geq 3$ , is NP-hard. The latter constraint is related to the dissociation number since in a graph with maximum degree 1 every connected component has at most two vertices.

Next, we consider constraints that are formulated in terms of edge sets of the skeleton or the moralized graph. We show that optimal networks with at most  $k$  arcs can be found in time  $2^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$ . In contrast, when we aim to compute an optimal network whose moralized graph has at most  $k$  edges, an  $f(k) \cdot |I|^{\mathcal{O}(1)}$ -time algorithm is unlikely. Thus, putting structural constraints on the moralized graph may make the problem much harder than putting similar structural constraints on the skeleton.

Furthermore, we consider the case where the edge deletion distance to trees is measured, in other words, the case where the skeleton or the moralized graph have a feedback edge set of size at most  $k$ . BNSL with tree skeletons is known as POLYTREE LEARNING. Thus, the learning problem for skeletons with feedback edge sets of size at most  $k$  is NP-hard even for  $k = 0$  (Dasgupta, 1999). For BNSL with moralized graphs with feedback edge sets of size at most  $k$ , we obtain the first hardness result: we show that an algorithm with running time  $f(k) \cdot |I|^{\mathcal{O}(1)}$  is unlikely, since the problem is W[1]-hard; an algorithm with running time  $n^{f(k)}$  is however still possible.

Finally, we obtain a further hardness result for VANILLA-BNSL: Under standard assumptions in complexity theory, it is impossible that we can transform a given instance of VANILLA-BNSL in polynomial time to an equivalent one of size  $n^{\mathcal{O}(1)}$ . Thus, it is sometimes necessary to keep an exponential number of parent scores to compute an optimal network.

Altogether, our results reveal that the difficulty of the learning problem may differ depending on whether we put the constraints on the skeleton or the moralized graph. Moreover, more general networks than those with bounded-vc moralized graphs can be computed efficiently. The room for generalization seems, however, very limited as even learning networks with constant degree or constant component size is NP-hard.

## 2. Preliminaries

In this section, we give an overview of the central definitions and concepts that we use in this paper.

### 2.1 Notation

A directed graph  $D = (N, A)$  consists of a *vertex set*  $N$  and an *arc set*  $A \subseteq N \times N$ . Let  $D = (N, A)$  be a directed graph. If  $D$  does not contain directed cycles, then  $D$  is a *directed acyclic graph* (DAG). An arc  $(u, v) \in A$  is called *incoming arc into  $v$*  and *outgoing arc from  $u$* . Given a vertex  $v$ , the number of incoming arcs into  $v$  is called *in-degree of  $v$* , and the number of outgoing arcs from  $v$  is the *out-degree of  $v$* . A vertex without incoming arcs is a *source*. A vertex without outgoing arcs is a *sink*. The set  $P_v^A := \{u \in N \mid (u, v) \in A\}$  is called *parent set of  $v$* . The vertices in  $P_v^A$  are called *parents of  $v$*  and for every  $u \in P_v^A$ , the vertex  $v$  is called *child of  $u$* . We call  $v_1$  an *ancestor of  $v_\ell$*  and  $v_\ell$  a *descendant of  $v_1$*  if there is a directed path  $(v_1, v_2, \dots, v_\ell)$  in  $D$ .

An undirected graph  $G = (V, E)$  consists of a vertex set  $V$  and an *edge set*  $E \subseteq \{\{u, v\} \mid u, v \in V\}$ . For a vertex  $v \in V$ , we write  $N_G(v) := \{u \mid \{u, v\} \in E\}$  to denote the neighborhood of  $v$  in  $G$ . The *degree of a vertex  $v$*  is denoted by  $\deg_G(v) := |N_G(v)|$ . For  $V_1, V_2 \subseteq V$ , we write  $E_G(V_1, V_2) := \{\{v_1, v_2\} \in E \mid v_1 \in V_1, v_2 \in V_2\}$  for the set of edges between  $V_1$  and  $V_2$ . Moreover we set  $E_G(K) := E_G(K, K)$ . Given an edge set  $E' \subseteq E$ , we let  $G \setminus E'$  denote the graph we obtain after deleting the edges of  $E'$  from  $G$ . Given a vertex set  $V' \subseteq V$ , we let  $G - V'$  denote the graph we obtain after deleting the vertices in  $V'$  and their incident edges from  $G$ . A set  $T \subseteq E$  is called *feedback edge set* if  $G \setminus T$  contains no cycles. The size of a smallest possible feedback edge set for  $G$  is the *feedback edge number of  $G$* . A set  $S \subseteq V$  is called *dissociation set* if  $G - S$  has maximum degree 1. The size of a smallest possible dissociation set for  $G$  is the *dissociation number of  $G$* .

A *graph class*  $\Pi$  is a set of undirected graphs. For a graph class  $\Pi$  and  $k \in \mathbb{N}$ , let  $\Pi + kv := \{G = (V, E) \mid \exists V' \subseteq V : (|V'| \leq k \wedge G - V' \in \Pi)\}$  denote the class of graphs that can be transformed into a graph in  $\Pi$  by performing at most  $k$  vertex deletions. Analogously, we define  $\Pi + ke := \{G = (V, E) \mid \exists E' \subseteq E : (|E'| \leq k \wedge G \setminus E' \in \Pi)\}$  as the class of graphs that can be transformed into a graph in  $\Pi$  by performing at most  $k$  edge deletions. We call  $\Pi$  *monotone* if  $\Pi$  is closed under edge- and vertex deletions. Note that  $\Pi$  being monotone implies that for every  $k \in \mathbb{N}_0$ , the graph classes  $\Pi + kv$  and  $\Pi + ke$  are monotone.

### 2.2 Bayesian Network Structure Learning

**Problem Definitions.** Given a vertex set  $N$ , we call a family  $\mathcal{F} = \{f_v : 2^{N \setminus \{v\}} \rightarrow \mathbb{N}_0 \mid v \in N\}$  a family of *local scores* for  $N$ . Intuitively, for a vertex  $v \in N$  and some  $P \in 2^{N \setminus \{v\}}$ , the value  $f_v(P) \in \mathbb{N}_0$  represents the score we obtain if we choose exactly the vertices of  $P$  as parents for  $v$ . Given a vertex set  $N$ , local scores  $\mathcal{F}$ , and some integer  $t \in \mathbb{N}_0$ , an arc set  $A \subseteq N \times N$  is called  $(N, \mathcal{F}, t)$ -*valid* if  $(N, A)$  is a DAG and  $\sum_{v \in N} f_v(P_v^A) \geq t$ . In BAYESIAN NETWORK STRUCTURE LEARNING (VANILLA-BNSL), the input is a vertex set  $N$ , local scores  $\mathcal{F}$ , and an integer  $t$ . The question is, whether there exists an  $(N, \mathcal{F}, t)$ -valid arc set.

In this work, we study VANILLA-BNSL under additional sparsity constraints. These sparsity constraints are posed on the skeleton and the moralized graph (Elidan & Gould, 2008) of the network. Let  $D := (N, A)$  be a DAG. The *skeleton of  $D$*  is the undirected graph  $\mathcal{S}(D) := (N, E)$ , with  $E := \{\{u, v\} \mid (u, v) \in A\}$ . The *moralized graph of  $D$*  is the undirected graph  $\mathcal{M}(D) := (N, E_1 \cup E_2)$  where the edge set is defined by  $E_1 := \{\{u, v\} \mid (u, v) \in A\}$  and  $E_2 := \{\{u, v\} \mid u \text{ and } v \text{ have a common child in } D\}$ . The edges in  $E_2$  are called *moral edges*. Given a DAG  $(N, A)$  we write  $\mathcal{S}(N, A) := \mathcal{S}((N, A))$  and  $\mathcal{M}(N, A) := \mathcal{M}((N, A))$  for sake of readability. The problems are defined as follows.

( $\Pi + v$ )-SKELETON BAYESIAN NETWORK STRUCTURE LEARNING (( $\Pi + v$ )-SKELETON BNSL)

**Input:** A set of vertices  $N$ , local scores  $\mathcal{F} = \{f_v \mid v \in N\}$ , and two integers  $t, k \in \mathbb{N}_0$ .

**Question:** Is there an  $(N, \mathcal{F}, t)$ -valid arc set  $A \subseteq N \times N$  such that  $\mathcal{S}(N, A) \in \Pi + kv$ ?

( $\Pi + v$ )-MORAL BAYESIAN NETWORK STRUCTURE LEARNING (( $\Pi + v$ )-MORAL BNSL)

**Input:** A set of vertices  $N$ , local scores  $\mathcal{F} = \{f_v \mid v \in N\}$ , and two integers  $t, k \in \mathbb{N}_0$ .

**Question:** Is there an  $(N, \mathcal{F}, t)$ -valid arc set  $A \subseteq N \times N$  such that  $\mathcal{M}(N, A) \in \Pi + kv$ ?

Furthermore, we define the problems ( $\Pi + e$ )-SKELETON-BNSL and ( $\Pi + e$ )-MORAL-BNSL on the same input and we ask if there exists an  $(N, \mathcal{F}, t)$ -valid arc set  $A$  such that  $\mathcal{S}(N, A) \in \Pi + ke$  or  $\mathcal{M}(N, A) \in \Pi + ke$ , respectively. Given a graph class  $\Pi$ , we refer to all problems described above as *constrained BNSL problems for  $\Pi$* . For a constrained BNSL problem we refer to the constraint on  $\mathcal{S}$  or  $\mathcal{M}$  as *sparsity constraint*. Given an instance  $I$  of a constrained BNSL problem for some  $\Pi$ , we call the requested arc set  $A$  a *solution of  $I$* . Note that, if  $\Pi$  is monotone and  $k = n$ , then the sparsity constraints  $\mathcal{S}(N, A) \in \Pi + kv$  and  $\mathcal{M}(N, A) \in \Pi + kv$  always hold, since the empty graph belongs to  $\Pi$ . Moreover, if  $\Pi$  is monotone and contains infinitely many graphs and  $k = n^2$ , then the sparsity constraints  $\mathcal{S}(N, A) \in \Pi + ke$  and  $\mathcal{M}(N, A) \in \Pi + ke$  always hold, since the edgeless graph on  $n$  vertices belongs to  $\Pi$  for every  $n$ . Hence, all problems considered in this work are generalizations of VANILLA-BNSL and thus NP-hard for every monotone and infinite  $\Pi$ . For formal reasons, the problems are stated as decision problems. However, the algorithms presented in this work solve the corresponding optimization problem within the same running time.

**Input Representation.** Throughout this work, we let  $n := |N|$  denote the number of vertices given in an instance  $I = (N, \mathcal{F}, t, k)$  of a constrained BNSL problem. Furthermore, we assume that for  $N = \{v_1, \dots, v_n\}$ , the local scores  $\mathcal{F}$  are given as a two-dimensional array  $\mathcal{F} := [Q_1, Q_2, \dots, Q_n]$ , where each  $Q_i$  is an array containing all triples  $(f_{v_i}(P), |P|, P)$  where  $f_{v_i}(P) > 0$  or  $P = \emptyset$ . Note that the triple  $(f_{v_i}(P), |P|, P)$  of a non-empty parent set  $P$  is only part of the input if its local score  $f_{v_i}(P)$  is not 0. This input representation is known as *non-zero representation* (Ordyniak & Szeider, 2013). The size  $|\mathcal{F}|$  is then defined as the number of bits we need to store this two-dimensional array. As the *size of  $I$*  we define  $|I| := n + |\mathcal{F}| + \log(t) + \log(k)$ .

**Basic Observations.** Let  $I$  be a yes-instance of a constrained BNSL problem. We call a solution  $A$  for  $I$  *nice* if  $f_v(P_v^A) \leq f_v(\emptyset)$  implies  $P_v^A = \emptyset$ . In this work, we consider constrained BNSL problems for some monotone graph classes  $\Pi$ . We next show that in these cases every yes-instance has a nice solution  $A$ .

**Proposition 1** *Let  $\Pi$  be a monotone graph property, and let  $(N, \mathcal{F}, t, k)$  be a yes-instance of a constrained BNSL problem for  $\Pi$ . Then, there exists a nice solution  $A$  for  $(N, \mathcal{F}, t, k)$ .*

PROOF Let  $A$  be a solution for  $I := (N, \mathcal{F}, t, k)$  such that there exist vertices  $v_1, \dots, v_\ell \in N$  with  $P_{v_i}^A \neq \emptyset$  and  $f_{v_i}(P_{v_i}^A) \leq f_{v_i}(\emptyset)$ . We then set  $A' := A \setminus \{(u, v_i) \mid u \in N, i \in \{1, \dots, \ell\}\}$ . Observe that  $P_{v_i}^{A'} = \emptyset$  for all  $i \in \{1, \dots, \ell\}$ . Moreover,  $f_v(P_v^{A'}) \geq f_v(P_v^A)$  for every  $v \in N$  and  $(N, A')$  is a DAG. Therefore,  $A'$  is  $(N, \mathcal{F}, t)$ -valid. Finally, since  $\Pi$  is monotone and  $\mathcal{S}(N, A)$  (or  $\mathcal{M}(N, A)$ , respectively) satisfies the sparsity constraint,  $\mathcal{S}(N, A')$  (or  $\mathcal{M}(N, A')$ , respectively) also satisfies the sparsity constraint.  $\square$

Observe that an instance  $I := (N, \mathcal{F}, t, k)$  of a constrained BNSL problem for some monotone  $\Pi$  is a trivial yes-instance if  $\sum_{v \in N} f_v(\emptyset) \geq t$ , since the empty arc set is a solution of  $I$ . Throughout this work, we assume that for a non-trivial instance of a constrained BNSL problem it holds that  $f_v(\emptyset) = 0$  for every  $v \in N$ . With the next proposition we assure that every instance can be preprocessed in  $\mathcal{O}(|\mathcal{F}|)$  time into an instance that satisfies our assumption.

**Proposition 2** *Let  $\Pi$  be a graph class, and let  $I := (N, \mathcal{F}, t, k)$  be an instance of a constrained BNSL problem for  $\Pi$  where  $t \geq \sum_{v \in N} f_v(\emptyset)$ . Then, there exist  $\mathcal{F}' := \{f'_v \mid v \in N\}$  with  $f'_v(\emptyset) = 0$  for every  $v \in N$  and  $t' \in \mathbb{N}_0$ , such that an arc set  $A$  is a nice solution for  $I$  if and only if  $A$  is a nice solution for  $I' := (N, \mathcal{F}', t', k)$ . Furthermore,  $I'$  can be computed in  $\mathcal{O}(|\mathcal{F}|)$  time.*

PROOF Let  $v \in N$ . We define the new local scores  $f'_v$  by setting  $f'_v(P) := f_v(P) - f_v(\emptyset)$ , if  $f_v(P) \geq f_v(\emptyset)$ , and  $f'_v(P) := 0$  otherwise. Note that  $f'_v(\emptyset) = 0$  for all  $v \in N$ . Furthermore, we set  $t' := t - \sum_{v \in N} f_v(\emptyset)$ . Obviously,  $\mathcal{F}'$  and  $t'$  can be computed in  $\mathcal{O}(|\mathcal{F}|)$  time by iterating over the two-dimensional array representing  $\mathcal{F}$ . Moreover,  $t' \geq 0$  since  $t \geq \sum_{v \in N} f_v(\emptyset)$ . We next show that  $A \subseteq N \times N$  is a nice solution for  $I$  if and only if  $A$  is a nice solution for  $I'$ .

( $\Rightarrow$ ) Let  $A$  be a nice solution for  $I$ . Obviously,  $(N, A)$  is a DAG and the sparsity constraint is satisfied. Furthermore, we have

$$\begin{aligned} \sum_{v \in N} f'_v(P_v^A) &\geq \sum_{v \in N} (f_v(P_v^A) - f_v(\emptyset)) \\ &\geq t - \sum_{v \in N} f_v(\emptyset) = t'. \end{aligned}$$

It remains to show that  $A$  is nice for  $I'$ . To this end, let  $f'_v(P_v^A) \leq f'_v(\emptyset)$ . We conclude  $f'_v(P_v^A) = 0$  and therefore  $f_v(P_v^A) \leq f_v(\emptyset)$ . Since  $A$  is nice for  $I$ , we conclude  $P_v^A = \emptyset$ . Hence,  $A$  is a nice solution for  $I'$ .

( $\Leftarrow$ ) Conversely, let  $A$  be nice for  $I'$ . We show that  $A$  is a nice solution for  $I$ . Obviously  $(N, A)$  is a DAG and the sparsity constraint is satisfied. Hence, it remains to show that  $\sum_{v \in N} f_v(P_v^A) \geq t$  and that  $A$  is nice for  $I$ .

To this end, we first show that for every  $v \in N$  it holds that  $f'_v(P_v^A) = f_v(P_v^A) - f_v(\emptyset)$ . Assume towards a contradiction that there exists some  $v \in N$  such that  $f'_v(P_v^A) \neq f_v(P_v^A) - f_v(\emptyset)$ . It then follows by the definition of  $\mathcal{F}'$ , that  $f_v(P_v^A) < f_v(\emptyset)$  and that  $f'_v(P_v^A) = 0$ . Note that  $f'_v(P_v^A) = 0$  implies  $f'_v(P_v^A) \leq f'_v(\emptyset)$  and therefore  $P_v^A = \emptyset$  since  $A$  is nice for  $I'$ . This contradicts the fact that  $f_v(P_v^A) < f_v(\emptyset)$ .

Since  $f'_v(P_v^A) = f_v(P_v^A) - f_v(\emptyset)$  for every  $v \in N$  the sum of the local scores is

$$\begin{aligned} \sum_{v \in N} f_v(P_v^A) &= \sum_{n \in N} (f'_v(P_v^A) + f_v(\emptyset)) \\ &\geq t' + \sum_{n \in N} f_v(\emptyset) = t. \end{aligned}$$

To show that  $A$  is nice for  $I$ , let  $f_v(P_v^A) \leq f_v(\emptyset)$ . By the construction of  $\mathcal{F}$ , this implies that  $f'_v(P_v^A) = 0$  and therefore  $f'_v(P_v^A) = f'_v(\emptyset)$ . Since  $A$  is nice for  $I'$  we conclude  $P_v^A = \emptyset$ . Hence,  $A$  is a nice solution for  $I$ .  $\square$

**Potential Parent Sets.** Given an instance  $I := (N, \mathcal{F}, t, k)$  and some  $v \in N$ , we denote the *potential parent sets of  $v$*  by  $\mathcal{P}_{\mathcal{F}}(v) := \{P \subseteq N \setminus \{v\} : f_v(P) > 0\} \cup \{\emptyset\}$ , which are exactly the parent sets stored in  $\mathcal{F}$ . If  $\Pi$  is monotone, we can assume by Proposition 1 that in a solution  $A$  of  $I$ , every vertex  $v$  has a parent set  $P_v^A \in \mathcal{P}_{\mathcal{F}}(v)$ . An important measurement for the running times of our algorithms is the maximum number of potential parent sets  $\delta_{\mathcal{F}}$  which is formally defined by  $\delta_{\mathcal{F}} := \max_{v \in N} |\mathcal{P}_{\mathcal{F}}(v)|$  (Ordyniak & Szeider, 2013). Given a vertex  $v \in N$ , we can iterate over all potential parent sets of  $v$  and the vertices in these sets in  $\mathcal{O}(\delta_{\mathcal{F}} \cdot n)$  time.

Another tool for designing algorithms for BNSL problems is the directed superstructure (Ordyniak & Szeider, 2013). Let  $N$  be a vertex set with local scores  $\mathcal{F}$ . The *directed superstructure* of  $N$  and  $\mathcal{F}$  is the directed graph  $S_{\vec{\mathcal{F}}} = (N, A_{\mathcal{F}})$  with  $A_{\mathcal{F}} = \{(u, v) \mid \exists P \in \mathcal{P}_{\mathcal{F}}(v) : u \in P\}$ . Intuitively, there exists an arc  $(u, v) \in A_{\mathcal{F}}$  if and only if  $u$  is a potential parent of  $v$ . Given  $N$  and  $\mathcal{F}$ , the directed superstructure  $S_{\vec{\mathcal{F}}}$  can be constructed in linear time. Throughout this work we let  $m := |A_{\mathcal{F}}|$  denote the number of arcs in the directed superstructure. Note that  $m < n^2$ .

### 2.3 Parameterized Complexity

In parameterized complexity (Cygan et al., 2015) one measures the running time of algorithms depending on the total input size and a problem parameter. A *parameterized problem* is a language  $L \subseteq \Sigma \times \mathbb{N}_0$  over a finite alphabet  $\Sigma$ . For an instance  $(I, k)$  of  $L$  we call  $k$  the *parameter*. A parameterized problem  $L$  has an *XP-time algorithm* if for every instance  $(I, k)$  it can be decided in  $\mathcal{O}(|I|^{f(k)})$  time for a computable function  $f$  whether  $(I, k) \in L$ . That is, the problem is solvable in polynomial time when the parameter is constant. A parameterized problem  $L$  is called *fixed-parameter tractable (FPT)* if for every instance  $(I, k)$  it can be decided in  $f(k) \cdot |I|^{\mathcal{O}(1)}$  time for a computable function  $f$  whether  $(I, k) \in L$ . A *problem kernelization* for a parameterized problem  $L$  is a polynomial-time preprocessing. Given an instance  $(I, k)$  of  $L$ , a problem kernelization computes an equivalent instance  $(I', k')$  of  $L$  in polynomial time such that  $|I'| + k' \leq g(k)$  for some computable function  $g$ . If  $g$  is a polynomial, then  $L$  admits a *polynomial kernel*. Some fixed-parameter tractable problems are known to not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$ .



