Critical node cut parameterized by treewidth and solution size is W[1]-hard[☆]

Akanksha Agrawal[a,*], Daniel Lokshtanov[a], Amer E. Mouawad[a]

[a] *University of Bergen, Bergen, Norway.*

Abstract

In the Critical Node Cut problem, given an undirected graph G and two non-negative integers k and μ, the goal is to find a set S of exactly k vertices such that after deleting S we are left with at most μ connected pairs of vertices. In 2015, Hermelin et al. studied the aforementioned problem under the framework of parameterized complexity. They considered various natural parameters, namely, the size k of the desired solution, the upper bound μ on the number of remaining connected pairs, the lower bound b on the number of connected pairs to be removed, and the treewidth $\mathsf{tw}(G)$ of the input graph G. For all but one combination of the above parameters, they determined whether Critical Node Cut is fixed-parameter tractable and whether it admits a polynomial kernel. The only question they left open is whether the problem remains fixed-parameter tractable when parameterized by $k + \mathsf{tw}(G)$. We answer this question in the negative via a new problem of independent interest, which we call SumCSP. We believe that SumCSP can be a useful starting point for showing hardness results of the same nature, i.e., when the treewidth of the graph is part of the parameter.

Keywords: parameterized complexity, W[1]-hardness, SumCSP

1. Introduction

Consider the following problem, called Critical Node Cut (or CNC for short). We are given an undirected graph G and two non-negative integers k and μ. The goal is to determine whether there exists a subset of the vertices

[☆] A preliminary version of this manuscript has been accepted for publication at the 43rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2017).
The research leading to these results received funding from the BeHard grant under the recruitment programme of the of Bergen Research Foundation and the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreements no. 306992 (PARAPPROX).

[*] Corresponding author.
Email addresses: akanksha.agrawal@uib.no (Akanksha Agrawal), daniello@ii.uib.no (Daniel Lokshtanov), a.mouawad@uib.no (Amer E. Mouawad)

of G, say S, of size (exactly) k such that, in the graph $G - S$, we are left with at most μ connected pairs of vertices; $G - S$ denotes the graph obtained from G after deleting vertices in S and the edges incident on them. Alternatively, if we let $\mathcal{C}(G - S) = \{C_1, \ldots, C_\ell\}$, for some integer ℓ, denote the set of connected components in $G - S$, the objective is to guarantee that $\sum_{C \in \mathcal{C}(G-S)} \binom{C}{2} \leq \mu$. The CNC problem, having many real-world applications such as controlling the spread of viruses in networks [1], has been investigated from various algorithmic perspectives, e.g., heuristics [2] and approximations algorithms [3]. For $\mu = 0$, CNC is exactly the same as the Vertex Cover problem, therefore, CNC is NP-complete. On the positive side, it is known that CNC can be solved in polynomial time if we restrict the input graph to trees [4]. More generally, for graphs of bounded treewidth, CNC can be solved in $\mathcal{O}(|V(G)|^{\mathsf{tw}(G)+1})$ time [5], where $\mathsf{tw}(G)$ is the treewidth of G. We refer the reader to [1] for a more extensive survey on CNC and its applications.

Table 1: Summary of results due to Hermelin et al. [1].

Parameter				Result	
k	μ	b	$\mathsf{tw}(G)$	FPT	Polynomial kernel
✓				no	no
	✓			no	no
		✓		yes	no
			✓	no	no
✓	✓			yes	yes
✓		✓		yes	no
✓			✓	**open**	no
	✓	✓		yes	yes
	✓		✓	yes	no
		✓	✓	yes	no
✓	✓	✓		yes	yes
✓	✓		✓	yes	yes
✓		✓	✓	yes	no
	✓	✓	✓	yes	yes
✓	✓	✓	✓	yes	yes

Hermelin et al. [1] initiated the study of the parameterized complexity of CNC. In parameterized complexity [6], we are interested in whether the problem can be solved in $f(\kappa) \cdot |V(G)|^{\mathcal{O}(1)}$ time, for various natural parameters κ and some function f. Alternatively, one can also ask whether or not CNC admits a polynomial kernel for parameter κ, i.e., whether there is an algorithm that reduces any instance of CNC in polynomial time to an equivalent instance of size $\kappa^{\mathcal{O}(1)}$. There are quite a few natural choices for κ in this case and the following choices were considered by Hermelin et al. [1].