A parameterized reduction maps an instance  $(I, k)$  of some parameterized problem  $L$  in FPT time to an equivalent instance  $(I', k')$  of a parameterized problem  $L'$  such that  $k' \leq f(k)$  for some computable funktion  $f$ . If the reduction runs in  $|I|^{\mathcal{O}(1)}$  time and  $f$  is a polynomial, then the parameterized reduction is called *polynomial parameter transformation*. If there exists a parameterized reduction from a  $W[i]$ -hard parameterized problem  $L$  to a parameterized problem  $L'$ , then  $L'$  is also  $W[i]$ -hard for  $i \in \mathbb{N}$ . If a problem is  $W[i]$ -hard, then it is assumed to be *fixed-parameter intractable*. If there exists a polynomial parameter transformation from a parameterized problem  $L$  to a parameterized problem  $L'$  and  $L$  does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$ , then  $L'$  does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$  (Bodlaender, Thomassé, & Yeo, 2011).

### 3. BNSL with Bounded Vertex Cover Number

We first study the task of learning Bayesian network structures with a bounded vertex cover number. In the framework of constrained BNSL problems, these are the problems  $(\Pi_0 + v)$ -SKELETON BNSL and  $(\Pi_0 + v)$ -MORAL BNSL, where  $\Pi_0$  is the class of edgeless graphs. Note that  $\Pi_0$  is monotone. Korhonen and Parviainen (Korhonen & Parviainen, 2015) analyzed the parameterized complexity of  $(\Pi_0 + v)$ -MORAL BNSL for parameter  $k$ . In their work, they provided an XP-time algorithm and proved  $W[1]$ -hardness. We adapt their approach to obtain an XP-time algorithm for  $(\Pi_0 + v)$ -SKELETON BNSL. Furthermore, we show a slightly stronger hardness result for both problems.

#### 3.1 An XP-time Algorithm for Skeletons with Small Vertex Cover

The XP-time algorithm for  $(\Pi_0 + v)$ -SKELETON BNSL follows the basic idea of the XP-time algorithm for  $(\Pi_0 + v)$ -MORAL BNSL (Korhonen & Parviainen, 2015): First, iterate over every possible choice of the vertex cover  $S$  and then split the arc set into two parts which are the arcs between  $S$  and the parents of  $S$  and the arcs between  $S$  and the children of  $S$ . These two arc sets can be learned and combined independently.

However, we would like to point out that our algorithm for learning a network with bounded vertex cover number in the skeleton differs from the moralized version in one technical point. In the moralized graph, every vertex of a vertex cover  $S$  has at most one parent outside  $S$ . For  $(\Pi_0 + v)$ -MORAL BNSL this can be exploited to find the arcs between the vertices of  $S$  and their parents. However, this does not hold for the skeleton: Consider a star where all the arcs are directed towards a center. In this case, the central vertex forms a minimum vertex cover but the vertex has many parents. In the moralized graph, such star becomes a clique and the vertex cover number is large. To overcome this issue, we split the resulting network into three disjoint arc sets: The incoming arcs of vertices of  $S$ , the incoming arcs of parents  $Q$  of vertices of  $S$ , and the incoming arcs of the remaining vertices.

In summary, the intuitive idea behind the algorithm is to find the vertex cover  $S$  and all parent sets of vertices in  $S$  via bruteforce. For each choice, we compute two further arc sets and combine them all to a solution of  $(\Pi_0 + v)$ -SKELETON BNSL. To find the incoming arcs of parents of  $S$ , we adapt a dynamic programming algorithm for VANILLA-BNSL (Ott & Miyano, 2003; Silander & Myllymäki, 2006). With the next two lemmas, we formalize how our solution is built from disjoint arc sets.

**Lemma 3** *Let  $(N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + v)$ -SKELETON BNSL, and let  $S$  and  $Q$  be disjoint subsets of  $N$ . Furthermore, let there be arc sets  $B_1 \subseteq (Q \cup S) \times S$ ,  $B_2 \subseteq S \times Q$ , and  $B_3 \subseteq S \times (N \setminus (S \cup Q))$ . If  $D' := (S \cup Q, B_1 \cup B_2)$  is a DAG where  $S$  is a vertex cover of  $\mathcal{S}(D')$ , then*

- a)  $D := (N, A)$  with  $A := B_1 \cup B_2 \cup B_3$  is a DAG,
- b)  $S$  is a vertex cover of  $\mathcal{S}(D)$ , and
- c)  $\sum_{v \in N} f_v(P_v^A) = \sum_{v \in S} f_v(P_v^{B_1}) + \sum_{v \in Q} f_v(P_v^{B_2}) + \sum_{v \in N \setminus (S \cup Q)} f_v(P_v^{B_3})$ .

PROOF Consider Statement a). Observe that if  $(v, w) \in B_3$ , then  $w$  is a sink in  $D$ . Together with the fact that  $D'$  is a DAG, this implies that  $D$  is a DAG. Moreover, Statement b) holds, since every arc in  $A$  has at least one endpoint in  $S$ . For Statement c), observe that  $S$ ,  $Q$ , and  $(N \setminus (S \cup Q))$  form a partition of  $N$ , and thus, every  $v \in N$  has incoming arcs from either  $B_1$ ,  $B_2$ , or  $B_3$ .  $\square$

**Lemma 4** *Let  $D := (N, A)$  be a DAG such that  $S \subseteq N$  is a vertex cover in  $\mathcal{S}(D)$ . Then, there exists a set  $Q \subseteq N \setminus S$  and arc sets  $B_1 \subseteq (Q \cup S) \times S$ ,  $B_2 \subseteq S \times Q$ , and  $B_3 \subseteq S \times (N \setminus (S \cup Q))$  that form a partition of  $A$ . Moreover, every vertex in  $Q$  has a child in  $S$ .*

PROOF We set  $Q := \{v \in N \setminus S \mid v \text{ has a child in } S\}$ . Then, every vertex in  $Q$  has a child in  $S$  by definition. Furthermore, we set  $B_1 := ((Q \cup S) \times S) \cap A$ ,  $B_2 := (S \times Q) \cap A$ , and  $B_3 := (S \times (N \setminus (S \cup Q))) \cap A$ .

Obviously,  $B_1 \cup B_2 \cup B_3 \subseteq A$ , and the sets are pairwise disjoint, since  $S$ ,  $Q$ , and  $N \setminus (S \cup Q)$  are disjoint subsets of  $N$ . It remains to show that  $B_1 \cup B_2 \cup B_3 \supseteq A$ . To this end, let  $(v, w) \in A$ . If  $w \in S$ , then  $v$  has a child in  $S$ . Consequently,  $v \in S \cup Q$  and therefore,  $(v, w) \in B_1$ . Otherwise, if  $w \notin S$ , then  $v \in S$ , since  $S$  is a vertex cover of  $\mathcal{S}(D)$ . Therefore,  $(v, w) \in B_2 \cup B_3$ .  $\square$

Intuitively, the algorithm works as follows: We iterate over all possible choices of  $S$ ,  $Q$ , and  $B_1$ . Then, for each such choice, we compute  $B_2$  and  $B_3$  that maximize the sum of local scores for  $A := B_1 \cup B_2 \cup B_3$ . In the following, we describe how to compute  $B_2$  when  $S$ ,  $Q$ , and  $B_1$  are given. This step is the main difference between this algorithm and the XP-time algorithm for  $(\Pi_0 + v)$ -MORAL BNSL (Korhonen & Parviainen, 2015).

**Proposition 5** *Let  $I := (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + v)$ -SKELETON BNSL, and let  $S$  and  $Q$  be disjoint subsets of  $N$ . Furthermore, let  $B_1 \subseteq (Q \cup S) \times S$  be an arc set such that  $(Q \cup S, B_1)$  is a DAG and every  $w \in Q$  has a child in  $S$ . Then, we can compute an arc set  $B_2$  that maximizes  $\sum_{v \in Q} f_v(P_v^{B_2})$  among all arc sets where  $(Q \cup S, B_1 \cup B_2)$  is a DAG and  $B_2 \subseteq S \times Q$  in  $2^{|S|} \cdot |I|^{\mathcal{O}(1)}$  time.*

PROOF We describe a dynamic programming algorithm.

*Intuition.* Before we present the algorithm, we provide some intuition. Given a subset  $S' \subseteq S$  and the set  $Q' \subseteq Q$  containing parents of vertices in  $S$ , we want to compute an arc-set  $B \subseteq S' \times Q'$  such that the sum of local scores for the arc-set  $B_1 \cup B$  is maximized. This is done by recursively choosing a vertex  $v \in S'$  that is a sink in the resulting DAG and letting all  $w \in Q'$  whose only child is  $v$  choose their best possible parent set in  $S' \setminus \{v\}$ .

*Algorithm.* To describe the algorithm, we introduce some notation. Given some  $w \in Q$ , we let  $C_w^{B_1}$  denote the set of children of  $w$  in  $(S \cup Q, B_1)$ . Note that  $C_w^{B_1} \subseteq S$  for all  $w \in Q$ . Given a subset  $S' \subseteq S$ , we let  $Q(S') := Q \cap (\bigcup_{v \in S'} P_v^{B_1})$  denote the set of parents of vertices in  $S'$  that belong to  $Q$ , and  $D(S')$  denote the DAG with vertex set  $S' \cup Q(S')$  and arc set  $B_1 \cap ((S' \cup Q(S')) \times S')$ . Furthermore, given  $S' \subseteq S$  and  $v \in S'$ , we let

$$X(S', v) := \{w \in Q(S') \mid C_w^{B_1} \cap S' = \{v\}\}$$

denote the vertices of  $Q(S')$  whose only child in  $S'$  is  $v$ . Finally, given  $S' \subseteq S$  and  $w \in Q$ , we define  $\hat{f}_w(S') := \max_{S'' \subseteq S'} f_w(S'')$  as the best possible score for a parent set of  $w$  containing only vertices from  $S'$ . The values  $\hat{f}_w(S')$  for all  $S' \subseteq S$  and  $w \in Q$  can be computed in overall  $2^{|S|} \cdot |I|^{\mathcal{O}(1)}$  time (Ott & Miyano, 2003).

The dynamic programming table  $T$  has entries of the type  $T[S']$  where  $S' \subseteq S$ . Each entry stores the score of the best possible arc set  $B \subseteq S' \times Q(S')$  such that  $(S \cup Q, B_1 \cup B)$  is a DAG. For one-element sets  $\{v\} \subseteq S$ , we set  $T[\{v\}] := \sum_{w \in P_v^{B_1}} f_w(\emptyset)$ . Note that, due to Proposition 2 we may assume that  $T[\{v\}] = 0$ . The recurrence to compute an entry for  $S'$  with  $|S'| > 1$  is

$$T[S'] := \max_{\substack{v \in S' \\ v \text{ is a sink in } D(S')}} \left( T[S' \setminus \{v\}] + \sum_{w \in X(S', v)} \hat{f}_w(S' \setminus \{v\}) \right).$$

The score of the best possible arc set  $B_2 \subseteq S \times Q$  such that  $(S \cup Q, B_1 \cup B_2)$  is a DAG can be computed by evaluating  $T[S]$ . The corresponding arc set can be found via traceback. The correctness proof is straightforward and thus omitted.

*Running Time.* Recall that all values  $\hat{f}_w(S')$  with  $S' \subseteq S$  and  $w \in Q$  can be computed in  $2^{|S|} \cdot |I|^{\mathcal{O}(1)}$  time. The dynamic programming table has  $2^{|S|}$  entries and each entry can be computed in  $|I|^{\mathcal{O}(1)}$  time. Thus, the overall running time is  $2^{|S|} \cdot |I|^{\mathcal{O}(1)}$  as claimed.  $\square$

We now present the XP-time algorithm for  $(\Pi_0 + v)$ -SKELETON BNSL. This algorithm uses the algorithm behind Proposition 5 as a subroutine.

**Theorem 6**  $(\Pi_0 + v)$ -SKELETON BNSL can be solved in  $(n\delta_{\mathcal{F}})^k \cdot 2^k \cdot |I|^{\mathcal{O}(1)}$  time.

*PROOF Algorithm.* Let  $I := (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + v)$ -SKELETON BNSL. The following algorithm decides whether  $I$  is a yes-instance or a no-instance: First, iterate over every possible choice of a vertex set  $S$  with  $|S| \leq k$  forming the vertex cover of the skeleton of the resulting network. For each choice of  $S$  iterate over every choice of potential parent sets for the vertices of  $S$ . Let  $B_1$  be the corresponding arc set, and let  $Q$  be the set of parents of  $S$  in  $(N, B_1)$ . For each choice of  $S$  and  $B_1$ , do the following:

- Use the algorithm behind Proposition 5 to compute an arc set  $B_2 \subseteq S \times Q$  that maximizes  $\sum_{v \in Q} f_v(P_v^{B_2})$  among all arc sets where  $(Q \cup S, B_1 \cup B_2)$  is a DAG.
- For every  $v \in N \setminus (S \cup Q)$ , compute a potential parent set that maximizes  $f_v(P)$  among all potential parent sets with  $P \subseteq S$ . Let  $B_3 \subseteq S \times (N \setminus (S \cup Q))$  be the resulting arc set.

- If  $\sum_{v \in S} f_v(P_v^{B_1}) + \sum_{v \in Q} f_v(P_v^{B_2}) + \sum_{v \in N \setminus (S \cup Q)} f_v(P_v^{B_3}) \geq t$ , then return *yes*.

If for none of the choices of  $S$  and  $B_1$  the answer *yes* was returned, then return *no*.

*Running Time.* First, we discuss the running time of the algorithm. Since  $|S| \leq k$ , there are  $\mathcal{O}(n^k)$  choices for  $S$  and  $\mathcal{O}(\delta_{\mathcal{F}}^k)$  choices for  $B_1$ . For each such choice, the algorithm behind Proposition 5 can be applied in  $2^k \cdot |I|^{\mathcal{O}(1)}$  time and the choice of the parent sets of vertices in  $N \setminus (S \cup Q)$  can be done in  $|I|^{\mathcal{O}(1)}$  time. This gives an overall running time of  $(n\delta_{\mathcal{F}})^k \cdot 2^k \cdot |I|^{\mathcal{O}(1)}$  as claimed.

*Correctness.* Second, we show that the algorithm returns *yes* if and only if  $I$  is a yes-instance.

( $\Rightarrow$ ) Suppose the algorithm returns *yes*. Then, there exist disjoint subsets  $S$  and  $Q$  of  $N$ , with  $|S| \leq k$  and arc sets  $B_1 \subseteq (S \cup Q) \times S$ ,  $B_2 \subseteq S \times Q$ , and  $B_3 \subseteq S \times N \setminus (S \cup Q)$  such that  $(Q \cup S, B_1 \cup B_2)$  is a DAG. Due to Lemma 3,  $D := (N, A)$  with  $A := B_1 \cup B_2 \cup B_3$  is a DAG and  $S$  is a vertex cover of  $\mathcal{S}(D)$ . Moreover,  $\sum_{v \in N} f_v(P_v^A) \geq t$  and therefore,  $I$  is a yes-instance.

( $\Leftarrow$ ) Let  $I$  be a yes-instance. Then, there exists an  $(N, \mathcal{F}, t)$ -valid arc set  $A$  such that the skeleton of  $D := (N, A)$  has a vertex cover  $S$  of size at most  $k$ . By Lemma 4, there exists a set  $Q \subseteq N \setminus S$  and arc sets  $B_1 \subseteq (S \cup Q) \times S$ ,  $B_2 \subseteq S \times Q$ , and  $B_3 \subseteq S \times N \setminus (S \cup Q)$  that form a partition of  $A$ . Since the algorithm iterates over all choices of  $S$  and  $B_1$  with  $|S| \leq k$ , it considers  $S$  and  $B_1$  at some point. For this choice of  $S$  and  $B_1$  the algorithm then computes an arc set  $B'_2 \subseteq S \times Q$  with

$$\sum_{v \in Q} f_v(P_v^{B'_2}) \geq \sum_{v \in Q} f_v(P_v^{B_2})$$

and an arc set  $B'_3 \subseteq S \times N \setminus (S \cup Q)$  with

$$\sum_{v \in N \setminus (S \cup Q)} f_v(P_v^{B'_3}) \geq \sum_{v \in N \setminus (S \cup Q)} f_v(P_v^{B_3}).$$

Then, since the sum of the local scores under  $A$  is at least  $t$ , the algorithm returns *yes*.  $\square$

### 3.2 W[2]-hardness for Skeletons with Small Vertex Cover

We complement the XP-time algorithm from the previous subsection by proving W[2]-hardness of  $(\Pi_0 + v)$ -SKELETON BNSL. Thus,  $(\Pi_0 + v)$ -SKELETON BNSL is not FPT for parameter  $k$  unless  $W[2] = \text{FPT}$ . We show that the hardness also holds for the task of learning a Bayesian network where the skeleton has a bounded dissociation number. Formally, this is  $(\Pi_1 + v)$ -SKELETON BNSL, with  $\Pi_1 := \{G \mid G \text{ has maximum degree } 1\}$ . Observe that  $\Pi_1$  is monotone.

**Theorem 7** *Let  $\Pi \in \{\Pi_0, \Pi_1\}$ . Then,  $(\Pi + v)$ -SKELETON BNSL is W[2]-hard for  $k$  even when the directed superstructure is a DAG, the maximum parent set size is 1, and every local score is either 1 or 0.*

**PROOF** We give a parameterized reduction from SET COVER. In SET COVER, one is given a universe  $U$ , a family  $\mathcal{X}$  of subsets of  $U$ , and an integer  $\ell$ . The question is, whether there

exists a subfamily  $\mathcal{X}' \subseteq \mathcal{X}$  with  $|\mathcal{X}'| \leq \ell$  that covers  $U$ . That is, every  $u \in U$  is contained in some set of  $\mathcal{X}'$ . SET COVER is W[2]-hard when parameterized by  $\ell$  (Cygan et al., 2015). We first describe a parameterized reduction from SET COVER to  $(\Pi_0 + v)$ -SKELETON BNSL and afterwards, we describe how this construction can be modified to obtain W[2]-hardness for  $(\Pi_1 + v)$ -SKELETON BNSL.

*Construction.* Let  $(U, \mathcal{X}, \ell)$  be an instance of SET COVER. We describe how to construct an equivalent instance  $I := (N, \mathcal{F}, t, k)$  with  $k = \ell$ . First, we set  $N := U \cup \{v_X \mid X \in \mathcal{X}\}$ . Next, we define the local scores  $\mathcal{F}$ . All local scores are either 1 or 0. For every  $u \in U$  we set  $f_u(P) = 1$  if and only if  $P = \{v_X\}$  for some  $X \in \mathcal{X}$  that contains  $u$ . Furthermore, for every  $v \in \{v_X \mid X \in \mathcal{X}\}$ , we set  $f_v(P) = 0$  for every  $P$ . To finish the construction, we set  $k := \ell$  and  $t := |U|$ .

Observe that for every arc  $(u, v)$  of the directed superstructure, we have  $u \in U$  and  $v \in \{v_X \mid X \in \mathcal{X}\}$ . Consequently, the super structure is a DAG. Furthermore, by the construction of  $\mathcal{F}$ , the maximum parent set size is 1.

*Intuition.* Before we show the correctness, we provide some intuition. To obtain a score of  $t = |U|$ , every vertex in  $U$  has to choose one parent vertex. The chosen parent vertices correspond to the subfamily  $\mathcal{X}' \subseteq \mathcal{X}$  that covers  $U$ . The vertex cover constraint on the network ensures that  $\mathcal{X}'$  has size at most  $k$ .

*Correctness.* We show that  $(U, \mathcal{X}, \ell)$  is a yes-instance of SET COVER if and only if  $I$  is a yes-instance of  $(\Pi_0 + v)$ -SKELETON BNSL.

( $\Rightarrow$ ) Let  $\mathcal{X}' \subseteq \mathcal{X}$  be a subfamily of size at most  $k$  that covers  $U$ . Then, for every  $u \in U$ , there exists some set  $X^u \in \mathcal{X}'$  that contains  $u$ . We define  $A := \{(v_{X^u}, u) \mid u \in U\}$  and show that  $A$  is a solution of  $I$ .

Consider the skeleton  $\mathcal{S}(N, A)$ . Each connected component of  $\mathcal{S}(N, A)$  is either an isolated vertex or a star consisting of a central vertex from  $\{v_X \mid X \in \mathcal{X}'\}$  and leaf vertices from  $U$ . Thus,  $(N, A)$  is a DAG and  $\{v_X \mid X \in \mathcal{X}'\}$  is a vertex cover of the skeleton. Thus,  $\mathcal{S}(N, A) \in \Pi_0 + kv$ , since  $|\mathcal{X}'| \leq k$ . Moreover, observe that  $f_u(P_u^A) = 1$  for every  $u \in U$ . Therefore,  $A$  is  $(N, \mathcal{F}, t)$ -valid.

( $\Leftarrow$ ) Conversely, let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that  $\mathcal{S}(N, A)$  has a vertex cover of size at most  $k$ . Then, since  $t = |U|$ , we have  $f_u(P_u^A) = 1$  for every  $u \in U$ . Thus, for every  $u \in U$  we have  $P_u^A = \{v_X\}$  for some  $X \in \mathcal{X}$  containing  $u$ . We define  $\mathcal{X}' := \{X \in \mathcal{X} \mid P_u^A = \{v_X\} \text{ for some } u \in U\}$ .

We first show that  $\mathcal{X}'$  covers  $U$ . Let  $u \in U$ . Then,  $P_u^A = \{v_X\}$  for some  $X$  containing  $u$  and therefore  $X \in \mathcal{X}'$ . Thus,  $\mathcal{X}'$  covers  $U$ . It remains to show that  $|\mathcal{X}'| \leq k$ . Assume towards a contradiction that  $|\mathcal{X}'| > k$ . Then, there exist pairwise distinct vertices  $u_1, \dots, u_{k+1}$  in  $U$  and  $v_1, \dots, v_{k+1}$  in  $\{v_X \mid X \in \mathcal{X}'\}$  such that  $(v_i, u_i) \in A$  for  $i \in \{1, \dots, k+1\}$ . Then, the edges  $\{(v_i, u_i)\}$  form a matching of size  $k+1$  in  $\mathcal{S}(N, A)$ . This contradicts the fact that  $\mathcal{S}(N, A)$  has a vertex cover of size at most  $k$ .

*BNSL with bounded Dissociation Number.* We now explain how to modify the construction described above, to obtain W[2]-hardness for  $(\Pi_1 + v)$ -SKELETON BNSL when parameterized by  $k$ .

In the construction, we set  $N := U \cup \{v_X \mid X \in \mathcal{X}\} \cup \{w_X \mid X \in \mathcal{X}\}$ . As in the construction described above, for  $u \in U$  we set  $f_u(P) := 1$  if and only if  $P = \{v_X\}$  for some  $X \in \mathcal{X}$  containing  $u$ , and for  $v \in \{v_X \mid X \in \mathcal{X}\}$  we set  $f_v(P) := 0$  for every  $P$ .

Additionally, for every  $w_X$ , we set  $f_{w_X}(P) := 1$  if and only if  $P = \{v_X\}$ . Furthermore, we set  $k := \ell$  and  $t := |U| + |\mathcal{X}|$

( $\Rightarrow$ ) Let  $\mathcal{X}' \subseteq \mathcal{X}$  be a subfamily with  $|\mathcal{X}'| \leq k$  that covers  $U$ . We set  $A := \{(v_{X^u}, u) \mid u \in U\} \cup \{(v_X, w_X) \mid X \in \mathcal{X}'\}$ . Then,  $(N, A)$  is a DAG and the sum of the local scores is  $t$ . Furthermore, the connected components of  $\mathcal{S}(N, A)$  are isolated edges or disjoint stars with central vertex in  $\{v_X \mid X \in \mathcal{X}'\}$ . Then,  $|\mathcal{X}'| \leq k$  implies  $\mathcal{S}(N, A) \in \Pi_1 + kv$ .

( $\Leftarrow$ ) Let  $A$  be a solution of  $I$ . Again, we define  $\mathcal{X}' := \{X \in \mathcal{X} \mid P_u^A = \{v_X\} \text{ for some } u \in U\}$ , which covers  $U$  by the same arguments as above. Note that the skeleton of  $(N, A)$  contains an edge  $\{v_X, w_X\}$  for every  $X \in \mathcal{X}$ , since the sum of local scores under  $A$  is at least  $t$ . Then, assuming  $|\mathcal{X}'| > k$  implies that there exist  $k+1$  vertex disjoint sets  $\{u, v_X, w_X\}$  where  $v_X$  is adjacent with  $u$  and  $w_X$  in  $\mathcal{S}(N, A)$ . This contradicts the fact that  $\mathcal{S}(N, A)$  has a dissociation set of size at most  $k$ .  $\square$

Observe that for a DAG  $D := (N, A)$  where each vertex has at most one parent, the skeleton  $\mathcal{S}(D)$  and the moralized graph  $\mathcal{M}(D)$  are the same. Thus, Theorem 7 also implies  $W[2]$ -hardness if the sparsity constraints are posed on the moralized graph. Note that a  $W[1]$ -hardness for  $(\Pi_0 + v)$ -MORAL BNSL when parameterized by  $k$  has been shown (Korhonen & Parviainen, 2015). We now obtain a slightly stronger hardness result with an additional restriction on the maximum parent set size.

**Corollary 8** *Let  $\Pi \in \{\Pi_0, \Pi_1\}$ . Then,  $(\Pi + v)$ -MORAL BNSL is  $W[2]$ -hard for  $k$  even when the directed superstructure is a DAG, the maximum parent set size is 1, and every local score is either 1 or 0.*

By Corollary 8 it is presumably not possible that  $(\Pi_1 + v)$ -MORAL BNSL is FPT for  $k$ .

Consider networks where the maximum parent set size is 1. These networks are also known as *branchings*. Learning a branching without further restrictions can be done in polynomial time (Chow & Liu, 1968; Gaspers et al., 2015). Due to Theorem 7 and Corollary 8, there is presumably no such polynomial-time algorithm if we add a sparsity constraint on the vertex cover size. Thus, the task to learn a branching is an example where it is harder to learn a more restricted network.

#### 4. BNSL with Bounded Dissociation Number

In this section we provide an algorithm for  $(\Pi_1 + v)$ -MORAL BNSL, that is, for Bayesian network learning where the moralized graph has dissociation number at most  $k$ . By the results above, an FPT algorithm for  $k$  is unlikely. We show that it can be solved in XP-time when parameterized by  $k$ . As detailed in the introduction, this shows that we can find optimal networks for a class of moral graphs that is larger than the ones with bounded vertex cover number, while maintaining the highly desirable property that the treewidth is bounded. In fact, graphs with dissociation number at most  $k$  have treewidth at most  $k + 1$  and thus the inference task can be preformed efficiently if  $k$  is small (Darwiche, 2009).

Before we describe the main idea of the algorithm, we provide the following simple observation about Bayesian networks whose moralized graph has a bounded dissociation number.

**Proposition 9** *Let  $D = (N, A)$  be a DAG and  $S \subseteq N$  be a dissociation set of  $\mathcal{M}(D)$ . Then, at most  $2|S|$  vertices in  $N \setminus S$  have descendants in  $S$ .*

PROOF Let  $v \in S$ . We call a vertex  $w \in N \setminus S$  is an *external ancestor of  $v$*  if there exists a path  $(w, w_1, \dots, w_\ell, v)$  in  $D$  such that  $w_i \in N \setminus S$  for all  $i \in \{1, \dots, \ell\}$ . We show that every vertex in  $S$  has at most two external ancestors.

First, assume that  $v$  has three distinct parents  $w_1, w_2$ , and  $w_3$  outside  $S$ . Then, there are moral edges  $\{w_1, w_2\}$ ,  $\{w_2, w_3\}$ , and  $\{w_3, w_1\}$  forming a triangle outside  $S$  in  $\mathcal{M}(D)$ . This contradicts the fact that  $S$  is a dissociation set of  $\mathcal{M}(D)$ . Hence, every  $v \in S$  has at most two parents outside  $S$ . Next, consider the following cases.

**Case 1:**  $|P_v^A \setminus S| = 0$ . Then,  $v$  has no external ancestors and nothing more needs to be shown.

**Case 2:**  $|P_v^A \setminus S| = 1$ . Then, let  $P_v^A \setminus S = \{w\}$ . Since  $S$  is a dissociation set of  $\mathcal{M}(D)$  it holds that  $\deg_{\mathcal{M}(D)-S}(w) \leq 1$ . Hence,  $w$  has at most one parent  $w'$  outside  $S$ . Moreover, since  $\deg_{\mathcal{M}(D)-S}(w') \leq 1$ , the vertex  $w'$  has no parent in  $N \setminus S$ . Therefore,  $v$  has at most two external ancestors.

**Case 3:**  $|P_v^A \setminus S| = 2$ . Then, let  $P_v^A \setminus S = \{w_1, w_2\}$ . Note that  $\{w_1, w_2\}$  is a moral edge in  $\mathcal{M}(D)$ . Then, since  $\deg_{\mathcal{M}(D)-S}(w_1) \leq 1$  and  $\deg_{\mathcal{M}(D)-S}(w_2) \leq 1$ , the vertices  $w_1$  and  $w_2$  do not have parents in  $N \setminus S$ . Therefore,  $v$  has exactly two external ancestors.  $\square$

The main idea of the algorithm for  $(\Pi_1 + v)$ -MORAL BNSL presented in this work is closely related to XP-algorithms for  $(\Pi_0 + v)$ -MORAL BNSL (Korhonen & Parviainen, 2015) and  $(\Pi_0 + v)$ -SKELETON BNSL (Theorem 6): If we know which vertices form the dissociation set  $S$  and the set  $Q$  of vertices that are the ancestors of  $S$ , the arcs of the network can be found efficiently. Roughly speaking, the steps of the algorithm are to iterate over every possible choice of  $S$  and  $Q$  and then find the arc set of the resulting network respecting this choice. Finding the arc set can then be done in two steps: First, we find all the arcs between the vertices of  $S \cup Q$  and afterwards, we find the remaining arcs of the network. Even though the basic idea of the algorithm is similar to algorithms for BNSL with bounded vertex cover number, several obstacles occur when considering  $\Pi_1$  instead of  $\Pi_0$ .

First, the arcs between  $S \cup Q$  and the remaining arcs of the DAG cannot be computed independently, since there might be arcs between vertices of  $Q$  and  $N \setminus (Q \cup S)$ . See Figure 1 for an example of a DAG  $D$  whose moralized graph has a dissociation set  $S$ . We overcome this obstacle by partitioning  $Q$  into two sets  $Q_0$  and  $Q_1$  and by considering arc sets  $A_Q \subseteq (S \cup Q) \times (S \cup Q)$  that respect a specific constraint regarding this partition.

Second, the vertices in  $N \setminus (S \cup Q)$  cannot choose their parent sets greedily from  $S$ , since they may also choose one parent from  $N \setminus S$ . Thus, we need a new technique to find this part of the network. To overcome this obstacle, we define the problem BASEMENT LEARNING and show that it can be solved in polynomial time.

This section is organized as follows: In Section 4.1, we introduce the terms of *attic arc sets* and *basement arc sets* which form the parts of the arc set that we later combine to a solution. In Sections 4.2, we describe how to find the attic arc set and in Section 4.3, we describe how to find the basement arc set. Finally, in Section 4.4, we combine the previous results and describe how to solve  $(\Pi_1 + v)$ -MORAL BNSL in XP-time.

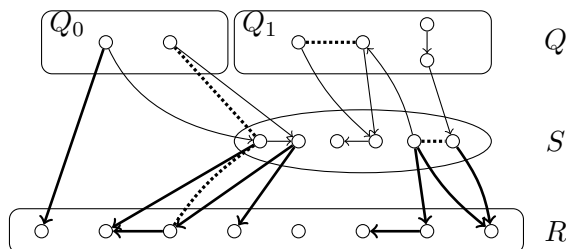


Figure 1: A DAG  $D$  whose moralized graph has a dissociation set  $S$ . The arc set of  $D$  is decomposed into an attic arc set  $A_Q$  and a basement arc set  $A_R$ . The thin arrows correspond to the arcs of  $A_Q$  and the thick arrows correspond to the arcs of  $A_R$ . The dotted edges are the moral edges.

We end this section by showing another hardness result for  $(\Pi_1 + v)$ -MORAL BNSL. Note that due to Corollary 8 it is unlikely that  $(\Pi_1 + v)$ -MORAL BNSL is FPT for  $k$ . In Section 4.5 we show that even for parameterization by  $k + t + \delta_{\mathcal{F}} + p$ , where  $p$  denotes the maximum parent set size, it is unlikely to obtain an FPT algorithm for  $(\Pi_1 + v)$ -MORAL BNSL.

#### 4.1 Attic Arc Sets and Basement Arc Sets

In this subsection we formally define *attic arc sets* and *basement arc sets*. As mentioned above, these are the two parts of the resulting network that our algorithm finds when the vertices of the dissociation set and their ancestor vertices are known. The intuitive idea behind the names *attic arc set* and *basement arc set* is that the dissociation set  $S$  forms the center of the network, the arcs between  $S$  and the ancestors of  $S$  form the upper part of the network (attic) and the remaining arcs from the lower part (basement) of the network. Figure 1 shows a DAG  $D$  where the arcs are decomposed into an attic arc set and a basement arc set.

Throughout this section, we let  $S$  denote the set of the vertices that form the dissociation set and we let  $Q$  denote the set of their ancestors outside  $S$ . Furthermore, we assume that  $Q$  is partitioned into two sets  $Q_0$  and  $Q_1$ . Intuitively, in the moralized graph of the resulting network, the vertices in  $Q_0$  have no neighbors in  $Q$  and the vertices in  $Q_1$  may have one neighbor in  $Q$ .

**Definition 10** *Let  $N$  be a vertex set and let  $S$ ,  $Q_0$ , and  $Q_1$  be disjoint subsets of  $N$ . An arc set  $A_Q$  is called attic arc set of  $S$ ,  $Q_0$ , and  $Q_1$ , if*

- a)  $D_Q := (S \cup Q_0 \cup Q_1, A_Q)$  is a DAG,
- b) in the moralized graph  $\mathcal{M}(D_Q)$ , no vertex of  $Q_0$  has neighbors outside  $S$ , and every vertex of  $Q_1$  has at most one neighbor outside  $S$ .

If  $S$ ,  $Q_0$ , and  $Q_1$  are clear from the context we may refer to  $A_Q$  as *attic arc set*. Throughout this section we use the following notation as a shorthand for some vertex sets: Given a



vertex set  $N$  and disjoint subsets  $S$ ,  $Q_0$ , and  $Q_1$  of  $N$ , we let

$$\begin{aligned} Q &:= Q_0 \cup Q_1, \text{ and we let} \\ R &:= N \setminus (S \cup Q) \text{ denote the remaining vertices of } N. \end{aligned}$$

We next define basement arc sets.

**Definition 11** *Let  $N$  be a vertex set and let  $S$ ,  $Q_0$ , and  $Q_1$  be disjoint subsets of  $N$ . An arc set  $A_R \subseteq (S \cup Q_0 \cup R) \times R$  is called basement arc set for  $S$ ,  $Q_0$ , and  $Q_1$  if  $A_R$  contains no self-loops and every  $w \in Q_0 \cup R$  has at most one incident arc in  $A_R \cap ((R \cup Q_0) \times R)$ .*

If  $S$ ,  $Q_0$ , and  $Q_1$  are clear from the context we may refer to  $A_R$  as *basement arc set*. The idea is that basement arc sets and attic arc sets can be combined to a solution of  $(\Pi_1 + v)$ -MORAL BNSL and that a solution can be splitted into an attic arc set and a basement arc set. With the next two lemmas, we formalize this idea. First, an attic arc set and a basement arc set can be combined to a DAG where  $S$  is a dissociation set of the moralized graph.

**Lemma 12** *Let  $N$  be a vertex set and let  $S$ ,  $Q_0$  and  $Q_1$  be disjoint subsets of  $N$ . Furthermore, let  $A_Q$  be an attic arc set, and let  $A_R$  be a basement arc set. It then holds that*

1.  $D := (N, A_Q \cup A_R)$  is a DAG, and
2.  $S$  is a dissociation set of  $\mathcal{M}(D)$ .

PROOF We first show that  $D$  is a DAG. Assume towards a contradiction that there is a directed cycle in  $D$ . Since  $A_Q$  is an attic arc set we conclude from Definition 10 a) that there is no directed cycle in  $(N, A_Q)$ . Hence, the cycle contains an edge  $(v, w) \in A_R$ . Note that  $w \in R$  and there exists an outgoing edge  $(w, w') \in A_Q \cup A_R$  that is also part of the cycle. Since no edge in  $A_Q$  is incident with vertices of  $R$  we conclude  $(w, w') \in A_R$  and therefore  $w' \in R$ . Note that  $w' \neq w$ , since  $A_R$  contains no self-loops. Since  $(w, w')$  is part of the directed cycle, there exists an edge  $(w', w'') \in A_R$  with  $w'' \in R$ . Then,  $w'$  is incident with two arcs in  $A_R \cap ((R \cup Q_0) \times R)$  which is a contradiction to the fact that  $A_R$  is a basement arc set. Consequently, there is no directed cycle in  $D$ .

It remains to show that  $S$  is a dissociation set of  $\mathcal{M}(D)$ . That is, we show that every vertex  $v$  has degree at most 1 in  $G := \mathcal{M}(D) - S$ .

If  $v \in Q_1$ , then  $v$  has degree at most one in  $\mathcal{M}(D_Q) - S$ . Since no arc in  $A_R$  is incident with  $v$ , we conclude  $\deg_G(v) = 1$ .

Otherwise,  $v \in Q_0 \cup R$ . Then, there is no arc in  $A_Q$  connecting  $v$  with a vertex in  $N \setminus S$ . Moreover, by Definition 11, there is at most one arc in  $A_R \cap ((R \cup Q_0) \times R)$  that is incident with  $v$ . To prove  $\deg_G(v) \leq 1$  it remains to show that there is no moral edge of  $\mathcal{M}(D)$  connecting  $v$  with some other vertex in  $N \setminus S$ . Assume towards a contradiction that there exists some  $v' \in N \setminus S$  such that  $v$  and  $v'$  have a common child  $w$ .

If  $w \in S \cup Q$ , then  $v \in Q_0$  and  $v' \in Q$ . Consequently,  $\{v, v'\}$  is a moral edge in  $\mathcal{M}(D_Q)$  which contradicts the fact that vertices in  $Q_0$  have degree 0 in  $\mathcal{M}(D_Q) - S$ . Hence, we conclude  $w \in R$  and therefore  $(v, w), (v', w) \in A_R$ . Then,  $w$  has two incident arcs in  $A_R \cap ((R \cup Q_0) \times R)$  which contradicts the fact that  $A_R$  is a basement arc set. Hence,  $\deg_G(v) \leq 1$ .

□

Next, we show that conversely, the arc set of every DAG whose moralized graph has a dissociation set  $S$  can be partitioned into the an attic arc set and a basement arc set for some  $Q_0$  and  $Q_1$ .

**Lemma 13** *Let  $D = (N, A)$  be a DAG and let  $S \subseteq N$  be a dissociation set of  $\mathcal{M}(D)$ . Furthermore, let  $Q := \{v \in N \setminus S \mid v \text{ has at least one descendant in } S\}$  and let  $Q$  be partitioned into*

$$Q_0 := \{v \in Q \mid v \text{ has degree } 0 \text{ in } \mathcal{M}(S \cup Q, A_Q) - S\}, \text{ and}$$

$$Q_1 := Q \setminus Q_0.$$

*Then,  $A_Q := ((S \cup Q) \times (S \cup Q)) \cap A$  is an attic arc set and  $A \setminus A_Q$  is a basement arc set. Moreover,  $|Q| \leq 2|S|$ .*

PROOF Note that Proposition 9 implies  $|Q| \leq 2|S|$ . We first show that Properties a) and b) from Definition 10 hold for  $A_Q$ . Since  $D = (N, A)$  is a DAG,  $S \cup Q \subseteq N$ , and  $A_Q \subseteq A$ , it clearly holds that  $D_Q$  is a DAG and therefore Property a) holds. Consider Property b). By the definition of  $Q_0$ , no vertex in  $Q_0$  has neighbors in  $Q$ . Furthermore, since  $S$  is a dissociation set of  $\mathcal{M}(D)$ , every vertex in  $Q_1$  has at most one neighbor in  $Q$ .

It remains to show that  $A \setminus A_Q$  is a basement arc set. To this end, we first show  $A \setminus A_Q \subseteq (S \cup Q_0 \cup R) \times R$ . Assume towards a contradiction that  $A \setminus A_Q \not\subseteq (S \cup Q_0 \cup R) \times R$ . Consider the following cases.

**Case 1:** There exists an arc  $(v, w) \in A \setminus A_Q$  with  $w \notin R$ . Then,  $w \in S \cup Q$  and therefore,  $v$  is an ancestor of  $S$ . Hence,  $(v, w) \in A_Q$  which contradicts the choice of  $(v, w)$ .

**Case 2:** There exists an arc  $(v, w) \in A \setminus A_Q$  with  $v \in Q_1$ . From the previous case we know  $w \in R$ . Since  $v$  has degree 1 in  $\mathcal{M}(N, A_Q) - S$  and an incident arc to some vertex in  $R$  we conclude  $\deg_{\mathcal{M}(D)-S}(v) \geq 2$  which contradicts the fact that  $S$  is a dissociation set of  $\mathcal{M}(D)$ . Since Cases 1 and 2 are contradictory, we have  $A \setminus A_Q \subseteq (S \cup Q_0 \cup R) \times R$ .

Finally, we show that Definition 11 holds for  $A \setminus A_Q$ . Since  $D$  is a DAG we conclude that  $A \setminus A_Q$  contains no self-loops. Moreover, since  $S$  is a dissociation set of  $\mathcal{M}(D)$  we conclude that every  $w \in (Q_0 \cup R)$  has at most one incident edge in  $(A \setminus A_Q) \cap ((R \cup Q_0) \times R)$ .  $\square$

In general, if we consider a union  $A_1 \cup A_2$  of two disjoint arc-sets, one vertex  $v$  may have incoming arcs from  $A_1$  and  $A_2$ . Thus, for the local scores we may have  $f_v(P_v^{A_1}) \neq f_v(P_v^{A_1 \cup A_2})$ . Given an attic arc set  $A_Q$  and a basement arc set  $A_R$ , all arcs in  $A_R$  have endpoints in  $R$  and all arcs in  $A_Q$  have endpoints in  $Q \cup S$ . Since  $Q \cup S$  and  $R$  are disjoint, for every vertex  $v$  either all incoming arcs are in  $A_Q$  or in  $A_R$ . Thus, the local scores under  $A_Q \cup A_R$  can be decomposed as follows.

**Lemma 14** *Let  $(N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_1 + v)$ -MORAL BNSL and let  $S, Q_0$ , and  $Q_1$  be disjoint subsets of  $N$ . Furthermore, let  $A_Q$  be an attic arc set and let  $A_R$  be a basement arc set. Then, score of  $A := A_Q \cup A_R$  under  $\mathcal{F}$  is*

$$\sum_{v \in N} f_v(P_v^A) = \sum_{v \in S \cup Q} f_v(P_v^{A_Q}) + \sum_{v \in R} f_v(P_v^{A_R}).$$

## 4.2 Finding the Attic Arc Set

Recall that the intuitive idea of the XP-time algorithm is to iterate over all possible vertices that may form the dissociation set and their possible ancestors. Then, for each choice we find an attic arc set and a basement arc set. In this subsection, we present an algorithm to efficiently compute the attic arc set when  $S$ ,  $Q_0$ , and  $Q_1$  are given.