- The size k of the desired solution.
- The upper bound μ on the number of remaining connected pairs.

- The lower bound b on the number of connected pairs to be removed.
- The treewidth $\mathsf{tw}(G)$ of the input graph G.

For all but one combination of the above parameters, Hermelin et al. determined whether Critical Node Cut is fixed-parameter tractable (FPT) and whether it admits a polynomial kernel. These results are summarized in Table 1.

In this work, we complete the table by showing that CNC is W[1]-hard (or equivalently not likely to be FPT) when parameterized by $k + \mathsf{tw}(G)$. We prove this result via a new problem of independent interest, which we call SumCSP. We believe that SumCSP can be a useful starting point for showing hardness results of the same nature, i.e., when the treewidth of the graph is part of the parameter.

Overview of the reduction. Our starting point is the 4-Regular Partitioned Subgraph Isomorphism (PSI) problem, which is known to be W[1]-hard [7]. The problem is formally defined below.

4-Regular Partitioned Subgraph Isomorphism (PSI)
Input: A 4-regular pattern graph P with $V(P) = \{p_1, p_2, \ldots, p_\ell\}$, a host graph H, and a coloring function $\mathsf{col} : V(H) \to [\ell]$.
Question: Does there exist an injective function $\phi : V(P) \to V(H)$ such that for each $i \in [\ell]$, $\mathsf{col}(\phi(p_i)) = i$ and for each $p_i p_j \in E(P)$, we have $\phi(p_i)\phi(p_j) \in E(H)$?
Parameter: $|V(P)|$.

We reduce PSI to SumCSP, which is formally defined next.

SumCSP
Input: A directed graph D with vertex set $V(D)$ and arc set $A(D)$, vertex weight function $w_V : V(D) \to \mathbb{N}$, arc weight function $w_A : A(D) \to \mathbb{N}$, and a list function $\varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}}$ such that for all $a \in A(D)$, and for all $(x, y) \in \varphi(a)$ we have $x + y = w_A(a)$.
Question: Does there exists a function $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ such that for each $a \in A(D)$, $\rho(a) \in \varphi(a)$ and for each $v \in V(D)$, $\sum_{u\in N^+(v)} \mathsf{fir}(\rho(vu)) + \sum_{u\in N^-(v)} \mathsf{sec}(\rho(uv)) = w_V(v)$, where $\mathsf{fir}((x, y)) = x$ and $\mathsf{sec}((x, y)) = y$?
Parameter: $|A(D)|$.

An illustration of an input instance to SumCSP and a corresponding solution is given in Figure 1. Bodlaender et al. [8] introduced a very closely related problem to show that Planar Capacitated Dominating Set is W[1]-hard. Planar Capacitated Dominating Set was shown to be W[1]-hard via a reduction from an intermediate problem called Planar Arc Supply. The main difference between Planar Arc Supply and SumCSP is the additional constraint we impose using the arc weight function, i.e., the fact that all pairs in $\varphi(a)$, $a \in A(D)$, must sum to $w_A(a)$. This constraint turns out to be crucial for our reduction. Roughly speaking, the reduction from PSI to SumCSP constructs a directed graph D whose structure is more or less similar to the pattern graph

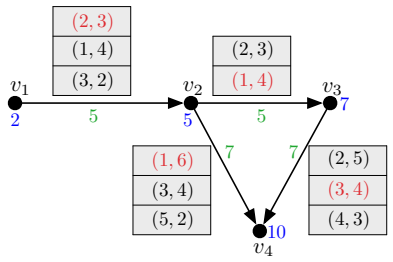

Figure 1: An instance of SumCSP and one corresponding solution. Here, blue and green numbers are vertex and arc weights, respectively. Moreover, the list in black (and red) is a list associated with each arc, and the pairs marked in red correspond to a valid solution.

P (and its size is linear in $|V(P)|$). Edges of H are encoded using the vertex and arc weight functions as well as the function φ. Having established the hardness of SumCSP, we then reduce SumCSP to Critical Note Cut. Let us first state a formal definition of the latter problem.

Critical Node Cut (CNC)
Input: An undirected graph G and integers k and μ.
Question: Does there exist a set $S \subseteq V(G)$ of size (exactly) k such that $\sum_{C \in \mathcal{C}(G-S)} \binom{C}{2} \leq \mu$, where $\mathcal{C}(G-S) = \{C_1, \ldots, C_\ell\}$ denotes the set of connected components in $G - S$?
Parameter: $k + \mathsf{tw}(G)$.