Let  $I := (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_1 + v)$ -MORAL BNSL and let  $S$ ,  $Q_0$ , and  $Q_1$  be disjoint subsets of  $N$ . An attic arc set  $A_Q$  is called *optimal*, if  $\sum_{v \in S \cup Q_0 \cup Q_1} f_v(P_v^{A_Q})$  is maximal among all attic arc sets for  $S$ ,  $Q_0$ , and  $Q_1$ .

Let  $\lambda := |S \cup Q_0 \cup Q_1|$ . Observe that, by iterating over every possible set of arcs between the vertices in  $S \cup Q_0 \cup Q_1$ , one can enumerate all possible  $A_Q$  in  $2^{\mathcal{O}(\lambda^2)} \cdot |I|^{\mathcal{O}(1)}$  time. Alternatively, by iterating over all possible parent sets of the vertices of  $S \cup Q_0 \cup Q_1$ , one can enumerate all possible  $A_Q$  in  $\delta_{\mathcal{F}}^\lambda \cdot |I|^{\mathcal{O}(1)}$  time. However, this might be expensive, since  $\delta_{\mathcal{F}}$  can be exponentially large in the number of vertices. We show that an optimal attic arc set can be computed in  $\lambda^{\mathcal{O}(\lambda)} \cdot |I|^{\mathcal{O}(1)}$  time. The intuitive idea of this algorithm is to find the connected vertex pairs in  $Q_1$  via brute force and use an algorithm for VANILLA-BNSL as a subroutine to find the arcs of  $A_Q$ .

**Proposition 15** *Let  $I := (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_1 + v)$ -MORAL BNSL, and let  $S$ ,  $Q_0$ , and  $Q_1$  be disjoint subsets of  $N$ . An optimal attic arc set for  $S$ ,  $Q_0$ , and  $Q_1$  can be computed in  $\lambda^{\mathcal{O}(\lambda)} \cdot |I|^{\mathcal{O}(1)}$  time, where  $\lambda := |S \cup Q_0 \cup Q_1|$ .*

**PROOF** Throughout this proof, let  $Q := Q_0 \cup Q_1$  and  $N' := S \cup Q$ . Consider  $Q_1$ . An *auxiliary graph*  $H$  is defined as an undirected graph with vertex set  $Q_1$ , such that each connected component of  $H$  has size at most 2. Note that there are  $\binom{\lambda^2}{\lambda} \in \lambda^{\mathcal{O}(\lambda)}$  many auxiliary graphs, since  $|Q_1| \leq \lambda$ .

Let  $H$  be a fixed auxiliary graph. For two vertices  $w_1 \in Q_1$  and  $w_2 \in Q_1$  we write  $w_1 \sim_H w_2$  if they belong to the same connected component of  $H$ . In the following, we define a family  $\mathcal{F}^H$  of local scores for  $N'$ . To this end, we introduce the term of *feasible parent sets regarding  $H$* : First, let  $v \in Q_0$ . A set  $P \subseteq N' \setminus \{v\}$  is called *feasible for  $v$*  if  $P \subseteq S$ . Second, let  $v \in Q_1$ . A set  $P \subseteq N' \setminus \{v\}$  is *feasible for  $v$*  if  $P \cap Q \subseteq \{w\}$  where  $w \sim_H v$ . Finally, let  $v \in S$ . A set  $P \subseteq N' \setminus \{v\}$  is *feasible for  $v$* , if  $|P \cap Q| \leq 1$ , or  $P \cap Q = \{w_1, w_2\}$  for some  $w_1, w_2 \in Q_1$  with  $w_1 \sim_H w_2$ . We then define  $\mathcal{F}^H$  by

$$f_v^H(P) := \begin{cases} f_v(P) & \text{if } P \text{ is feasible for } v, \text{ or} \\ 0 & \text{otherwise.} \end{cases}$$

Note that for every vertex  $v \in N'$ , every potential parent set  $P \in \mathcal{P}_{\mathcal{F}^H}(v)$  is feasible for  $v$  by the definition of  $\mathcal{F}^H$ .

*Algorithm.* The algorithm to compute an optimal arc set for  $S$ ,  $Q_0$ , and  $Q_1$  can be described as follows: Iterate over all auxiliary graphs  $H$ . For every choice of  $H$  compute an arc set  $A_H \subseteq N' \times N'$  that maximizes  $\sum_{v \in N'} f_v^H(P_v^{A_H})$ . Return an arc set  $A \in \{A_H \mid H \text{ is an auxiliary graph}\}$  that maximizes  $\sum_{v \in N'} f_v(P_v^A)$ .

*Running time.* We first consider the running time of the algorithm. As mentioned above, we can iterate over all auxiliary graphs in  $\lambda^{\mathcal{O}(\lambda)}$  time. For every auxiliary graph, we compute the arc set  $A_H$ . This can be done by solving VANILLA-BNSL for the vertex set  $N'$

and local scores  $\mathcal{F}^H$ . This can be done in  $2^\lambda \cdot |I|^{\mathcal{O}(1)}$  time (Ott & Miyano, 2003; Silander & Myllymäki, 2006). Thus, the overall running time of the algorithm is  $\lambda^{\mathcal{O}(\lambda)} \cdot |I|^{\mathcal{O}(1)}$ .

*Correctness.* It remains to show that the algorithm is correct. That is, the returned arc set  $A$  is an optimal attic arc set for  $S$ ,  $Q_0$ , and  $Q_1$ . Note that  $A = A_H$  for some auxiliary graph  $H$ . Therefore,  $A$  is a solution of an instance of VANILLA-BNSL with vertex set  $N'$  and local scores  $\mathcal{F}^H$ . By Proposition 1, we may assume that  $A$  is nice and therefore, for every  $v \in N'$  the parent set  $P_v^A$  is feasible for  $v$  regarding  $H$ . Consequently,  $f_v^H(P_v^A) = f_v(P_v^A)$  for every  $v \in N$ .

We first show that  $A$  is an attic arc set for  $S$ ,  $Q_0$ , and  $Q_1$ . That is, we show that Properties a) and b) from Definition 10 hold. Since  $A$  is a solution of a VANILLA-BNSL instance with vertex set  $N'$ , the graph  $(N', A)$  is a DAG. Thus, Property a) from Definition 10 holds. We next check Property b). First, consider  $v \in Q_0$  and assume towards a contradiction  $v$  has a neighbor  $w \notin S$  in  $\mathcal{M}(N', A)$ . If  $(v, w) \in A$  or  $(w, v) \in A$ , either  $v$  or  $w$  has a non-feasible parent set regarding  $H$ . A contradiction. Otherwise, if  $\{v, w\}$  is a moral edge, then there exists a vertex  $u \in N'$  with  $\{v, w\} \in P_u^A$ . Then,  $P_u^A$  is not feasible for  $u$  regarding  $H$ . A contradiction. Second, consider  $v \in Q_1$ . Then, due to the definition of feasible parent sets, outside  $S$ ,  $v$  can only be adjacent to a vertex  $w \in Q \setminus \{v\}$  if  $v \sim_H w$ . Since the connected components in  $H$  have size at most 2,  $v$  has at most one neighbor outside  $S$  in  $\mathcal{M}(N', A)$ . Therefore, Property b) from Definition 10 holds. Thus,  $A$  is an attic arc set.

We next show that  $A$  is optimal. That is, we show that  $\sum_{S \cup Q} f_v(P_v^A)$  is maximal among all attic arc sets for  $Q_0$ ,  $Q_1$ , and  $S$ . To this end, let  $A' \neq A$  be another attic arc set. Consider  $\mathcal{M}(N', A')$ . Since every vertex in  $Q_1$  has at most one neighbor outside  $S$  in  $\mathcal{M}(N', A')$ , the graph  $H' := \mathcal{M}(N', A')[Q_1]$  has connected components of size at most 2. Consequently,  $H'$  is an auxiliary graph. To show that  $\sum_{S \cup Q} f_v(P_v^{A'}) \leq \sum_{S \cup Q} f_v(P_v^A)$  we use the following claim.

**Claim 1** *For every  $n \in N'$ , the parent set  $P_v^{A'}$  is feasible for  $v$  regarding the auxiliary graph  $H'$ .*

PROOF We consider the following cases.

**Case 1:**  $v \in Q_0$ . Then,  $v$  has no neighbors outside  $S$  in  $\mathcal{M}(N', A')$ . Thus,  $v$  has only incoming arcs from  $S$ . Therefore,  $P_v^{A'}$  is feasible for  $v$ .

**Case 2:**  $v \in Q_1$ . Then,  $v$  has at most one neighbor  $w$  outside  $S$  in  $\mathcal{M}(N', A')$ . Observe that  $w \sim_{H'} v$  by the definition of  $H'$ . Therefore,  $P_v^{A'}$  is feasible for  $v$ .

**Case 3:**  $v \in S$ . Then, if  $|P_v^{A'} \cap Q| \leq 1$ ,  $P_v^{A'}$  is feasible for  $v$ . Furthermore, if  $|P_v^{A'} \cap Q| \geq 3$ , the vertices in  $P_v^{A'} \cap Q$  have degree at least 2 outside  $S$  in  $\mathcal{M}(N', A')$  contradicting the fact that  $A'$  is an attic arc set. Thus, it remains to consider the case where  $|P_v^{A'} \cap Q| = 2$ . Let  $P_v^{A'} \cap Q = \{w_1, w_2\}$ . Then,  $w_1$  and  $w_2$  are connected by a moral edge in  $\mathcal{M}(N', A')$  implying  $w_1 \sim_{H'} w_2$ . Thus,  $P_v^{A'}$  is feasible for  $v$ .  $\diamond$

Since every  $v \in N'$  has a feasible parent set under  $A'$  regarding  $H'$ , we have  $f_v^{H'}(P_v^{A'}) = f_v(P_v^{A'})$ . Since the score of  $A$  under  $\mathcal{F}^H$  is at least as big as the score of the best possible DAG under  $\mathcal{F}^{H'}$ , we conclude

$$\sum_{S \cup Q} f_v(P_v^{A'}) = \sum_{S \cup Q} f_v^{H'}(P_v^{A'}) \leq \sum_{S \cup Q} f_v^H(P_v^A) = \sum_{S \cup Q} f_v(P_v^A).$$

□

### 4.3 Finding the Basement Arc Set

We now show that we can compute a basement with maximal score in polynomial time if  $S$ ,  $Q_0$ , and  $Q_1$  are given. More precisely, we solve the following problem.

**BASEMENT LEARNING**

**Input:** A set of vertices  $N$ , disjoint subsets  $S$ ,  $Q_0$ ,  $Q_1$  of  $N$ , local scores  $\mathcal{F} = \{f_v \mid v \in N\}$ , and an integer  $t$ .

**Question:** Is there a basement arc set  $A_R$  for  $S$ ,  $Q_0$ , and  $Q_1$  with  $\sum_{v \in N \setminus (S \cup Q_0 \cup Q_1)} f_v(P_v^{A_R}) \geq t$ ?

**Proposition 16** **BASEMENT LEARNING** *can be solved in  $\mathcal{O}(n^3 \delta_{\mathcal{F}})$  time.*

**PROOF** We give a polynomial-time reduction to **MAXIMUM WEIGHT MATCHING**. In **MAXIMUM WEIGHT MATCHING** one is given a graph  $G = (V, E)$ , edge-weights  $\omega : E \rightarrow \mathbb{N}$ , and  $\ell \in \mathbb{N}$  and the question is if there exists a set  $M \subseteq E$  of pairwise non-incident edges such that  $\sum_{e \in M} \omega(e) \geq \ell$ .

*Construction:* Let  $I := (N, S, Q_0, Q_1, \mathcal{F}, t)$  be an instance of **BASEMENT LEARNING**. Throughout this proof let  $Q := Q_0 \cup Q_1$ , and let  $R := N \setminus (S \cup Q)$ . We construct an equivalent instance  $(G, \omega, \ell)$  of **MAXIMUM WEIGHT MATCHING**. We first define  $G := (V, E)$  with  $V := Q_0 \cup R \cup R'$ , where  $R' := \{v' \mid v \in R\}$ , and  $E := X \cup Y \cup Z$ , where

$$\begin{aligned} X &:= \{\{v, w\} \mid v, w \in R, v \neq w\}, \\ Y &:= \{\{v, w\} \mid v \in R, w \in Q_0\}, \text{ and} \\ Z &:= \{\{v, v'\} \mid v \in R\}. \end{aligned}$$

Next, we define edge-weights  $\omega : E \rightarrow \mathbb{N}$ : For  $e = \{v, v'\} \in Z$ , we set

$$\omega(e) := \max_{S' \subseteq S} f_v(S').$$

Furthermore, for  $e = \{v, w\} \in Y$  with  $v \in R$  and  $w \in Q_0$ , we set

$$\omega(e) := \max_{S' \subseteq S} f_v(S' \cup \{w\}).$$

Finally, for  $e = \{v, w\} \in X$ , we set  $\omega(e) := \max(\varphi(v, w), \varphi(w, v))$ , where

$$\varphi(u_1, u_2) := \max_{S' \subseteq S} f_{u_1}(S' \cup \{u_2\}) + \max_{S' \subseteq S} f_{u_2}(S').$$

To complete the construction of  $(G, \omega, \ell)$ , we set  $\ell := t$ .

*Intuition:* Before we prove the correctness of the reduction we provide some intuition. A maximum-weight matching  $M$  in  $G$  corresponds to the parent sets of vertices in  $R$  and therefore to arcs in a solution  $A_R$  of  $I$ . More precisely, an edge  $\{v, v'\} \in Z$  with  $v \in R$  corresponds to a parent set of  $v$  that contains only vertices from  $S$ . Moreover, an edge  $\{v, w\} \in Y$  with  $v \in R$  corresponds to a parent set of  $v$  that contains  $w \in Q_0$  and vertices from  $S$ . Finally, an edge  $\{v, w\} \in X$  means that either  $v \in P_w^{A_R}$  or  $w \in P_v^{A_R}$ . An example of the construction is shown in Figure 2.

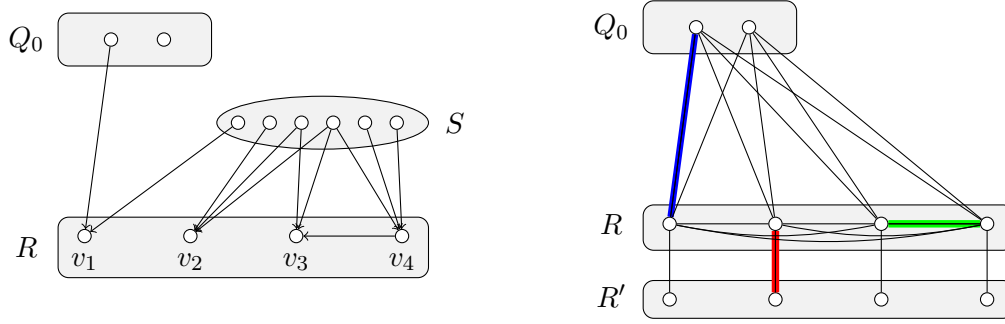


Figure 2: An example of the construction from the proof of Proposition 16. The left side shows the vertex sets  $Q_0$ ,  $S$ , and  $R$  of an instance of BASEMENT LEARNING together with an optimal basement arc set. The right side shows the edges of the corresponding instance of WEIGHTED MATCHING where the edges of a solution are labeled with colors blue, red, and green. The blue edge corresponds to the parent set of  $v_1$ , the red edge corresponds to the parent set of  $v_2$ , and the green edge corresponds to the parent sets of  $v_3$  and  $v_4$ .

*Correctness:* We now prove the correctness of the reduction, that is, we show that  $I$  is a yes-instance of BASEMENT LEARNING if and only if  $(G, \omega, \ell)$  is a yes-instance of MAXIMUM WEIGHT MATCHING.

( $\Rightarrow$ ) Let  $A_R$  be a basement arc set of  $S$ ,  $Q_0$ , and  $Q_1$  with  $\sum_{v \in R} f_v(P_v^{A_R}) \geq t$ . We define a matching  $M$  with  $\sum_{e \in M} \omega(e) \geq t$ . To this end, we describe which edges of  $X$ ,  $Y$ , and  $Z$  we add to  $M$  by defining sets  $M_X$ ,  $M_Y$ , and  $M_Z$  and set  $M := M_X \cup M_Y \cup M_Z$ .

First, for every pair  $v, w \in R$  with  $v \in P_w^{A_R}$  or  $w \in P_v^{A_R}$ , we add  $\{v, w\} \in X$  to  $M_X$ . Second, for every pair  $v, w$  with  $v \in R$ ,  $w \in Q_0$ , and  $w \in P_v^{A_R}$ , we add  $\{v, w\}$  to  $M_Y$ . Third, for every  $v \in R$  that is not incident with one of the edges in  $M_X \cup M_Y$ , we add  $\{v, v'\}$  to  $M_Z$ . Obviously,  $M_X$ ,  $M_Y$ , and  $M_Z$  are pairwise disjoint.

We first show that  $M$  is a matching by proving that there is no pair of distinct edges in  $M$  that share an endpoint. Consider the following cases.

**Case 1:**  $e_1, e_2 \in M_Z$ . Then, if  $e_1, e_2$  share one endpoint  $v \in R$  or  $v' \in R'$  it follows by the definition of  $M_Z$  that  $e_1 = e_2 = \{v, v'\}$  and, therefore, there are no distinct edges  $e_1, e_2 \in M_Z$  that share exactly one endpoint.

**Case 2:**  $e_1, e_2 \in M_X \cup M_Y$ . Then, assume towards a contradiction that  $e_1 = \{u, v\}$  and  $e_2 = \{v, w\}$  have a common endpoint  $v$ . Now,  $\{u, v\} \in M_X \cup M_Y$  implies  $(u, v) \in A_R$  or  $(v, u) \in A_R$ . Moreover  $\{v, w\} \in M_X \cup M_Y$  implies  $(w, v) \in A_R$  or  $(v, w) \in A_R$ . Then,  $v \in R \cup Q_0$  is incident with two arcs in  $A_R \cap ((R \cup Q_0) \times R)$  which contradicts the fact that  $A_R$  is a basement arc set.

**Case 3:**  $e_1 \in M_X \cup M_Y, e_2 \in M_Z$ . Then,  $e_1$  and  $e_2$  can only have a common endpoint in  $R$  which is not possible by the definition of  $M_Z$ .

We conclude by the above that  $M$  is a matching. It remains to show that  $\sum_{e \in M} \omega(e) \geq t$ . Observe that every  $v \in R$  is incident with some edge in  $M$ . Conversely, every edge in  $M_Y \cup M_Z$  has exactly one endpoint in  $R$ , and every edge in  $M_X$  has both endpoints

in  $R$ . Given an edge  $e \in M_Y \cup M_Z$ , we let  $\pi(e)$  denote its unique endpoint in  $R$ . By the construction of  $M_X$  and the fact that  $A_R$  is a basement arc set we know that for every  $\{v, w\} \in M_X$  it holds that either  $(v, w) \in A_R$  or  $(w, v) \in A_R$ . We let  $\pi_1(e)$  and  $\pi_2(e)$  denote the endpoints of  $e = \{v, w\}$  such that  $(\pi_2(e), \pi_1(e)) \in A_R$ . Since every  $v \in R$  is incident with some edge in  $M$  and  $M$  is a matching, the following sets form a partition of  $R$ .

$$\begin{aligned} R_1 &:= \{\pi_1(e) \mid e \in M_X\}, & R_2 &:= \{\pi_2(e) \mid e \in M_X\}, \\ R_3 &:= \{\pi(e) \mid e \in M_Y\}, & R_4 &:= \{\pi(e) \mid e \in M_Z\}. \end{aligned}$$

Observe that by the definitions of  $M_X, M_Y$ , and  $M_Z$  it holds that all  $v \in R_2 \cup R_4$  have a parent set  $S'$  under  $A_R$ , where  $S' \subseteq S$ . Moreover, all  $\pi(e) \in R_3$  have parent set  $P_{\pi(e)}^{A_R} = S' \cup (e \setminus \{\pi(e)\})$  with  $S' \subseteq S$ , and all  $\pi_1(e) \in R_1$  have parent sets  $P_{\pi_1(e)}^{A_R} = S' \cup \{\pi_2(e)\}$  with  $S' \subseteq S$ . For the weight of  $M$  it then holds that

$$\begin{aligned} & \sum_{e \in M_X} \omega(e) + \sum_{e \in M_Y} \omega(e) + \sum_{e \in M_Z} \omega(e) \\ & \geq \sum_{e \in M_X} \max_{S' \subseteq S} f_{\pi_1(e)}(S' \cup \{\pi_2(e)\}) + \sum_{e \in M_X} \max_{S' \subseteq S} f_{\pi_2(e)}(S') \\ & \quad + \sum_{e \in M_Y} \max_{S' \subseteq S} f_{\pi(e)}(S' \cup (e \setminus \{\pi(e)\})) + \sum_{e \in M_Z} \max_{S' \subseteq S} f_{\pi(e)}(S') \\ & \geq \sum_{v \in R_1 \cup R_2 \cup R_3 \cup R_4} f_v(P_v^{A_R}) \geq t, \end{aligned}$$

and therefore  $\sum_{e \in M} \omega(e) \geq t$ .

( $\Leftarrow$ ) Conversely, let  $M \subseteq E$  be a matching of  $G$  with  $\sum_{e \in M} \omega(e) \geq t$ . Note that in  $G$ , every edge  $e \in E$  has at least one endpoint in  $R$  and consequently every  $e \in M$  has at least one endpoint in  $R$ . Moreover, without loss of generality we can assume that every vertex of  $R$  is incident with an edge of  $M$ : If a vertex  $v \in R$  is not incident with an edge of  $M$ , we replace  $M$  by  $M' := M \cup \{\{v, v'\}\}$ . Then,  $\sum_{e \in M'} \omega(e) \geq t + \omega(\{v, v'\}) \geq t$  and  $M'$  is still a matching since  $\deg_G(v') = 1$ .

We define a set  $A_R \subseteq (S \cup Q_0 \cup R) \times R$  and show that  $\sum_{v \in R} f_v(P_v^{A_R}) \geq t$  and that  $A_R$  is a basement arc set. To this end, we define a parent set with vertices in  $S \cup Q_0 \cup R$  for every  $v \in R$ . First, if  $v$  is incident with an edge  $\{v, v'\} \in M \cap Z$ , we set  $P_v^{A_R} := \operatorname{argmax}_{S' \subseteq S} f_v(S')$ . Second, if  $v$  is incident with an edge  $\{v, w\} \in M \cap Y$ , then  $w \in Q_0$  and we set  $P_v^{A_R} := \{w\} \cup \operatorname{argmax}_{S' \subseteq S} f_v(S' \cup \{w\})$ . Third, it remains to define the parent sets of vertices in  $R$  that are endpoints of some edge in  $M \cap X$ . Let  $\{v, w\} \in M \cap X$ , where  $\varphi(v, w) \geq \varphi(w, v)$ . We then set  $P_v^{A_R} := \{w\} \cup \operatorname{argmax}_{S' \subseteq S} f_v(S' \cup \{w\})$  and we set  $P_w^{A_R} := \operatorname{argmax}_{S' \subseteq S} f_w(S')$ .

We first show that  $A_R$  is a basement arc set. Obviously,  $A_R$  does not contain self-loops and no  $v \in R$  has a parent in  $Q_1$ . It remains to show that every vertex in  $Q_0 \cup R$  has at most one incident arc in  $A_R \cap ((R \cup Q_0) \times R)$ . Let  $v \in Q_0 \cup R$ . Assume towards a contradiction that  $v$  is incident with two distinct arcs in  $A_R \cap ((R \cup Q_0) \times R)$ . Then, there exists a vertex  $w_1 \in Q_0 \cup R$  with  $(v, w_1) \in A_R$  or  $(w_1, v) \in A_R$ . Moreover, there exists a vertex  $w_2 \in (Q_0 \cup R) \setminus \{w_1\}$  with  $(w_2, v) \in A_R$  or  $(v, w_2) \in A_R$ . Then, by the definition

of  $A_R$  we conclude  $\{v, w_1\}, \{v, w_2\} \in M$  which contradicts the fact that no two edges in  $M$  share one endpoint. We conclude that  $A_R$  is a basement arc set.

It remains to show that  $\sum_{v \in R} f_v(P_v^{AR}) \geq t$ . To this end consider the following.

**Claim 2**

- a) If  $\{v, w\} \in M \cap (Y \cup Z)$  with  $v \in R$ , then  $\omega(\{v, w\}) = f_v(P_v^{AR})$ .
- b) If  $\{v, w\} \in M \cap X$ , then  $\omega(\{v, w\}) = f_v(P_v^{AR}) + f_w(P_w^{AR})$ .

PROOF a) If  $\{v, w\} \in M \cap Z$ , then  $w = v'$  and by the definition of  $A_R$  we have  $f_v(P_v^{AR}) = \max_{S' \subseteq S} f_v(S') = \omega(\{v, v'\})$ . Otherwise, if  $\{v, w\} \in M \cap Y$ , then  $w \in Q_0$  and analogously  $f_v(P_v^{AR}) = \max_{S' \subseteq S} f_v(S' \cup \{w\}) = \omega(\{v, w\})$ .

b) If  $\{v, w\} \in M \cap X$ , then  $v, w \in R$ . We only consider the case  $\varphi(v, w) \geq \varphi(w, v)$ , since the other case is analogue. It then follows from the definition of  $A_R$ , that

$$\begin{aligned} & f_v(P_v^{AR}) + f_w(P_w^{AR}) \\ &= \max_{S' \subseteq S} f_v(S' \cup \{w\}) + \max_{S' \subseteq S} f_w(S') \\ &= \max(\varphi(v, w), \varphi(w, v)) = \omega(\{v, w\}). \end{aligned}$$

◇

Now, let  $\tilde{R} \subseteq R$  be the set of vertices in  $R$  that are incident with an edge in  $M \cap X$ . We conclude by Claim 2 and the assumption that every  $v \in R$  is incident with an edge in  $M$  that

$$\begin{aligned} \sum_{v \in R} f_v(P_v^{AR}) &= \sum_{v \in R \setminus \tilde{R}} f_v(P_v^{AR}) + \sum_{v \in \tilde{R}} f_v(P_v^{AR}) \\ &= \sum_{e \in M \cap (Y \cup Z)} \omega(e) + \sum_{e \in M \cap X} \omega(e) \\ &= \sum_{e \in M} \omega(e) \geq t, \end{aligned}$$

which completes the correctness proof.

*Running Time.* The constructed instance of MAXIMUM WEIGHT MATCHING contains  $\mathcal{O}(n)$  vertices and  $\mathcal{O}(n^2)$  edges. For each edge  $e$ , the edge weight  $\omega(e)$  can be computed in  $\mathcal{O}(n\delta_{\mathcal{F}})$  time. Hence, we can compute the described instance of MAXIMUM WEIGHT MATCHING from an instance of BASEMENT LEARNING in  $\mathcal{O}(n^3\delta_{\mathcal{F}})$  time. Together with the fact that MAXIMUM WEIGHT MATCHING can be solved in  $\mathcal{O}(\sqrt{|V|} \cdot |E|)$  time (Micali & Vazirani, 1980), we conclude that BASEMENT LEARNING can be solved in  $\mathcal{O}(n^3\delta_{\mathcal{F}})$  time. □

#### 4.4 An XP-time Algorithm for $(\Pi_1 + v)$ -Moral BNSL

We now combine the previous results of this section to obtain an XP-time algorithm for  $(\Pi_1 + v)$ -MORAL BNSL. Recall that the intuitive idea of the algorithm is to find the dissociation set  $S$  and the ancestors  $Q = Q_0 \cup Q_1$  of  $S$  via brute force. Then, for every choice of  $S$ ,  $Q_0$ , and  $Q_1$  we find an attic arc set and a basement arc set and combine these arc sets to a solution of  $(\Pi_1 + v)$ -MORAL BNSL.



**Theorem 17**  $(\Pi_1 + v)$ -MORAL BNSL can be solved in  $n^{3k} \cdot k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$  time.

*PROOF Algorithm.* Let  $I = (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_1 + v)$ -MORAL BNSL. The following algorithm decides whether  $I$  is a yes-instance or a no-instance: First, iterate over all possible choices of  $S$ ,  $Q_0$ , and  $Q_1$  where  $|S| \leq k$  and  $|Q_0 \cup Q_1| \leq 2k$ . For each such choice do the following:

- Compute an optimal attic arc set  $A_Q$  using the algorithm behind Proposition 15.
- Let  $t' := t - \sum_{v \in S \cup Q} f_v(P_v^{A_Q})$  and check if  $(N, S, Q_0, Q_1, \mathcal{F}, t')$  is a yes-instance of BASEMENT LEARNING. If this is the case, return *yes*.

If for none of the choices of  $S$ ,  $Q_0$ , and  $Q_1$  the answer *yes* was returned, then return *no*.

*Running Time.* First, we discuss the running time of the algorithm: Since  $|S| \leq k$  and  $|Q_0 \cup Q_1| \leq 2k$ , there are at most  $\binom{n}{k} \cdot \binom{n-k}{2k} \in \mathcal{O}(n^{3k})$  choices for  $S$  and  $Q := Q_0 \cup Q_1$ . For each such choice we can compute all  $4^k$  possible partitions of  $Q$  into two sets. Hence, we can iterate over all possible choices for  $S$ ,  $Q_0$  and  $Q_1$  in  $\mathcal{O}(n^{3k} \cdot 4^k)$  time. Afterwards, for each such choice we apply the algorithm behind Proposition 15 in  $k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$  time, and the algorithm behind Proposition 16 in  $\mathcal{O}(n^3 \delta_{\mathcal{F}})$  time. This gives an overall running time of  $n^{3k} \cdot k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$  as claimed.

*Correctness.* Second, we show the correctness of the algorithm by proving that the algorithm returns yes if and only if  $I$  is a yes-instance of  $(\Pi_1 + v)$ -MORAL BNSL.

( $\Rightarrow$ ) Let the algorithm return yes for  $I$ . Then, there exist disjoint sets  $S$ ,  $Q_0$ , and  $Q_1$  with  $|S| \leq k$ , an attic arc set  $A_Q$ , and a basement arc set  $A_R$  such that  $\sum_{v \in S \cup Q} f_v(P_v^{A_Q}) + \sum_{v \in R} f_v(P_v^{A_R}) \geq t$ . Then, by Lemma 12, the graph  $(N, A_Q \cup A_R)$  is a DAG whose moralized graph has dissociation set  $S$  and by Lemma 14 its score is the sum of the local scores for  $A_R$  and  $A_Q$ . Hence,  $I$  is a yes-instance.

( $\Leftarrow$ ) Conversely, let  $(N, \mathcal{F}, t, k)$  be a yes-instance. Then, there is an  $(N, \mathcal{F}, t)$ -valid DAG  $D = (N, A)$  whose moralized graph has a dissociation set  $S$  of size at most  $k$ . Then, by Lemma 13, there exist disjoint sets  $Q_0, Q_1$ , an attic arc set  $A_Q$  and a basement arc set  $A_R$  such that  $A_R \cup A_Q = A$ . Furthermore,  $|Q_0 \cup Q_1| \leq 2k$ . Since the algorithm iterates over all possible choices for  $S$ ,  $Q_0$ , and  $Q_1$  with  $|S| \leq k$  and  $|Q_0 \cup Q_1| \leq 2k$ , it considers  $S$ ,  $Q_0$ , and  $Q_1$  at some point. Since  $A$  is  $(N, \mathcal{F}, t)$ -valid, the arc set  $A_R$  satisfies  $\sum_{v \in R} f_v(P_v^{A_R}) \geq t - \sum_{v \in S \cup Q} f_v(P_v^{A_Q})$ . Hence, the algorithm returns yes.  $\square$

The running time stated in Theorem 17 contains a factor of  $k^{\mathcal{O}(k)}$ . Let us remark that the constant in the exponent hidden by the  $\mathcal{O}$ -notation is not too high: The constant relies on the running time of the algorithm behind Proposition 15 where we iterate over the possible auxiliary graphs. Since  $|Q_1| \leq 2k$ , the number of iterations is  $\mathcal{O}(\binom{2k}{2k}^2)$ . Thus, the hidden constant is 4. While it seems possible that this can be improved, it would be more interesting to determine whether the factor of  $k^{\mathcal{O}(k)}$  can be replaced by  $2^{\mathcal{O}(k)}$ .

#### 4.5 $(\Pi_1 + v)$ -Moral BNSL Parameterized by $\mathbf{k} + \mathbf{t}$

In this section, we show another hardness result for  $(\Pi_1 + v)$ -MORAL BNSL. Due to Corollary 8, the problem is presumably not FPT for parameter  $k$  even if the maximum parent set

size is 1. Observe that this implies that even a parameterization by  $k + p$ , where  $p$  denotes the maximum parent set size, is unlikely to lead to an FPT algorithm. In the next theorem, we prove  $W[1]$ -hardness for  $k + t$ , even if  $p = 3$  and  $\delta_{\mathcal{F}} = 2$  implying that not even for the sum  $k + t + \delta_{\mathcal{F}} + p$  there is an FPT algorithm unless  $W[1] = \text{FPT}$ . The proof is closely related to the  $W[1]$ -hardness proof for  $(\Pi_0 + v)$ -MORAL BNSL (Korhonen & Parviainen, 2015). We provide it here for sake of completeness.

**Theorem 18**  $(\Pi_1 + v)$ -MORAL BNSL is  $W[1]$ -hard for  $k + t$ , even when the directed superstructure  $S_{\overline{\mathcal{F}}}$  is a DAG, the maximum parent set size is 3,  $\delta_{\mathcal{F}} = 2$ , and every local score is either 1 or 0.

**PROOF** We give a parameterized reduction from CLIQUE. In CLIQUE we are given an undirected graph  $G = (V, E)$  and an integer  $\ell$  and the question is if there exists a subset  $K \subseteq V$  of size  $\ell$  such that the vertices in  $K$  are pairwise adjacent in  $G$ . CLIQUE is  $W[1]$ -hard if parameterized by  $\ell$  (Downey & Fellows, 1995). Let  $(G = (V, E), \ell)$  be an instance of CLIQUE. We describe how to construct an equivalent instance of  $(\Pi_1 + v)$ -MORAL BNSL where  $k + t \in \mathcal{O}(\ell^2)$ .

*Construction.* We first define the vertex set  $N$  by  $N := V \cup (E \times \{1\}) \cup (E \times \{2\})$ . We write  $e_i := (e, i)$  for the elements in  $E \times \{i\}$ ,  $i \in \{1, 2\}$ . Next, we define the local scores  $\mathcal{F}$ . For every edge  $e = \{u, v\} \in E$  we set  $f_{e_1}(\{e_2, u, v\}) := 1$ . All other local scores are set to 0. Finally, we set  $t := \binom{\ell}{2}$  and  $k := \ell$ . Note that  $k + t \in \mathcal{O}(\ell^2)$  and that the maximum parent set size is 3. Moreover, in the directed superstructure  $S_{\overline{\mathcal{F}}}$ , the vertices in  $E \times \{1\}$  are the only vertices that have incoming arcs, and all vertices in  $E \times \{1\}$  are sinks. Hence,  $S_{\overline{\mathcal{F}}}$  is a DAG.

*Correctness.* We next show that there is a clique of size  $k$  in  $G$  if and only if there exists an  $(N, \mathcal{F}, t)$ -valid arc set  $A$  such that  $\mathcal{M}(N, A)$  has a dissociation set of size at most  $k$ .

( $\Rightarrow$ ) Let  $K$  be a clique of size  $\ell$  in  $G$ . We then define  $A := \{(e_2, e_1), (u, e_1), (v, e_1) \mid e = \{u, v\} \text{ with } u, v \in K\}$ . We prove that  $A$  is  $(N, \mathcal{F}, t)$ -valid and  $\mathcal{M}(D)$  for  $D := (N, A)$  has a dissociation set of size at most  $k$ .

From the fact that  $S_{\overline{\mathcal{F}}}$  is a DAG, we conclude that  $D$  is a DAG. Moreover, since the vertices of  $K$  are pairwise adjacent in  $G$ , it follows by the construction of  $\mathcal{F}$  that  $f_{e_1}(P_{e_1}^A) = 1$  for all  $e \in E_G(K)$ . Since  $|K| = \ell$  we conclude  $\sum_{v \in N} f_v(P_v^A) = \binom{\ell}{2} = t$ . Hence,  $A$  is  $(N, \mathcal{F}, t)$ -valid.

Finally, we show that  $K$  is a dissociation set of size  $k$  in  $\mathcal{M}(D)$ . Observe that the vertices  $e_1 \in N$  with  $e \in E_G(K)$  are the only vertices that have a non-empty parent set in  $D$ . Hence, the moralized graph is  $\mathcal{M}(D) = (N, E_{\mathcal{M}})$ , where

$$E_{\mathcal{M}} = \bigcup_{e: \{u, v\} \in E_G(K)} \{\{w_1, w_2\} \mid w_1, w_2 \in \{u, v, e_1, e_2\}\}.$$

It follows that  $\mathcal{M}(D) - K = (N \setminus K, E'_{\mathcal{M}})$  with

$$E'_{\mathcal{M}} = \bigcup_{e \in E_G(K)} \{\{e_1, e_2\}\}.$$

Therefore, the maximum degree in  $\mathcal{M}(D) - K$  is at most 1.

( $\Leftarrow$ ) Conversely, let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that the moralized graph  $\mathcal{M}(D)$ , where  $D := (N, A)$ , has a dissociation set  $S$  of size at most  $k$ . We prove that there is a clique of size  $k$  in  $G$ .

Since  $A$  is  $(N, \mathcal{F}, t)$ -valid, we know that  $\sum_{v \in N} f_v(P_v^A) \geq \binom{k}{2}$ . Together with the definition of  $\mathcal{F}$  it follows that there are at least  $\binom{k}{2}$  vertices  $e_1 \in N$  such that  $P_{e_1}^A = \{e_2, u, v\}$  with  $\{u, v\} = e$ . Let  $X \subseteq N$  be the set of these vertices  $e_1$ . Note that  $|X| \geq \binom{k}{2}$  and recall that for every  $e_1 \in X$  it holds that  $e$  is an edge of  $G$ .

For every  $e_1 \in X$  with  $e = \{u, v\}$  the set  $\{e_1, e_2, u, v\}$  is a clique of size 4 in  $\mathcal{M}(D)$ . Since  $S$  is a dissociation set in  $\mathcal{M}(D)$ , at least two vertices from each set  $\{e_1, e_2, u, v\}$  belong to  $S$ . We may assume that these vertices are  $u$  and  $v$  by a simple exchange argument.

Now, consider  $S \cap V$ . Since for every  $e_1 \in X$  with  $e = \{u, v\}$  the vertices  $u$  and  $v$  belong to  $S$ , we conclude that  $|E_G(S \cap V)| \geq \binom{k}{2}$ . Together with the fact that  $|S| \leq k$ , it follows that  $S \cap V$  is a set of  $k$  vertices with at least  $\binom{k}{2}$  edges between them. Therefore,  $S \cap V$  is a clique of size  $k$  in  $G$ .  $\square$

### 5. Constrained BNSL for Related Graph Classes

In this section we outline the limits of learning Bayesian networks under a sparsity constraints that are related to a bounded vertex cover number and a bounded dissociation number. Recall that a bound on the vertex cover number or the dissociation number automatically implies a bound on the treewidth, and that for efficient inference it is desirable to have a small treewidth in the moralized graph (Darwiche, 2009). Thus, it is well motivated to study constrained BNSL problems regarding graph classes that give a bound on the treewidth that lies between the treewidth and the dissociation number.

In terms of graph-classes, the bound on the vertex cover number is formalized as the graph class  $\Pi_0 + kv$  and the bound on the dissociation number is formalized as the graph class  $\Pi_1 + kv$ . Recall that  $\Pi_0$  is the class of edgeless graphs. Equivalently,  $\Pi_0$  is the class of graphs with maximum degree 0, or the graphs with maximum connected component size 1. Analogously,  $\Pi_1$  is the class of graphs with maximum degree 1, or the class of graphs with maximum connected component size 2. In this section, we consider two superclasses of  $\Pi_1$  and show that XP-time algorithms for constrained BNSL problems regarding these superclasses are presumably not possible.

Let  $\Pi_2$  be the class of graphs that have maximum degree 2, and let  $\Pi_c^{\text{COC}}$  be the class of graphs where each connected component has size at most  $c$  for a fixed integer  $c \geq 3$ . These graph classes are superclasses of  $\Pi_1$ , that is  $\Pi_1 \subseteq \Pi_2$  and  $\Pi_1 \subseteq \Pi_c^{\text{COC}}$ . Consequently, if a graph  $G$  belongs to the graph class  $\Pi_1 + kv$  for some  $k \in \mathbb{N}_0$ , then there exist  $k' \leq k$  and  $k'' \leq k$  such that  $G \in \Pi_2 + k'v$  and  $G \in \Pi_c^{\text{COC}} + k''v$ . Moreover, observe that the treewidth of  $G$  is not bigger than  $\min(k', k'') + \mathcal{O}(1)$ .

With the next two theorems we show that there is little hope that  $(\Pi + v)$ -SKELETON BNSL or  $(\Pi + v)$ -MORAL BNSL with  $\Pi \in \{\Pi_2, \Pi_3^{\text{COC}}\}$  has an XP-time algorithm when parameterized by  $k$ . To prove the first result we use a reduction from HAMILTONIAN PATH. This construction was already used to show that BNSL is NP-hard if one adds the restriction that the resulting network must be a directed path (Meek, 2001). In the following we show that it also works for  $(\Pi_2 + v)$ -SKELETON BNSL and  $(\Pi_2 + v)$ -MORAL BNSL.

**Theorem 19**  $(\Pi_2 + v)$ -SKELETON BNSL and  $(\Pi_2 + v)$ -MORAL BNSL are NP-hard even if  $k = 0$  and the maximum parent set size is 1.

PROOF We give a polynomial-time reduction from the NP-hard HAMILTONIAN PATH problem to  $(\Pi_2 + v)$ -SKELETON BNSL. Afterwards we show that the reduction is also correct for  $(\Pi_2 + v)$ -MORAL BNSL. In HAMILTONIAN PATH one is given an undirected graph  $G$  and the question is whether there exists a Hamiltonian path, that is, a path which contains every vertex of  $G$  exactly once.

*Construction.* Let  $G = (V, E)$  be an instance of HAMILTONIAN PATH with  $n$  vertices. We describe how to construct an equivalent instance of  $(\Pi_2 + v)$ -BNSL where  $k = 0$ . We first set  $N := V$ . Next, for every  $v \in N$  we set  $f_v(\{w\}) = 1$  if  $w \in N_G(v)$  and  $f_v(P) = 0$  for every other  $P \subseteq N \setminus \{v\}$ . Finally, we set  $t := n - 1$  and  $k := 0$ .