As stated earlier, our reduction from SumCSP to CNC heavily relies on the arc weight function. Another crucial ingredient is the following proposition (which follows by the convexity of $\frac{x(x-1)}{2}$).

Proposition 1. *Let $x_1, \ldots, x_k$ be non-negative integers and let $x_1 + \ldots + x_k = kn$. Then, $\sum_{i=1}^{i=k} \binom{x_i}{2}$ is minimized if $x_i = n$, for all i. In other words, $\sum_{i=1}^{i=k} \binom{x_i}{2}$ is minimized if $\sum_{i=1}^{i=k} \binom{x_i}{2} = k\binom{n}{2}$.*

At a very high level, starting from an instance of SumCSP, we create a graph G (of bounded treewidth) where an optimal solution for CNC must separate the graph into a fixed number of connected components, all having the same size.

2. Preliminaries

We denote the set of natural numbers by $\mathbb{N}$. For $k \in \mathbb{N}$, by $[k]$ we denote the set $\{1, 2, \ldots, k\}$. For sets X, Y, by $X \times Y$ we denote the set $\{(x, y) \mid x \in X, y \in Y\}$. Furthermore, for $(x, y) \in X \times Y$, we let $\mathsf{fir}((x, y)) = x$ and $\mathsf{sec}((x, y)) = y$, i.e., the first and second coordinate of the (ordered) pair (x, y), respectively.

We use standard terminology from the book of Diestel [9] for graph-related terms that are not explicitly defined here. We consider only finite graphs. For a graph G, by $V(G)$ and $E(G)$ we denote the vertex and edge sets of G, respectively.

Similarly, for a directed graph or digraph D, by $V(D)$ and $A(D)$ we denote the vertex and arc sets of D, respectively. For a graph G and $v \in V(G)$, by $N_G(v)$ we denote the set $\{u \in V(G) \mid vu \in E(G)\}$. For a digraph D and $v \in V(D)$, by $N_D^+(v)$ we denote the set $\{u \in V(D) \mid vu \in A(D)\}$, and by $N_D^-(v)$ we denote the set $\{u \in V(D) \mid uv \in A(D)\}$. We drop the subscript G (or D) from $N_G(v)$, $N_D^+(v)$, or $N_D^-(v)$ when the context is clear. For a vertex subset $S \subseteq V(G)$, by $G[S]$ we denote the subgraph of G induced by S, i.e. the graph with vertex set S and edge set $\{vu \in E(G) \mid v, u \in S\}$. By $G - S$ we denote the graph $G[V(G) \setminus S]$.

A *path* in a graph is a sequence of vertices $P = v_1, v_2, \ldots, v_\ell$ such that for all $i \in [\ell - 1]$, $v_i v_{i+1} \in E(G)$. We say that such a path is a path between v_1 and v_ℓ or a $v_1 - v_\ell$ path of length $\ell - 1$, and vertices $v_1, v_2, \ldots, v_\ell$ lie on the path P. Two vertices $u, v \in V(G)$ are said to be *connected* if there exists a $u - v$ path in G. A graph is *connected* if there is a path between every pair of vertices. A maximal connected subgraph of G is called a *connected component* of G. For a pair of vertices $u, v \in V(G)$, by $\mathsf{dist}_G(u, v)$ we denote the length of the shortest path between u and v in G. For a graph G, by G^2 we denote the graph with vertex set $V(G^2) = V(G)$ and edge set $E(G^2) = \{uv \mid \mathsf{dist}_G(u, v) \leq 2\}$

A *tree* is a connected graph without any cycles. Note that a tree on n vertices has exactly $n - 1$ edges. A tree is said to be a *rooted tree* if exactly one vertex in it has been designated as its *root*. A *coloring* of a graph G with $\alpha \in \mathbb{N}$ colors is a map $\varphi : V(G) \to [\alpha]$. A coloring φ of G is said to be a *proper coloring* if for each $uv \in E(G)$, $\varphi(u) \neq \varphi(v)$.