*Correctness.* We next show that  $G$  is a yes-instance of HAMILTONIAN PATH if and only if  $(N, \mathcal{F}, t, 0)$  is a yes-instance of  $(\Pi_2 + v)$ -SKELETON BNSL.

( $\Rightarrow$ ) Let  $P = (v_1, v_2, \dots, v_n)$  be a Hamiltonian path in  $G$ . We set  $A := \{(v_i, v_{i+1}) \mid i \in \{1, \dots, n-1\}\}$  and show that  $A$  is  $(N, \mathcal{F}, t)$ -valid and that  $\mathcal{S}(N, A)$  has maximum degree 2.

Since  $P$  is a Hamiltonian path, no vertex appears twice on  $P$ . Hence,  $(N, A)$  does not contain directed cycles. Moreover, it holds that  $v_i \in N_G(v_{i-1})$  for every  $i \in \{2, \dots, n\}$  and therefore  $\sum_{v \in N} f_v(P_v^A) = n - 1 = t$ . Hence,  $A$  is  $(N, \mathcal{F}, t)$ -valid. Moreover, observe that  $\mathcal{S}(N, A) = (N, \{(v_i, v_{i+1}) \mid i \in \{1, \dots, n-1\}\})$  and this,  $\mathcal{S}(N, A)$  has maximum degree 2.

( $\Leftarrow$ ) Conversely, let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that  $\mathcal{S}(D)$  has maximum degree at most 2, where  $D := (N, A)$ . Since  $t = n - 1$  and every local score is either 1 or 0 we conclude that  $f_v(P_v^A) = 1$  for at least  $n - 1$  vertices. Then, there are at least  $n - 1$  arcs in  $A$ , and thus there are at least  $n - 1$  edges in  $\mathcal{S}(D)$ . Furthermore, we may assume that in  $D$  no vertex  $v$  has a non-empty parent set with score  $f_v(P_v^A) = 0$  since otherwise we may replace it with  $\emptyset$ . This implies that no vertex has more than one parent in  $D$ . Consequently,  $\mathcal{S}(D)$  is acyclic.

Since  $\mathcal{S}(D)$  is acyclic, the maximum degree is 2, and there are at least  $n - 1$  edges, there is a Hamiltonian path  $P := (v_1, \dots, v_n)$  in  $\mathcal{S}(D)$ . We show that  $P$  is a Hamiltonian path in  $G$ . Let  $v_i, v_{i+1}$  be two consecutive vertices on  $P$ . Then, either  $(v_i, v_{i+1}) \in A$  or  $(v_{i+1}, v_i) \in A$ . By the construction of  $\mathcal{F}$  we have  $v_i \in N_G(v_{i+1})$  and thus,  $\{v_i, v_{i+1}\} \in E$ . Therefore,  $P$  is a Hamiltonian path in  $G$ .

*Moralized Graph.* We next argue why the construction described above is also a correct reduction from HAMILTONIAN PATH to  $(\Pi_2 + v)$ -MORAL BNSL.

For the forward direction, let  $P$  be a Hamiltonian path in  $G$  and let the arc set  $A$  be defined as above. Since every vertex has at most one incoming arc from  $A$ , the moralized graph  $\mathcal{M}(N, A)$  has no moral edges and therefore  $\mathcal{M}(N, A)$  and  $\mathcal{S}(N, A)$  have the same set of edges. Thus,  $\mathcal{M}(N, A)$  has maximum degree 2.

For the backwards direction, let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that the moralized graph  $\mathcal{M}(N, A)$  has maximum degree at most 2. Since the edge set of the skeleton of  $(N, A)$  is a subset of the edge set of  $\mathcal{M}(N, A)$ , we conclude that  $\mathcal{S}(N, A)$  has maximum degree at most 2. Then, by the above argumentation, there exists a Hamiltonian path in  $G$ .  $\square$

**Theorem 20** *Let  $c \geq 3$ . Then,  $(\Pi_c^{\text{COC}} + v)$ -SKELETON BNSL and  $(\Pi_c^{\text{COC}} + v)$ -MORAL BNSL are NP-hard even if  $k = 0$ .*

**PROOF** We give a polynomial-time reduction from the NP-hard problem  $c$ -CLIQUE COVER (Kirkpatrick & Hell, 1983) to  $(\Pi_c^{\text{COC}} + v)$ -SKELETON BNSL. Afterwards, we show that the reduction is also correct for  $(\Pi_c^{\text{COC}} + v)$ -MORAL BNSL. In  $c$ -CLIQUE COVER one is given an undirected graph  $G = (V, E)$  and the question is whether there exists a packing  $\mathcal{P}$  of vertex-disjoint cliques of size  $c$  such that every vertex of  $G$  belongs to one clique of the packing. We represent the packing of cliques  $\mathcal{P} := \{K_c^i \mid i \in \{1, \dots, \lfloor \frac{|V|}{c} \rfloor\}\}$  as a partition of  $V$ , where every  $K_c^i$  is a clique of size  $c$ .

*Construction.* Let  $G = (V, E)$  be an instance of  $c$ -CLIQUE COVER with  $n$  vertices. We describe how to construct an equivalent instance of  $(\Pi_c^{\text{COC}} + v)$ -SKELETON BNSL where  $k = 0$ . We first set  $N := V$ . Next, for every  $v \in N$  we set  $f_v(P) = 1$  if  $G[P]$  is a  $K_{c-1}$  and  $P \subseteq N_G(v)$ . Otherwise, we set  $f_v(P) = 0$ . Note that  $\mathcal{F}$  can be computed in polynomial time since  $c$  is a constant. Finally, we set  $t := \frac{n}{c}$  and  $k := 0$ .

*Correctness.* We next show that  $G$  is a yes-instance of  $c$ -CLIQUE COVER if and only if  $(N, \mathcal{F}, t, 0)$  is a yes-instance of  $(\Pi_c^{\text{COC}} + v)$ -SKELETON BNSL.

( $\Rightarrow$ ) Let  $\mathcal{P} = \{K_c^i \mid i \in \{1, \dots, \frac{n}{c}\}\}$  be a packing of vertex-disjoint cliques of size  $c$  that cover  $G$ . For each  $i$ , let  $v_i$  be one vertex of  $K_c^i$ . We set  $A := \bigcup_{i \in \{1, \dots, \frac{n}{c}\}} \{(u, v_i) \mid u \in K_c^i \setminus \{v_i\}\}$ . Then,  $D := (N, A)$  is a union of disjoint stars, where the arcs of each star point to the center  $v_i$ . Therefore,  $D$  is a DAG and every connected component of  $\mathcal{S}(D)$  has order  $c$ . Therefore,  $\mathcal{S}(D) \in \Pi_c^{\text{COC}}$ . Finally, observe that  $f_{v_i}(P_{v_i}^A) = 1$  for each vertex  $v_i$  by the definition of  $\mathcal{F}$ . Consequently,  $\sum_{v \in N} f_v(P_v^A) \geq \frac{n}{c} = t$ .

( $\Leftarrow$ ) Conversely, let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that every connected component of  $\mathcal{S}(N, A)$  has order at most  $c$ . Since  $t = \frac{n}{c}$  and the local scores are either 0 or 1, there are pairwise distinct vertices  $v_1, \dots, v_{\frac{n}{c}}$  with  $f_{v_i}(P_{v_i}^A) = 1$  for every  $i \in \{1, \dots, \frac{n}{c}\}$ . We define  $\mathcal{P} := \{K^i := \{v_i\} \cup P_{v_i}^A \mid i \in \{1, \dots, \frac{n}{c}\}\}$  and show that  $\mathcal{P}$  is a packing of vertex disjoint size  $c$  cliques that cover  $G$ .

By the definition of  $\mathcal{F}$ , each  $P_{v_i}^A$  is a clique of size  $c - 1$  that is completely contained in  $N_G(v_i)$ . Thus, every  $K^i \in \mathcal{P}$  is a size- $c$  clique in  $G$ . Next, assume towards a contradiction that there are distinct indices  $i$  and  $j$  with  $K^i \cap K^j \neq \emptyset$ . Then, since  $v_i \neq v_j$  it follows that  $K^i \cup K^j$  is a connected component of size bigger than  $c$  in  $\mathcal{M}(N, A)$ . This contradicts the choice of  $A$ .

*Moralized Graph.* We next argue why the construction described above is also a correct reduction from  $c$ -CLIQUE COVER to  $(\Pi_3^{\text{COC}} + v)$ -MORAL BNSL.

( $\Rightarrow$ ) Let  $\mathcal{P}$  be a packing of vertex disjoint size- $c$  cliques that cover  $G$ , and let  $A$  be defined as above. Then,  $(N, A)$  is a disjoint union of stars that point to the center of  $v_i$ . Thus,  $\mathcal{M}(N, A)$  is a disjoint union of cliques of size  $c$ . Consequently  $\mathcal{M}(N, A) \in \Pi_c^{\text{COC}}$ .

( $\Leftarrow$ ) Let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that every connected component of  $\mathcal{M}(N, A)$  has order at most  $c$ . Since the edge set of the skeleton of  $(N, A)$  is a subset of the edge set of the moralized graph, we conclude that  $\mathcal{S}(N, A) \in \Pi_c^{\text{COC}}$ . Then, by the above argumentation,  $G$  is a yes-instance of  $c$ -CLIQUE COVER.  $\square$

Note that, given a graph class  $\Pi$ , we have  $\Pi = \Pi + 0v = \Pi + 0e$ . Thus, Theorems 19 and 20 imply the following.

**Corollary 21** *Let  $\Pi \in \{\Pi_2\} \cup \{\Pi_c^{COG} \mid c \geq 3\}$ . Then,  $(\Pi + e)$ -SKELETON BNSL and  $(\Pi + e)$ -MORAL BNSL are NP-hard even if  $k = 0$ .*

## 6. BNSL with Bounded Number of Edges

In this section we study BNSL, where we aim to learn a network such that the skeleton or the moralized graph have a bounded number of edges. Formally, we study  $(\Pi_0 + e)$ -SKELETON BNSL and  $(\Pi_0 + e)$ -MORAL BNSL, where  $\Pi_0$  is the class of edgeless graphs. Clearly,  $\Pi_0$  is monotone.

We first consider  $(\Pi_0 + e)$ -SKELETON BNSL in Subsection 6.1. We show that it is fixed-parameter tractable when parameterized by  $k$ . Afterwards we consider  $(\Pi_0 + e)$ -MORAL BNSL in Subsection 6.2. We observe that it has an XP-time algorithm when parameterized by  $k$  and it is W[1]-hard for parameterization by  $k + t$ . Thus, putting the constraint of a bounded number of edges on the moralized graph makes the learning problem harder than putting a similar constraint on the skeleton.

### 6.1 $(\Pi_0 + e)$ -Skeleton BNSL

In this subsection we consider a version of BAYESIAN NETWORK STRUCTURE LEARNING where we want to learn a Bayesian network with a bounded number of arcs or—equivalently—a bounded number of edges in the skeleton. Formally, this is the constrained BNSL problem  $(\Pi_0 + e)$ -SKELETON BNSL, where  $\Pi_0$  is the class of edgeless graphs.

#### A Polynomial-Time Algorithm when the Directed Superstructure is Acyclic.

We first show that  $(\Pi_0 + e)$ -SKELETON BNSL becomes polynomial-time solvable if the directed superstructure is a DAG. The algorithm uses dynamic programming over a topological ordering of  $S_{\mathcal{F}}$ . That is, an ordering  $(v_1, \dots, v_n)$  of the vertices of  $N$  such that for every arc  $(v_i, v_j)$  of  $S_{\mathcal{F}}$  it holds that  $i < j$ .

**Proposition 22**  *$(\Pi_0 + e)$ -SKELETON BNSL can be solved in  $\mathcal{O}(\delta_{\mathcal{F}} \cdot k \cdot n)$  time if the directed superstructure is a DAG.*

PROOF Let  $N := \{1, \dots, n\}$ , and let  $(N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + e)$ -SKELETON BNSL such that  $S_{\mathcal{F}}$  is a DAG. Without loss of generality, let  $(n, n - 1, \dots, 2, 1)$  be a topological ordering of  $S_{\mathcal{F}}$ . Hence, for every arc  $(a, b)$  of  $S_{\mathcal{F}}$  it holds that  $a > b$ .

The dynamic programming table  $T$  has entries of the type  $T[i, j]$  for all  $i \in \{0, 1, \dots, n\}$  and  $j \in \{0, 1, \dots, k\}$ . Each entry stores the maximum sum of local scores of the vertices  $(i, \dots, 1)$  of the topological ordering that can be obtained by an arc set  $A$  of size at most  $j$ . For  $i = 0$ , we set  $T[0, j] = 0$  for all  $j \in \{0, \dots, k\}$ . The recurrence to compute an entry for  $i > 0$  is

$$T[i, j] = \max_{P \in P_{\mathcal{F}}(i), |P| \leq j} (f_i(P) + T[i - 1, j - |P|]),$$

and the result can then be computed by checking if  $T[n, k] \geq t$ . The corresponding network can be found by traceback. The correctness proof is straightforward and thus omitted. The size of  $T$  is  $\mathcal{O}(n \cdot k)$  and each entry  $T[i, j]$  can be computed in  $\mathcal{O}(\delta_{\mathcal{F}})$  time by iterating over the at most  $\delta_{\mathcal{F}}$  triples  $(f_i(P), |P|, P)$  in  $\mathcal{F}$  for the vertex  $i$ . Therefore,  $(\Pi_0 + e)$ -SKELETON BNSL can be solved in  $\mathcal{O}(\delta_{\mathcal{F}} \cdot k \cdot n)$  time if  $S_{\mathcal{F}}$  is a DAG.  $\square$

**A Randomized FPT Algorithm.** The dynamic programming algorithm behind Proposition 22 can be adapted to obtain an FPT algorithm for  $(\Pi_0 + e)$ -SKELETON BNSL when parameterized by the number of arcs  $k$ . The algorithm is based on color coding (Alon, Yuster, & Zwick, 1995): In a Bayesian network with at most  $k$  arcs, there are at most  $2k$  vertices which are endpoints of such arcs. The idea of color coding is to randomly color the vertices of  $N$  with  $2k$  colors and find a solution  $A$  where all vertices that are incident with arcs of  $A$  are colored with pairwise distinct colors.

To describe the color coding algorithm, we introduce some notation. Let  $N$  be a set of vertices. A function  $\chi : N \rightarrow \{1, \dots, 2k\}$  is called a *coloring (of  $N$  with  $2k$  colors)*. Given a color  $c \in \{1, \dots, 2k\}$ , we call  $\chi^{-1}(c) := \{v \in N \mid \chi(v) = c\}$  the *color class of  $c$* . For a subset  $N' \subseteq N$ , we let  $\chi(N') := \{\chi(v) \mid v \in N'\}$ , and for a subset  $C \subseteq \{1, \dots, 2k\}$  we let  $\chi^{-1}(C) := \bigcup_{c \in C} \chi^{-1}(c)$ . The following definition is important for our algorithm.

**Definition 23** *Let  $N$  be a set of vertices and let  $\chi : N \rightarrow \{1, \dots, 2k\}$  be a coloring of  $N$ . An arc set  $A \subseteq N \times N$  is called color-loyal for  $\chi$  if for every color class  $\chi^{-1}(c)$  it holds that*

- a) *there is no  $(v, w) \in A$  with  $v, w \in \chi^{-1}(c)$ , and*
- b) *there exists an ordering  $(c_1, \dots, c_{2k})$  of the colors  $1, \dots, 2k$  such that every  $(v, w) \in A$  satisfies  $v \in \chi^{-1}(c_i)$  and  $w \in \chi^{-1}(c_j)$  for some  $i < j$ .*

Consider the following auxiliary problem.

COLORED  $(\Pi_0 + e)$ -SKELETON BNSL

**Input:** A set of vertices  $N$ , local scores  $\mathcal{F} = \{f_v \mid v \in N\}$ , two integers  $t, k \in \mathbb{N}$ , and a coloring  $\chi : N \rightarrow \{1, \dots, 2k\}$ .

**Question:** Is there an  $(N, \mathcal{F}, t)$ -valid arc set  $A \subseteq N \times N$  that is color-loyal for  $\chi$  and  $|A| \leq k$ ?

Recall that the intuitive idea behind the color coding algorithm is to randomly color the vertices of  $N$  with  $2k$  colors and find a solution  $A$  satisfying a constraint regarding the random coloring. COLORED  $(\Pi_0 + e)$ -SKELETON BNSL is the problem that we solve after we randomly choose the coloring. The correspondence between  $(\Pi_0 + e)$ -SKELETON BNSL and its colored version is stated in the following proposition.

**Proposition 24** *Let  $I = (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + e)$ -SKELETON BNSL. If  $I$  is a yes-instance of  $(\Pi_0 + e)$ -SKELETON BNSL, then there exist at least  $(2k)!(2k)^{(n-2k)}$  colorings  $\chi : N \rightarrow \{1, 2, \dots, 2k\}$  such that  $(N, \mathcal{F}, t, k, \chi)$  is a yes-instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL.*

**PROOF** Let  $I$  be a yes-instance of  $(\Pi_0 + e)$ -SKELETON BNSL. Then, there is an  $(N, \mathcal{F}, t)$ -valid arc set  $A$  with  $|A| \leq k$ . Observe that  $|A| \leq k$  implies that there are at most  $2k$  vertices of  $N$  that are endpoints of arcs in  $A$ .

We define a set  $\mathbb{X}$  of colorings of  $N$ , such that  $\chi \in \mathbb{X}$  if  $\chi$  assigns all vertices incident with arcs of  $A$  to pairwise distinct colors. Since at most  $2k$  vertices are endpoints of arcs in  $A$ , we conclude that  $|\mathbb{X}| \geq (2k)!(2k)^{(n-2k)}$ . Let  $I' := (N, \mathcal{F}, t, k, \chi)$  be an instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL for some arbitrary  $\chi \in \mathbb{X}$ . We show that  $A$  is a solution of  $I'$ . Note that  $A$  is  $(N, \mathcal{F}, t)$ -valid, so it remains to show that  $A$  is color-loyal for  $\chi$ .

Since all endpoints of arcs in  $A$  have pairwise distinct colors under  $\chi$ , at most one vertex in each color class  $\chi^{-1}(c)$  has a non-empty parent set. Let  $D' := (N', A)$  be the subgraph we obtain when removing all isolated vertices from  $(N, A)$ . Then,  $D'$  has at most  $2k$  vertices. Consider a topological ordering  $\tau := (v_1, \dots, v_{|N'|})$  of the DAG  $D'$ . Since the vertices of  $N'$  belong to pairwise different color classes, there exists a color sequence  $(c_1, \dots, c_{2k})$  where  $v_i \in \chi^{-1}(c_i)$  for all  $i \in \{1, \dots, |N'|\}$ . Let  $(v, w) \in A$ . Since  $\tau$  is a topological ordering of  $D'$  we have  $v \in \chi^{-1}(c_i)$  and  $w \in \chi^{-1}(c_j)$  for some  $i < j$ . Thus,  $A$  is color loyal for  $\chi$ . Consequently,  $I'$  is a yes-instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL.  $\square$

We next show that COLORED  $(\Pi_0 + e)$ -SKELETON BNSL parameterized by  $k$  is fixed-parameter tractable.

**Proposition 25** COLORED  $(\Pi_0 + e)$ -SKELETON BNSL can be solved in  $\mathcal{O}(4^k k^2 n^2 \delta_{\mathcal{F}})$  time.

PROOF Let  $I = (N, \mathcal{F}, t, k, \chi)$  be an instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL with  $f_v(\emptyset) = 0$  for every  $v \in N$  and let  $C := \{1, 2, \dots, 2k\}$  denote the set of colors. By Proposition 2, every instance of a constrained BNSL problem can be transformed into such an instance in  $\mathcal{O}(|\mathcal{F}|) = \mathcal{O}(n^2 \cdot \delta_{\mathcal{F}})$  time.

We fill a dynamic programming table  $T$  with entries of type  $T[C', k']$  where  $C' \subseteq C$  and  $k' \in \{0, 1, \dots, k\}$ . Every entry stores the maximum value of  $\sum_{v \in \chi^{-1}(C')} f_v(P_v^A)$  over all possible DAGs  $D = (N, A)$ , where  $A \subseteq \chi^{-1}(C') \times \chi^{-1}(C')$  is color-loyal for  $\chi$  and contains at most  $k'$  arcs. We set  $T[\{c\}, k'] := \sum_{w \in \chi^{-1}(c)} f_w(\emptyset) = 0$  for every  $c \in C$  and  $k' \in \{0, 1, \dots, 2k\}$ . The recurrence to compute the entry for  $C' \subseteq C$  with  $|C'| > 1$  is

$$T[C', k'] = \max_{c \in C'} \max_{v \in \chi^{-1}(c)} \max_{\substack{P \in \mathcal{P}_{\mathcal{F}}(v) \\ |P| \leq k' \\ \chi(P) \subseteq C' \setminus \{c\}}} T[C' \setminus \{c\}, k' - |P|] + f_v(P) + \sum_{w \in \chi^{-1}(c) \setminus \{v\}} f_w(\emptyset).$$

The result can be computed by checking if  $T[C, k] \geq t$ . Note that the corresponding network can be found via traceback. The correctness proof is straightforward and thus omitted. We next consider the running time. The size of  $T$  is  $\mathcal{O}(2^{2k} \cdot k)$ . Note that  $\sum_{w \in \chi^{-1}(c) \setminus \{v\}} f_w(\emptyset) = 0$  since we applied the preprocessing from Proposition 2. Therefore, each entry can be computed in  $\mathcal{O}(2k \cdot n^2 \cdot \delta_{\mathcal{F}})$  time by iterating over all  $2k$  possible colors  $c$ , all  $\mathcal{O}(n)$  vertices  $v$  in the corresponding color class, and all  $\mathcal{O}(\delta_{\mathcal{F}} n)$  vertices in possible parent sets of  $v$ . Altogether, COLORED  $(\Pi_0 + e)$ -SKELETON BNSL can be solved in  $\mathcal{O}(4^k k^2 n^2 \delta_{\mathcal{F}})$  time.  $\square$

Propositions 24 and 25 give the following.

**Theorem 26** There exists a randomized algorithm for  $(\Pi_0 + e)$ -SKELETON BNSL that, in  $\mathcal{O}((2e)^{2k} \cdot k^2 n^2 \delta_{\mathcal{F}})$  time returns no, if given a no-instance and returns yes with probability at least  $1 - \frac{1}{e}$ , if given a yes-instance.

PROOF *Algorithm.* We describe the randomized algorithm applied on an instance  $I = (N, \mathcal{F}, t, k)$ . Repeat the following two steps  $e^{2k}$  times independently:

1. Color every vertex of  $N$  independently with one color from the set  $\{1, \dots, 2k\}$  with uniform probability. Let  $\chi : N \rightarrow \{1, \dots, 2k\}$  be the resulting coloring.



2. Apply the algorithm behind Proposition 25 to decide if  $(N, \mathcal{F}, t, k, \chi)$  is a yes-instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL. If this is the case, then return *yes*.

If for none of the  $e^{2k}$  applications the answer *yes* was returned in Step 2, then return *no*.

*Running Time.* We first consider the running time of the algorithm. By Proposition 25, one application of the algorithm described above can be performed in  $\mathcal{O}(2^{2k} \cdot k^2 n^2 \delta_{\mathcal{F}})$  time. Thus, the overall running time of the algorithm is  $\mathcal{O}((2e)^{2k} \cdot k^2 n^2 \delta_{\mathcal{F}})$  as claimed.

*Error Probability.* We next consider the error probability of the algorithm. Given a no-instance, there exists no  $(N, \mathcal{F}, t)$ -valid arc set  $A$  with  $|A| \leq k$  and therefore the answer *no* is always returned in Step 2. Conversely, given a yes-instance  $I$ , we conclude from Proposition 24 that there exist at least  $(2k)!(2k)^{(n-2k)}$  colorings  $\chi$  such that  $(N, \mathcal{F}, t, k, \chi)$  is a yes-instance of COLORED  $(\Pi_0 + e)$ -SKELETON BNSL. As explained in the textbook by Cygan et al. (2015), the probability of randomly choosing such a coloring  $\chi$  is at least

$$\frac{(2k)!(2k)^{(n-2k)}}{(2k)^n} \geq e^{-2k}.$$

Since we repeat the algorithm independently  $e^{2k}$  times, the probability that *no* is returned is at most

$$(1 - e^{-2k})^{e^{2k}} \leq (e^{-e^{-2k}})^{e^{2k}} = \frac{1}{e},$$

where the first inequality relies on the inequality  $1 + x \leq e^x$ . Consequently, our algorithm returns *yes* with probability at least  $1 - \frac{1}{e}$ .  $\square$

The randomized algorithm from Theorem 26 can be derandomized with standard techniques (Naor, Schulman, & Srinivasan, 1995; Cygan et al., 2015).

**Corollary 27**  $(\Pi_0 + e)$ -SKELETON BNSL can be solved in  $(2e)^{2k} \cdot k^{\mathcal{O}(\log(k))} \cdot |I|^{\mathcal{O}(1)}$  time.

Bounding the number of arcs appears to be not so relevant for practical use. However, the algorithm might be useful as a heuristic upper bound: If we want to add a restricted number of dependencies to a given Bayesian network, the result of  $(\Pi_0 + e)$ -SKELETON BNSL gives an upper bound for the profit we can expect from that modification.

## 6.2 $(\Pi_0 + e)$ -Moral BNSL

We now study a version of BNSL, where we aim to learn a network whose moralized graph has a bounded number of edges. Formally, this is the constrained BNSL problem  $(\Pi_0 + e)$ -MORAL BNSL, where  $\Pi_0$  is the class of edgeless graphs.

Note that, given some  $k \in \mathbb{N}$ , a DAG  $D = (N, A)$  with  $\mathcal{M}(D) \in (\Pi_0 + ke)$  is a DAG whose moralized graph contains at most  $k$  edges. Observe that there is a simple XP-time algorithm that solves  $(\Pi_0 + e)$ -MORAL BNSL when parameterized by  $k$ : Let  $I = (N, \mathcal{F}, t, k)$  be an instance of  $(\Pi_0 + e)$ -MORAL BNSL. If  $(N, A)$  is a Bayesian network whose moralized graph has  $k$  or less edges, then  $|A| \leq k$ . We can find  $A$  by iterating over all  $\mathcal{O}(n^{2k})$  possible arc sets  $A$  with  $|A| \leq k$ . If we consider the directed superstructure  $S_{\mathcal{F}} = (N, A_{\mathcal{F}})$  we can instead iterate over all possible subsets  $A' \subseteq A_{\mathcal{F}}$  with  $|A'| \leq k$ . Afterwards, we check if  $A'$  is  $(N, \mathcal{F}, t)$ -valid and if  $\mathcal{M}(N, A') \in \Pi_0 + ke$ . This implies the following.

**Proposition 28**  $(\Pi_0 + e)$ -BNSL can be solved in  $m^k \cdot |I|^{\mathcal{O}(1)}$  time.

To put this simple XP-time algorithm into context, we show that  $(\Pi_0 + e)$ -BNSL is W[1]-hard when parameterized by  $t + k$ . Hence, there is little hope to solve  $(\Pi_0 + e)$ -BNSL in time  $g(t + k) \cdot |I|^{\mathcal{O}(1)}$  for any computable function  $k$ .

**Theorem 29**  $(\Pi_0 + e)$ -MORAL BNSL is W[1]-hard when parameterized by  $t + k$ , even when  $S_{\mathcal{F}}$  is a DAG, the maximum parent set size is 3, and every local score is either 1 or 0.

**PROOF** We prove W[1]-hardness by giving a parameterized reduction from the following problem.

MULTICOLORED CLIQUE

**Input:** A properly  $\ell$ -colored undirected graph  $G = (V, E)$  with color classes  $C_1, \dots, C_\ell \subseteq V$ .

**Question:** Is there a clique containing one vertex from each color class in  $G$ ?

MULTICOLORED CLIQUE is W[1]-hard when parameterized by  $\ell$  (Pietrzak, 2003; Fellows, Hermelin, Rosamond, & Vialette, 2009).

*Construction.* Let  $G = (V, E)$  be a properly  $\ell$ -colored undirected graph with color classes  $C_1, \dots, C_\ell$ . We describe how to construct an equivalent instance  $I = (N, \mathcal{F}, t, k)$  of  $(\Pi_0 + e)$ -MORAL BNSL from  $G$ . First, we define the vertex set  $N$ . Every vertex  $v \in V$  becomes a vertex in  $N$  and for every pair  $\{C_i, C_j\}$  ( $i \neq j$ ) of color classes we add a vertex  $w_{\{i,j\}}$  to  $N$ . Let  $W$  be the set of all such vertices  $w_{\{i,j\}}$ . Moreover, we add a vertex  $x$  to  $N$  which we will call the *central vertex* for the rest of the proof.

Second, we define the local scores  $\mathcal{F}$ . For every vertex  $u \in V \cup \{x\}$  and every  $P \subseteq N \setminus \{u\}$ , we set  $f_u(P) := 0$ . It remains to define the local scores for the vertices in  $W$ . Let  $i, j \in \{1, \dots, \ell\}$  with  $i \neq j$ . We set  $f_{w_{\{i,j\}}}(\{u, v, x\}) := 1$  if there is an edge  $\{u, v\} \in E$  connecting a vertex  $u \in C_i$  and  $v \in C_j$ . For all other sets  $P$ , we set  $f_{w_{\{i,j\}}}(P) := 0$ . Observe that the value of the local scores is either 0 or 1 and that there are exactly  $|E|$  values with  $f_v(P) = 1$ .

Finally, we set  $t := \binom{\ell}{2}$  and  $k := 4\binom{\ell}{2} + \ell$ . Note that  $t + k \in \mathcal{O}(\ell^2)$ .

Observe that the maximum parent set size is 3 and the directed superstructure  $S_{\mathcal{F}}$  is a DAG since every vertex in  $V \cup \{x\}$  has in-degree 0 in  $S_{\mathcal{F}}$  and every vertex in  $W$  is a sink in  $S_{\mathcal{F}}$ . Figure 3 shows an example of the construction.

*Intuition.* Before we show the correctness of the reduction, we start with some intuition. To obtain score  $t = \binom{\ell}{2}$ , every vertex in  $W$  must choose a parent set with score 1. Hence, every  $w_{\{i,j\}}$  chooses a parent set  $\{u, v, x\}$  with  $u \in C_i$  and  $v \in C_j$ . This choice represents the choice of an edge  $\{u, v\} \in E$  between the vertices  $u$  and  $v$  of a multicolored clique in  $G$ . Considering the moralized graph of the resulting Bayesian network, the bound of the number of edges gives a bound of the number of moral edges that are incident with the central vertex  $x$ . This guarantees that the chosen edges form a multicolored clique in the following sense: If the parent sets of vertices in  $W$  do not correspond to the edges of a multicolored clique in  $G$ , then the moralized graph has more than  $k = 4\binom{\ell}{2} + \ell$  edges.

*Correctness.* We show that  $G$  is a yes-instance of MULTICOLORED CLIQUE if and only if  $(N, \mathcal{F}, t, k)$  is a yes-instance of  $(\Pi_0 + e)$ -MORAL BNSL.

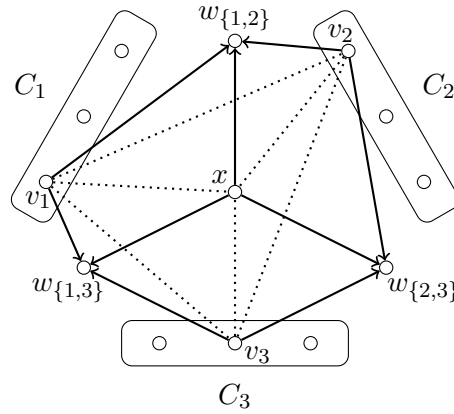


Figure 3: An example of the construction given in the proof of Theorem 29. The original instance contains a multicolored clique on the vertices  $v_1 \in C_1$ ,  $v_2 \in C_2$ , and  $v_3 \in C_3$ . The directed edges represent the arcs of a DAG with score 3 such that the moralized graph contains 15 edges. The dotted edges correspond to the moralized edges.

( $\Rightarrow$ ) Let  $S := \{v_1, \dots, v_\ell\}$  with  $v_i \in C_i$  be a multicolored clique in  $G$ . We define the arc set  $A := \{(v_i, w_{\{i,j\}}), (v_j, w_{\{i,j\}}), (x, w_{\{i,j\}}) \mid w_{\{i,j\}} \in W\}$ . We show that  $A$  is  $(N, \mathcal{F}, t)$ -valid and that there are at most  $k$  edges in  $\mathcal{M}(N, A)$ .

Since the vertices of  $S$  are pairwise adjacent in  $G$ , it holds that  $f_{w_{\{i,j\}}}(P_{w_{\{i,j\}}}^A) = 1$  for every  $w_{\{i,j\}} \in W$  and therefore  $\sum_{v \in N} f_v(P_v^A) = \binom{\ell}{2} = t$ . Moreover, since  $S_{\mathcal{F}}$  is a DAG, we conclude that  $(N, A)$  is a DAG. Hence,  $A$  is  $(N, \mathcal{F}, t)$ -valid.

It remains to check that there are at most  $k = 4 \cdot \binom{\ell}{2} + \ell$  edges in  $\mathcal{M}(N, A)$ . First, we consider the number of arcs in  $(N, A)$ . Since every vertex in  $W$  has three parents in  $(N, A)$ , we conclude  $|A| = 3 \cdot \binom{\ell}{2}$ . Next, we consider the moral edges in  $\mathcal{M}(N, A)$ . Let  $a, b \in N$  be two vertices that have a common child in  $(N, A)$ . Observe that all vertices in  $V \setminus S \cup W$  have out-degree 0 in  $(N, A)$ . We conclude  $a, b \in S \cup \{x\}$ . Then, there are at most  $|\{\{a, b\} \mid a \in S, b \in S \cup \{x\}\}| = \binom{\ell}{2} + \ell$  moral edges. Hence, there are at most  $k = 4 \cdot \binom{\ell}{2} + \ell$  edges in  $\mathcal{M}(N, A)$ .

( $\Leftarrow$ ) Conversely, let  $A \subseteq N \times N$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that  $\mathcal{M}(N, A)$  contains at most  $k = 4 \cdot \binom{\ell}{2} + \ell$  edges. We show that there exists a multicolored clique  $S$  in  $G$ .

Since  $A$  is  $(N, \mathcal{F}, t)$ -valid, we know that  $\sum_{v \in N} f_v(P_v^A) = \binom{\ell}{2}$  and thus  $f_{w_{\{i,j\}}}(P_{w_{\{i,j\}}}^A) = 1$  for every  $w_{\{i,j\}} \in W$ . By the construction of  $\mathcal{F}$  this implies  $|P_{w_{\{i,j\}}}^A| = 3$  for every  $w_{\{i,j\}} \in W$ . We conclude  $|A| = 3 \cdot \binom{\ell}{2}$ . Hence, there are at most  $\binom{\ell}{2} + \ell$  moral edges in  $\mathcal{M}(N, A)$ .

Before we define the multicolored clique  $S$ , we take a closer look at the moral edges that are incident with vertices of the color classes  $C_1, \dots, C_\ell$ . Let  $C_i$  and  $C_j$  be distinct color classes. Then, since  $P_{w_{\{i,j\}}}^A$  contains one vertex from  $C_i$  and one vertex from  $C_j$ , there exists a moral edge between the vertices of  $C_i$  and  $C_j$ . Hence, there are at least  $\binom{\ell}{2}$  moral edges between the color classes of  $C_1, \dots, C_\ell$ . Now, since the overall number of moral edges in  $\mathcal{M}(N, A)$  is at most  $\binom{\ell}{2} + \ell$ , we may conclude that there are at most  $\ell$  moral edges that

are incident with the central vertex  $x$ . We use the following claim to define a multicolored clique  $S$  in  $G$ .

**Claim 3** *For every color class  $C_i$  it holds that  $|E_{\mathcal{M}(N,A)}(C_i, \{x\})| = 1$ .*

PROOF Let  $C_i$  be a color class. Note that there is no arc in  $A$  connecting  $x$  with some vertices in  $C_i$ . So,  $E_{\mathcal{M}(N,A)}(C_i, \{x\})$  contains only moral edges. For every  $j \in \{1, \dots, \ell\}$  with  $j \neq i$ , the vertex  $w_{\{i,j\}}$  has a parent set  $P_{w_{\{i,j\}}}^A$  containing some  $v \in C_i$ ,  $u \in C_j$  and  $x$ . Then, there exist moral edges  $\{u, v\}$ ,  $\{v, x\}$ , and  $\{u, x\}$ . Therefore, every color class contains a vertex that is adjacent to  $x$  by a moral edge. Since there are at most  $\ell$  moral edges incident with  $x$ , we conclude  $|E_{\mathcal{M}(N,A)}(C_i, \{x\})| = 1$ .  $\diamond$

We define  $S := \{v_1, v_2, \dots, v_\ell\}$ , where  $(v_i, x)$  is the unique element of  $E_{\mathcal{M}(N,A)}(C_i, \{x\})$ . Observe that this implies  $P_{w_{\{i,j\}}}^A = \{v_i, v_j, x\}$  for all  $v_i, v_j \in S$  with  $i \neq j$ . We show that  $S$  is a multicolored clique in  $G$ . Obviously, the vertices of  $S$  are elements of distinct color classes. Thus, it remains to show that the vertices in  $S$  are pairwise adjacent in  $G$ . Let  $v_i, v_j \in S$  with  $i \neq j$ . Then,  $P_{w_{\{i,j\}}}^A = \{v_i, v_j, x\}$  and since  $f_{w_{\{i,j\}}}(P_{w_{\{i,j\}}}^A) = 1$  it follows from the construction of  $\mathcal{F}$  that there is an edge  $\{v_i, v_j\} \in E$ . Hence,  $S$  is a multicolored clique in  $G$ .  $\square$

## 7. BNSL with Bounded Feedback Edge Set

In this section, we provide a first step into the study of the parameterized complexity of learning a Bayesian network whose moralized graph has a feedback edge set of bounded size. Formally, this is the constrained BNSL problem  $(\Pi_F + e)$ -MORAL BNSL, where  $\Pi_F$  is the class of forest, which are undirected acyclic graphs. Recall that for efficient inference it is desirable to have a small treewidth in the moralized graph (Darwiche, 2009). As all other parameters considered in this work, the size of a feedback edge set is an upper bound for the treewidth. Thus, learning a network where the moralized graph has a bounded feedback edge set is motivated from a practical point of view.

Before we consider  $(\Pi_F + e)$ -MORAL BNSL, we briefly discuss  $(\Pi_F + e)$ -SKELETON BNSL. When  $k = 0$ , this is the problem of learning a Bayesian network with an acyclic skeleton, also known as polytree. Finding an optimal polytree is NP-hard even on instances with maximum parent set size 2 (Dasgupta, 1999). Consequently,  $(\Pi_F + e)$ -SKELETON BNSL is NP-hard even if  $k = 0$  is fixed. In contrast, the case  $k = 0$  can be solved efficiently if we consider the moralized graph instead of the skeleton. This can be seen as follows. Let  $D := (N, A)$  be a DAG such that  $\mathcal{M}(D)$  is acyclic. Then, each  $v \in N$  has at most one parent in  $D$ , since otherwise  $\mathcal{M}(D)$  contains a triangle. Thus,  $D := (N, A)$  is a branching. Consequently,  $(\Pi_F + e)$ -MORAL BNSL with  $k = 0$  can be solved by computing an optimal branching which can be done in polynomial time (Chow & Liu, 1968; Gaspers et al., 2015).

**Proposition 30**  *$(\Pi_F + e)$ -MORAL BNSL can be solved in polynomial time when limited to instances with  $k = 0$ .*

This positive result makes it interesting to study the parameterized complexity of  $(\Pi_F + e)$ -MORAL BNSL when parameterized by  $k$ . In the following, we provide a first step into

this parameterized complexity analysis and show that  $(\Pi_F + e)$ -MORAL BNSL is  $W[1]$ -hard when parameterized by  $k$ . Thus,  $(\Pi_F + e)$ -MORAL BNSL can presumably not be solved in  $g(k) \cdot |I|^{\mathcal{O}(1)}$  time for a computable function  $g$ . However, an XP-time algorithm might still be possible.

**Theorem 31**  $(\Pi_F + e)$ -MORAL BNSL is  $W[1]$ -hard when parameterized by  $k$ , even when restricted to instances where  $S_{\bar{\mathcal{F}}}$  is a DAG and the maximum parent set size is 4.

**PROOF** We give a parameterized reduction from  $(\Pi_0 + e)$ -MORAL BNSL parameterized by the number of edges  $k$  which is  $W[1]$ -hard even on instances where the directed superstructure is a DAG and the maximum parent set size is 3 due to Theorem 29.

*Construction.* Let  $I := (N, \mathcal{F}, t, k)$  be such an instance of  $(\Pi_0 + e)$ -MORAL BNSL. We describe how to construct an equivalent instance  $I' := (N', \mathcal{F}', t', k')$  of  $(\Pi_F + e)$ -MORAL BNSL where  $k' = k$ . The vertex set is  $N' := N \cup \{x\}$  for some  $x \notin N$ .

We define the vertex set by  $N' := N \cup \{x\}$  for some  $x \notin N$ . To define the local scores  $\mathcal{F}'$ , we set  $\ell^+ := 1 + \sum_{v \in N} \max_{P \subseteq N \setminus \{v\}} f_v(P)$ . For every  $v \in N$  we set  $f'_v(P) = f_v(P \setminus \{x\}) + \ell^+$  if  $x \in P$  and  $P \setminus \{x\} \in \mathcal{P}_{\mathcal{F}}(v)$ . In all other cases, we set  $f'_v(P) = 0$ . For the vertex  $x$ , we set  $f'_x(P) = 0$  for every  $P$ . Finally, we set  $t' := t + n \cdot \ell^+$ .

We can obviously compute  $I'$  from  $I$  in polynomial time. Since  $I$  is an instance where  $S_{\bar{\mathcal{F}}}$  is a DAG and the maximum parent set size is 3, we conclude that the maximum parent set size of  $I'$  is 4 and that  $S_{\bar{\mathcal{F}'}}$  is a DAG.

*Intuition.* Before we prove the correctness of the reduction we provide some intuition. To obtain an  $(N', \mathcal{F}', t')$ -valid arc set  $A'$ , the vertex  $x$  must be a parent of every vertex of  $N$ . Hence, for every  $v \in N$ , there exists an edge  $\{x, v\}$  in  $\mathcal{M}(N', A')$ . The idea is that  $\mathcal{M}(N', A')$  can be transformed into an acyclic graph by deleting all edges between the vertices of  $N$ .

*Correctness.* We now prove that  $I$  is a yes-instance of  $(\Pi_0 + e)$ -MORAL BNSL if and only if  $I'$  is a yes-instance of  $(\Pi_F + e)$ -MORAL BNSL.