A *tree decomposition* of a graph is a pair $(\mathcal{X}, \mathcal{T})$, where an element $X \in \mathcal{X}$ is a subset of $V(G)$, called a bag, and $\mathcal{T}$ is a rooted tree with vertex set $\mathcal{X}$ satisfying the following properties: (i) $\cup_{X \in \mathcal{X}} X = V(G)$; (ii) For every $uv \in E(G)$, there exists $X \in \mathcal{X}$ such that $u, v \in X$; (iii) For all $X, Y, Z \in \mathcal{X}$, if Y lies on the unique path between X and Z in $\mathcal{T}$, then $X \cap Z \subseteq Y$. For a graph G and its tree decomposition $(\mathcal{X}, \mathcal{T})$, the *width* of the tree decomposition $(\mathcal{X}, \mathcal{T})$ is defined to be $\max_{X \in \mathcal{X}}(|X| - 1)$. The *treewidth* of a graph G, $\mathsf{tw}(G)$, is the minimum of the widths of all its tree decompositions. A *path decomposition* of a graph is a tree decomposition $(\mathcal{X}, \mathcal{T})$, where $\mathcal{T}$ is a path. The width of a path decomposition is defined to be $\max_{X \in \mathcal{X}}(|X| - 1)$. The *pathwidth* of a graph G, $\mathsf{pw}(G) \geq \mathsf{tw}(G)$, is the minimum of the widths of all its path decompositions.

A *parameterized problem* Π is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. An instance of a parameterized problem is a tuple (x, κ), where κ is called the *parameter*. A parameterized problem is said to be *fixed-parameter tractable* (FPT) if, for a given instance (x, κ), we can decide $(x, \kappa) \in \Pi$ in time $f(\kappa) \cdot |x|^{\mathcal{O}(1)}$, where $f(\cdot)$ is an arbitrary function depending only on κ. To prove that a problem is FPT, it is possible to give an explicit algorithm, called a *parameterized algorithm* (or FPT algorithm), which runs in time $f(\kappa) \cdot |x|^{\mathcal{O}(1)}$. On the other hand, to show that a problem is unlikely to be FPT, it is possible to use a *parameterized reduction* running in FPT time. In the following, we formally define the notion of a parameterized reduction.

Definition 1. Let $\Pi, \Gamma \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A *parame-*

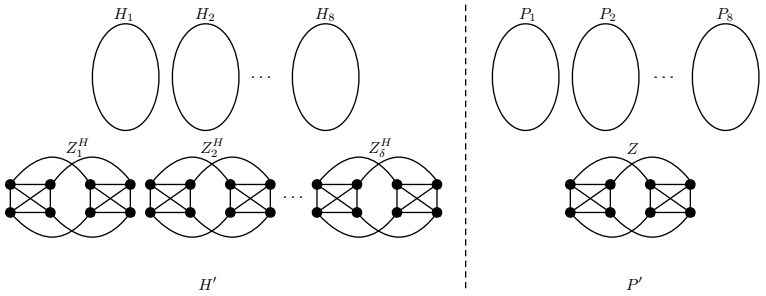

Figure 2: Construction of H' and P'.

terized reduction from Π to Γ is an algorithm that, given an instance (x, k) of Π, outputs an instance (x', k') of Γ such that the following conditions are satisfied:

1. (x, k) is a yes-instance of Π if and only if (x', k') is a yes-instance of Γ,
2. $k' \leq g(k)$, where $g(\cdot)$ is some (non-decreasing) computable function, and
3. the running time of the algorithm is bounded by $f(k)|x|^{\mathcal{O}(1)}$, where $f(\cdot)$ is some (non-decreasing) computable function.

We say that an instance (x, k) of Π and (x', k') of Γ are equivalent if (x, k) is a yes-instance of Π if and only if (x', k') is a yes-instance of Γ. We note that, if there is a parameterized reduction from a parameterized problem Π to a parameterized problem Γ, and Γ is FPT, then Π is also FPT. The notion of parameterized reduction is analogous to the concept of reductions in classical complexity theory. Here, the notion of W[1]-hardness replaces that of NP-hardness. For more details on parameterized complexity we refer to the books of Downey and Fellows [6], Flum and Grohe [10], Niedermeier [11], and the recent book by Cygan et al. [12].

3. W[1]-hardness of SumCSP

Let $(P, H, \mathsf{col} : V(H) \to [\ell])$ be an instance of PSI, where $V(P) = \{p_i \mid i \in [\ell]\}$ and $V(H) = \{h_i \mid i \in [n]\}$. For $i \in [\ell]$, we let $C_i^H = \{h \in V(H) \mid \mathsf{col}(h) = i\}$. We make a few assumption and adopt some conventions that will help simplify the presentation. All numbers that appear in the construction will be represented in binary. We assume that $|V(H)| = n = 2^t$, for some $t \in \mathbb{N}$, i.e., $t = \log n$. This assumption is justified in the following paragraph.

Ensuring the size constraint on H. Assume that $|V(H)| = 2^{t'} - \delta$, for some $0 < \delta < 2^{t'-1}$, as otherwise we already have an instance $(P, H, \mathsf{col} : V(H) \to [\ell])$ of PSI, with $|V(H)| = 2^t$, where $t \in \mathbb{N}$. We construct an equivalent instance $(P', H', \mathsf{col}' : V(H') \to [\ell'])$ of PSI with $|V(H')| = 2^{t'+3}$ as follows (see Figure 2). Initially, we let H' and P' to be the disjoint union of 8 copies of H and P, respectively. Here, for each $i \in [8]$, we denote the ith copy of H and P in H' and P' by H_i and P_i, respectively. For each $i \in [8]$, we let

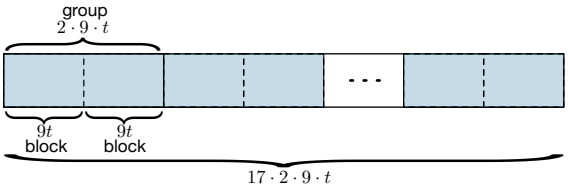

Figure 3: An illustration of the division of a bitstring into groups and blocks.

$V(P_i) = \{p_{\ell(i-1)+j} \mid j \in [\ell]\}$ and $V(H_i) = \{h_{\ell(i-1)+j} \mid j \in [n]\}$. For each $i \in [8]$ and $v \in V(H_i)$, we let $\mathsf{col}'(v) = \ell(i-1) + \mathsf{col}(v)$. The intuition behind the partial construction of P' and H' so far is to ensure that vertices in P_i get mapped only to vertices in H_i. In the above partial construction, the sizes of H' and P' are $8 \cdot 2^{t'} - 8\delta$ and 8ℓ, respectively. We then add δ graphs each on 8 vertices to H' and a graph on 8 vertices to P' as follows. For $i \in [\delta]$, let Z_i^H be the graph with vertex set $V(Z_i^H) = \{z_j^i \mid j \in [8]\}$. Moreover, $Z_i^H[\{z_j^i \mid j \in [4]\}]$ and $Z_i^H[\{z_j^i \mid j \in [8] \setminus [4]\}]$ are cliques on 4 vertices and for each $j \in [4]$, we have the edge $z_j^i z_{j+4}^i$ in Z_i^H. Let Z be a graph with vertex set $\{p_{8\ell+j} \mid j \in [8]\}$. We have that $Z[\{p_{8\ell+j} \mid j \in [4]\}]$ and $Z[\{p_{8\ell+j} \mid j \in [8] \setminus [4]\}]$ are cliques on 4 vertices and for each $j \in [4]$, we have the edge $z_{8\ell+j} z_{8\ell+j+4}$ in Z. For each $i \in [\delta]$ and $j \in [8]$, we let $\mathsf{col}'(z_j^i) = 8\ell + j$. This completes the description of the new instance $(P', H', \mathsf{col}' : V(H') \to [\ell'])$ of PSI, where $\ell' = 8(\ell+1)$ and $|V(H')| = 2^{t'+3}$. Note that Z is isomorphic to Z_i^H, for each $i \in [\delta]$, and, by construction, each vertex in Z can only be mapped to exactly one vertex of Z_i^H, where $i \in [\delta]$. It is easy to see that $(P, H, \mathsf{col} : V(H) \to [\ell])$ and $(P', H', \mathsf{col}' : V(H') \to [\ell'])$ are equivalent instances of PSI.