$(\Rightarrow)$  Let  $A \subseteq N \times N$  be an  $(N, \mathcal{F}, t)$ -valid arc set such that  $\mathcal{M}(D)$  for  $D := (N, A)$  contains at most  $k$  edges. We then define  $A' := A \cup \{(x, v) \mid v \in N\}$  and let  $D' := (N', A')$ . We show that  $A'$  is  $(N', \mathcal{F}', t')$ -valid and  $\mathcal{M}(D')$  has a feedback edge set of size at most  $k$ .

We first show that  $A'$  is  $(N', \mathcal{F}', t')$ -valid. Since  $S_{\bar{\mathcal{F}'}}$  is a DAG we conclude that  $D$  is a DAG. Moreover,  $P_v^{A'} = P_v^A \cup \{x\}$  for every  $v \in N$  and therefore

$$\begin{aligned} \sum_{v \in N'} f'_v(P_v^{A'}) &= \sum_{v \in N} (f_v(P_v^A) + \ell^+) \\ &\geq t + n \cdot \ell^+ = t'. \end{aligned}$$

Consequently,  $D'$  is  $(N', \mathcal{F}', t')$ -valid. It remains to show that  $\mathcal{M}(D')$  has a feedback edge set of size at most  $k$ . To this end, consider the following claim.

**Claim 4** Let  $v, w \in N$ . Then,  $\{v, w\}$  is a moral edge in  $\mathcal{M}(D)$  if and only if  $\{v, w\}$  is a moral edge in  $\mathcal{M}(D')$ .

**PROOF** Let  $\{v, w\}$  be a moral edge in  $\mathcal{M}(D)$ . Then, there exists a vertex  $u \in N$  such that  $(v, u), (w, u) \in A$ . Since  $A \subseteq A'$  we conclude that  $\{v, w\}$  is a moral edge in  $\mathcal{M}(D')$ .

Conversely, let  $\{v, w\}$  be a moral edge in  $\mathcal{M}(D')$ . Then,  $v$  and  $w$  have a common child  $u$  in  $D'$ . Since  $x$  has no incoming arcs, we conclude  $u \in N$  and therefore  $(v, u), (w, u) \in A$ . Hence,  $\{v, w\}$  is a moral edge in  $\mathcal{M}(D)$ .  $\diamond$

Claim 4 together with the fact that  $(N \times N) \cap A' = A$  implies that  $v, w \in N$  are adjacent in  $\mathcal{M}(D)$  if and only if they are adjacent in  $\mathcal{M}(D')$ . Hence, if we delete every edge of  $\mathcal{M}(D)$  from  $\mathcal{M}(D')$  we obtain the graph  $G := (N', \{\{x, v\} \mid v \in N\})$  which is acyclic. Since there are at most  $k$  edges in  $\mathcal{M}(D)$  we conclude that there exists a feedback edge set of size at most  $k$  for  $\mathcal{M}(D')$ .

( $\Leftarrow$ ) Conversely, let  $A'$  be an  $(N', \mathcal{F}', t')$ -valid arc set such that  $\mathcal{M}(D')$  for  $D' := (N', A')$  has a feedback edge set of size at most  $k$ . Observe that  $x \in P_v^{A'}$  for every  $v \in N$ : If there exists a vertex  $w \in N$  with  $x \notin P_w^{A'}$ , then  $f'_w(P_w^{A'}) = 0$  and therefore the sum of the local scores is smaller than  $n \cdot \ell^+$ . This contradicts the fact that  $A'$  is  $(N', \mathcal{F}', t')$ -valid. We define  $A := (N \times N) \cap A'$ . Note that  $P_v^A = P_v^{A'} \setminus \{x\}$  for every  $v \in N$ .

We first show that  $D := (N, A)$  is  $(N, \mathcal{F}, t)$ -valid. Obviously,  $D$  is a DAG since  $S_{\mathcal{F}}$  is a DAG. Moreover, it holds that

$$\begin{aligned} \sum_{v \in N} f_v(P_v^A) &= \sum_{v \in N} f_v(P_v^{A'} \setminus \{x\}) \\ &\geq t' - n \cdot \ell^+ = t. \end{aligned}$$

Consequently,  $D$  is  $(N, \mathcal{F}, t)$ -valid.

Next, assume towards a contradiction that there are more than  $k$  edges in  $\mathcal{M}(D)$ . Since  $A \subseteq A'$ , this implies that in  $\mathcal{M}(D')$  there are more than  $k$  edges between the vertices of  $N$ . Furthermore, since  $x \in P_v^{A'}$  for every  $v \in N$  we conclude that every vertex in  $N$  is adjacent to  $x$  in  $\mathcal{M}(D')$ . Hence,  $\mathcal{M}(D')$  consists of  $n + 1$  vertices and at least  $n + k + 1$  edges which contradicts the fact that  $\mathcal{M}(D')$  has a feedback edge set of size at most  $k$ .  $\square$

## 8. On Problem Kernelization

In this section we prove a new hardness result for VANILLA-BNSL: We show that under the standard assumption  $\text{NP} \not\subseteq \text{coNP}/\text{poly}$ , VANILLA-BNSL does not admit a polynomial problem kernel when parameterized by the number of vertices. That is, it is presumably impossible to transform an instance of VANILLA-BNSL in polynomial time into an equivalent instance of size  $|I| = n^{\mathcal{O}(1)}$ . Thus, it is sometimes necessary to keep an exponential number of parent scores to compute an optimal network. The kernel lower bound even holds for instances where all local scores are either 0 or 1. Thus, the kernel lower bound is not based on the fact that large local scores might be incompressible.

We then use the kernel lower bound for VANILLA-BNSL to complement the FPT result from Corollary 27 and show that there is little hope that  $(\Pi_0 + e)$ -SKELETON BNSL admits a polynomial problem kernel.

**Theorem 32** *VANILLA-BNSL parameterized by  $n + t$  does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$  even when restricted to instances where all local scores are either 0 or 1.*

PROOF We prove the theorem by giving a polynomial parameter transformation from the following problem.

MULTICOLORED INDEPENDENT SET

**Input:** A properly  $\ell$ -colored undirected graph  $G = (V, E)$  with color classes  $C_1, \dots, C_\ell \subseteq V$ .

**Question:** Is there an independent set containing one vertex from each color in  $G$ ?

MULTICOLORED INDEPENDENT SET does not admit a polynomial kernel when parameterized by  $|C_1 \cup \dots \cup C_{\ell-1}|$  unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$  (Grüttemeier & Komusiewicz, 2020).

*Construction.* Let  $G = (V, E)$  be an instance of MULTICOLORED INDEPENDENT SET with the color classes  $C_1, \dots, C_\ell$ . We describe how to construct an equivalent instance  $(N, \mathcal{F}, t)$  of VANILLA-BNSL. First, set  $N := C_1 \cup C_2 \cup \dots \cup C_{\ell-1} \cup \{x\}$  for some  $x \notin V$ . Second, we define the local scores  $\mathcal{F}$  as follows: Let  $i \in \{1, \dots, \ell - 1\}$ . For  $v \in C_i$  we set  $f_v(P) = 1$  if  $P = (C_i \setminus \{v\}) \cup (N_G(v) \setminus C_\ell) \cup \{x\}$ . Otherwise, we set  $f_v(P) = 0$ . For  $x$ , we set  $f_x(P) = 1$  if there exists some  $w \in C_\ell$  with  $N_G(w) = P$ . Otherwise we set  $f_x(P) = 0$ . Finally, we set  $t := \ell$ .

Observe that the value of the local scores is either 1 or 0, and that there are exactly  $|V|$  values where  $f_v(P) = 1$ . Hence,  $|\mathcal{F}| \in \mathcal{O}(|V|)$ . We can obviously compute  $(N, \mathcal{F}, t)$  in polynomial time from  $G$ . Furthermore, recall that  $|N| = |C_1 \cup \dots \cup C_{\ell-1}| + 1$  and therefore,  $n + t$  is polynomially bounded in  $|C_1 \cup \dots \cup C_{\ell-1}|$ .

*Intuition:* Before we show the correctness of the polynomial parameter transformation, we start with some intuition. To reach the score  $t = \ell$ , exactly one vertex per color class  $C_1, \dots, C_{\ell-1}$  and the vertex  $x$  must learn a parent set with score 1. The vertices from  $C_1, \dots, C_{\ell-1}$  and the choice of the parent set of  $x$  then correspond to a multicolored set in  $G$ . The condition that the resulting directed graph must be a DAG guarantees that the chosen vertices form an independent set.

*Correctness.* ( $\Rightarrow$ ) Let  $S = \{v_1, \dots, v_\ell\}$  be a multicolored independent set in  $G$  with  $v_i \in C_i$  for all  $i \in \{1, \dots, \ell\}$ . We define the arc set  $A$  by defining the parent sets of all vertices in  $N$ : For all  $v \in N \setminus \{v_1, \dots, v_{\ell-1}\}$  we set  $P_v^A := \emptyset$ . Next, for  $v_i \in \{v_1, \dots, v_{\ell-1}\}$  we set  $P_{v_i}^A := (C_i \setminus \{v_i\}) \cup (N_G(v_i) \setminus C_\ell) \cup \{x\}$ . Finally, we set  $P_x^A = N_G(v_\ell)$ . We now prove that  $A$  is  $(N, \mathcal{F}, t)$ -valid. By definition of  $\mathcal{F}$  it holds that  $f_v(P_v^A) = 1$  for every  $v \in \{v_1, \dots, v_{\ell-1}, x\}$ . Hence,  $\sum_{c \in N} f_c(P_c^A) = t$ .

It remains to show that  $D := (N, A)$  is a DAG. If  $D$  contains a directed cycle, all vertices on the directed cycle have incoming and outgoing arcs. Observe that  $v_1, \dots, v_{\ell-1}$ , and  $x$  are the only vertices with incoming arcs.

We first prove that every  $v \in \{v_1, \dots, v_{\ell-1}\}$  is a sink in  $D$ . Assume towards a contradiction that there is some  $v \in \{v_1, \dots, v_{\ell-1}\}$  that has an outgoing arc  $(v, w) \in A$ . Without loss of generality, let  $v = v_1$ . Since  $v_1 \notin P_{v_1}^A$  and only the vertices  $v_1, v_2, \dots, v_{\ell-1}, x$  have parents under  $A$ , we conclude  $w \in \{v_2, \dots, v_{\ell-1}, x\}$ . If  $w \in \{v_2, \dots, v_{\ell-1}\}$ , then  $v \in P_w^A$  and therefore  $v \in N_G(w)$ . Otherwise, if  $w = x$ , then  $v \in P_x^A$  and therefore  $v \in N_G(v_\ell)$ . Both cases contradict the fact that  $S$  is an independent set in  $G$  and therefore every  $v \in \{v_1, \dots, v_{\ell-1}\}$  is a sink in  $D$ .

We conclude that  $x$  is the only vertex that might have incoming and outgoing arcs in  $D$ . Hence,  $x$  is the only vertex that might be part of a directed cycle. Since  $x \notin P_x^A$  we conclude that there is no cycle in  $D$  and therefore  $D$  is a DAG.

( $\Leftarrow$ ) Let  $A$  be an  $(N, \mathcal{F}, t)$ -valid arc set. We show that there exists a multicolored independent set  $S$  in  $G$ . To this end, consider the following claim.

**Claim 5**

- a) *There are at least  $\ell$  vertices  $v \in N$  with  $f_v(P_v^A) = 1$ .*
- b) *For every  $C_i$  with  $i \in \{1, \dots, \ell - 1\}$  there is at most one vertex  $v_i \in C_i$  with  $f_{v_i}(P_{v_i}^A) = 1$ .*

PROOF We first show statement a). Since  $A$  is  $(N, \mathcal{F}, t)$ -valid we know that  $\sum_{v \in V} f_v(P_v^A) \geq t = \ell$ . Since every local score is either 0 or 1, statement a) follows.

We next show statement b). Note that  $(N, A)$  is a DAG since  $A$  is  $(N, \mathcal{F}, t)$ -valid. Let  $i \in \{1, \dots, \ell - 1\}$ . Assume towards a contradiction that there are distinct  $u, v \in C_i$  with  $f_u(P_u^A) = f_v(P_v^A) = 1$ . Then, by the construction of  $\mathcal{F}$  we conclude  $C_i \setminus \{u\} \subseteq P_u^A$  and  $C_i \setminus \{v\} \subseteq P_v^A$ . Hence  $(u, v), (v, u) \in A$  which contradicts the fact that  $(N, A)$  is a DAG. Consequently, statement b) holds.  $\diamond$

From Claim 5 a) and b) we conclude that for each  $C_i$  with  $i \in \{1, \dots, \ell - 1\}$  there is exactly one  $v_i \in C_i$  with  $f_{v_i}(P_{v_i}^A) = 1$  and that  $f_x(P_x^A) = 1$ . Moreover,  $f_x(P_x^A) = 1$  implies that there exists a vertex  $v_\ell \in C_\ell$  with  $N_G(v_\ell) = P_x^A$ . We define  $S := \{v_1, \dots, v_{\ell-1}, v_\ell\}$  and show that  $S$  is a multicolored independent set in  $G$ .

Obviously, the vertices of  $S$  are from pairwise distinct color classes. Thus, it remains to show that no two vertices in  $S$  are adjacent in  $G$ . Assume towards a contradiction that there exist  $v, w \in S$  such that  $\{v, w\} \in E$ . Without loss of generality, let  $v = v_1$ . Consider the following cases.

**Case 1:**  $w \in \{v_2, \dots, v_{\ell-1}\}$ . Then,  $\{v, w\} \in E$  implies  $w \in N_G(v) \setminus C_\ell$  and  $v \in N_G(w) \setminus C_\ell$ . Together with the fact that  $f_v(P_v^A) = f_w(P_w^A) = 1$  we conclude  $(v, w), (w, v) \in A$  which contradicts the fact that  $(N, A)$  is a DAG.

**Case 2:**  $w = v_\ell$ . Then,  $\{v, w\} \in E$  implies  $v \in N_G(w)$  and therefore  $v \in P_x^A$ . Moreover  $f_v(P_v^A) = 1$  implies  $x \in P_v^A$ . Hence,  $(v, w), (w, v) \in A$  contradicting the fact that  $(N, A)$  is a DAG.

We conclude that no two vertices of  $S$  are adjacent in  $G$  and therefore,  $S$  is a multicolored independent set in  $G$ .  $\square$

We may also use Theorem 32 to complement the FPT result for  $(\Pi_0 + e)$ -SKELETON BNSL by a kernel lower-bound. More generally, we show that all four constrained BNSL problems for any monotone and infinite graph class  $\Pi$  have no polynomial kernel when parameterized by  $k + t$  unless  $\text{NP} \subseteq \text{coNP/poly}$ .

Let  $\Pi$  be a monotone graph class that contains infinitely many graphs and we consider the constrained BNSL problem  $(\Pi + e)$ -SKELETON BNSL. We now observe that there is a simple polynomial parameter transformation from VANILLA-BNSL parameterized by  $n + t$  to  $(\Pi + e)$ -SKELETON BNSL parameterized by  $k + t$ : Let  $I := (N, \mathcal{F}, t)$  be an instance of VANILLA-BNSL. We then set  $k := n^2$  and construct an instance  $I' := (N, \mathcal{F}, t, k)$  of  $(\Pi + e)$ -SKELETON BNSL. Since  $k = n^2$ , we have  $S(N, A) \in \Pi + ke$  for every possible  $A \subseteq N \times N$ . Thus,  $I'$  is a yes-instance of  $(\Pi + e)$ -SKELETON BNSL if and only if there exists an  $(N, \mathcal{F}, t)$ -valid arc set  $A \subseteq N \times N$ . Consequently,  $I'$  is a yes-instance of  $(\Pi + e)$ -SKELETON BNSL



if and only if  $I$  is a yes-instance of VANILLA-BNSL. Since  $k + t = n^2 + t$ , the construction described above is a polynomial parameter transformation.

It is easy to see that the same construction also provides a polynomial parameter transformation from VANILLA-BNSL to the other constrained BNSL problems for  $\Pi$ . Together with the kernel lower-bound from Theorem 32, this implies the following.

**Corollary 33** *Let  $\Pi$  be a monotone graph class that contains infinitely many graphs. Then, all four constrained BNSL problems for  $\Pi$  parameterized by  $k + t$  do not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$ .*

## 9. Conclusion

We have outlined the tractability borderline of BAYESIAN NETWORK STRUCTURE LEARNING with respect to several structural constraints on the learned network or on its moralized graph. In particular, we have shown that putting structural sparsity constraints on the moralized graph may make the problem harder than putting similar constraints on the network. This is somewhat counterintuitive since the moralized graph is a supergraph of the underlying undirected graph of the network. It seems interesting to investigate this issue further, that is, to find other structural constraints such that putting these constraints on the network leads to an easier problem than putting them on the moralized graph.

The two most important concrete questions left open by our work are the following. First, can we compute an optimal network where the skeleton has dissociation number at most  $k$  in polynomial time for constant  $k$ ? Second, can we compute an optimal network whose moralized graph has a feedback edge set of size at most  $k$  in polynomial time for constant  $k$ ?

Another important direction for future work is to study how well the algorithms for  $(\Pi_0 + v)$ -SKELETON-BNSL (Theorem 6) and  $(\Pi_1 + v)$ -MORAL-BNSL (Theorem 17) perform on benchmark data sets. This way, one can extend the work of Korhonen and Parviainen (2015) who experimentally evaluated the performance of a similar algorithm for  $(\Pi_0 + v)$ -MORAL BNSL.

As a final remark, the algorithm for learning a Bayesian network with a bounded number of arcs (Theorem 26) seems to have no direct practical applications. It may, however, be useful as subroutines in a branch-and-bound scenario where one may aim to compute upper or lower bounds on the score that can be achieved by adding  $k$  arcs to a network that is currently considered in the search. Thus, it would be interesting to explore variants of BAYESIAN NETWORK STRUCTURE LEARNING where the input contains a partial network and the aim is to extend it. Do the positive results for BAYESIAN NETWORK STRUCTURE LEARNING also hold for this more general problem?

## Acknowledgments

We would like to thank the reviewers of *Journal of Artificial Intelligence Research* for their helpful comments. Furthermore, we would like to thank our colleague Nils Morawietz (Philipps-Universität Marburg) for his helpful discussions that led to the proof of Theorem 7. A preliminary version of this work appeared in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI '20)*, pages 4245–4251.

The full version contains all missing proofs, new results for BNSL with constraints on the skeleton, and an improved algorithm for  $(\Pi_1 + v)$ -MORAL BNSL leading to a slightly better running time.

## References

- Alon, N., Yuster, R., & Zwick, U. (1995). Color-coding. *Journal of the ACM*, 42(4), 844–856.
- Bodlaender, H. L., Thomassé, S., & Yeo, A. (2011). Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35), 4570–4578.
- Chickering, D. M. (1995). Learning Bayesian networks is NP-complete. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Statistics, (AISTATS'95)*, pp. 121–130. Springer.
- Chow, C. K., & Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3), 462–467.
- Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3), 393–405.
- Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized Algorithms*. Springer.
- Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.
- Dasgupta, S. (1999). Learning polytrees. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 134–141. Morgan Kaufmann.
- Downey, R. G., & Fellows, M. R. (1995). Parameterized computational feasibility. In *Feasible mathematics II*, pp. 219–244. Springer.
- Elidan, G., & Gould, S. (2008). Learning bounded treewidth Bayesian networks. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, (NIPS'08)*, pp. 417–424. Curran Associates, Inc.
- Fellows, M. R., Hermelin, D., Rosamond, F. A., & Vialette, S. (2009). On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1), 53–61.
- Gaspers, S., Koivisto, M., Liedloff, M., Ordyniak, S., & Szeider, S. (2015). On finding optimal polytrees. *Theoretical Computer Science*, 592, 49–58.
- Grüttemeier, N., & Komusiewicz, C. (2020). On the relation of strong triadic closure and cluster deletion. *Algorithmica*, 82(4), 853–880.
- Kirkpatrick, D. G., & Hell, P. (1983). On the complexity of general graph factor problems. *SIAM J. Comput.*, 12(3), 601–609.
- Korhonen, J. H., & Parviainen, P. (2013). Exact learning of bounded tree-width Bayesian networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, (AISTATS'13)*, pp. 370–378. JMLR.org.

- Korhonen, J. H., & Parviainen, P. (2015). Tractable Bayesian network structure learning with bounded vertex cover number. In *Proceedings of the Twenty-Eighth Annual Conference on Neural Information Processing Systems, (NIPS'15)*, pp. 622–630. MIT Press.
- Meek, C. (2001). Finding a path is harder than finding a tree. *Journal of Artificial Intelligence Research, 15*, 383–389.
- Micali, S., & Vazirani, V. V. (1980). An  $O(\sqrt{|V||E|})$  Algorithm for finding maximum matching in general graphs. In *Proceedings of the Twenty-First Annual Symposium on Foundations of Computer Science, (FOCS'80)*, pp. 17–27. IEEE Computer Society.
- Naor, M., Schulman, L. J., & Srinivasan, A. (1995). Splitters and near-optimal derandomization. In *Proceedings of the Thirty-Sixth Annual Symposium on Foundations of Computer Science, (FOCS'95)*, pp. 182–191. IEEE Computer Society.
- Ordyniak, S., & Szeider, S. (2013). Parameterized complexity results for exact Bayesian network structure learning. *Journal of Artificial Intelligence Research, 46*, 263–302.
- Ott, S., & Miyano, S. (2003). Finding optimal gene networks using biological constraints. *Genome Informatics, 14*, 124–133.
- Pietrzak, K. (2003). On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences, 67*(4), 757–771.
- Silander, T., & Myllymäki, P. (2006). A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the Twenty-Second Conference in Uncertainty in Artificial Intelligence (UAI'06)*. AUAI Press.