From the above discussion, we therefore assume that $(P, H, \mathsf{col} : V(H) \to [\ell])$ is an instance of PSI, such that $|V(H)| = n = 2^t$, where $t \in \mathbb{N}$. Note that P is a 4-regular graph, which implies that it has no isolated vertices. We assume a fixed cyclic ordering $\prec_H$ on the vertices in H and a fixed cyclic ordering $\prec_P$ on the vertices in P. Simply put, we have $h_1 \prec_H \ldots \prec_H h_n \prec_H h_1$ and $p_1 \prec_P \ldots \prec_P p_\ell \prec_P p_1$. With each vertex $h_i \in V(H)$, or equivalently integer $i \in [n]$, we assign two binary strings (or bitstrings for short) B_{h_i} and $\overline{B}_{h_i}$ as follows. We let $\mathbb{B}_i$ denote the binary representation of the integer i and $\overline{\mathbb{B}}_i$ denote the (bitwise) complement of $\mathbb{B}_i$. We use $\mathbb{O}_z$ and $\mathbb{1}_z$ to denote the bitstrings of length z consisting of all zeros and all ones, respectively. We let $B_{h_i} = \mathbb{O}_{4t}\mathbb{B}_i\mathbb{O}_{4t}$ and $\overline{B}_{h_i} = \mathbb{O}_{4t}\overline{\mathbb{B}}_i\mathbb{O}_{4t}$. Note that B_{h_i} and $\overline{B}_{h_i}$ are of length $9 \log n = 9t$. The purpose of the additional zero bits is to allow us to "correctly" handle overflows when summing binary numbers. For two bitstrings B and B', we slightly abuse notation and sometimes treat the result of $B + B'$ as another bitstring (obtained after applying the usual binary addition operator) or as an integer (in base 10). The context will be clear.

We also assume that, along with instance $(P, H, \mathsf{col} : V(H) \to [\ell])$, we are given a proper coloring $\mathsf{col}_{P^2} : V(P) \to [17]$ of P^2. Observe that such a coloring

exists, and can be computed in time polynomial in the size of the graph P; the maximum degree of a vertex in P^2 is bounded by 16 and a graph with maximum degree d admits a $(d+1)$-proper coloring. For a vertex $p_i \in V(P)$, we let $\mathsf{id}_i = \mathsf{col}_{P^2}(p_i)$. In what follows, we will always deal with bitstrings of length $17 \cdot 2 \cdot 9 \cdot t$. A *block* consists of $9t$ consecutive bits. We note that two distinct blocks do not intersect in any bit position. Blocks will usually be set to bistrings of the form $\mathbb{O}_{4t}\mathbb{B}_i\mathbb{O}_{4t}$, $\mathbb{O}_{4t}\overline{\mathbb{B}}_i\mathbb{O}_{4t}$, $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$, $\mathbb{O}_{9t}$, or $\mathbb{1}_{9t}$, $i \in [n]$. A *group* consists of $2 \cdot 9 \cdot t$ consecutive bits. Two distinct groups do not intersect in any bit position and a group consists of two consecutive blocks (see Figure 3). Note that we have exactly 17 groups, which is equal to the number of colors in col_{P^2}. Groups and/or blocks are concatenated consecutively one after the other with no overlaps. The reason why we need col_{P^2} will become clear later. Intuitively, since we will be encoding the possible edges (from H) between a vertex in P and its four neighbors, we need to make sure that no two of its neighbors get assigned the same group in a bitstring (regardless of whether these two neighbors share an edge in P or not). Given a bitstring S of length $17 \cdot 2 \cdot 9 \cdot t$, we let $\mathsf{block}[i](S)$ denote the ith block of S, $i \in [34]$, and we let $\mathsf{group}[j](S)$ denote the jth group of S, $j \in [17]$. We also use the notation $\mathsf{group}[i \mid j](S)$ to denote the ith and jth group of S, $i, j \in [17]$. Finally, we note that, since the length of bitstrings will be bounded by $\mathcal{O}(\log n)$, all numbers in the construction will be bounded by $n^{\mathcal{O}(1)}$. We are now ready to describe the construction of instance $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ of SumCSP. We start with the description of the edge selection gadget.

Edge selection gagdet. For every (unordered) pair of (distinct) numbers $i, j \in [\ell]$ such that $p_ip_j \in E(P)$, we add an edge selection gadget E_{ij} (E_{ij} is a graph and not an edge set) to D. Note that both E_{ij} and E_{ji} refer to the same edge selection gadget, which will be responsible for selecting an edge in the host graph H. Moreover, $\mathsf{id}_i \neq \mathsf{id}_j$, since col_{P^2} is a proper coloring of P^2. We assume, without loss of generality, that $i < j$. We let $V(E_{ij}) = \{a_i^{ij}, a_j^{ij}, b_i^{ij}, b_j^{ij}, w_{ij}\}$ and we let $A(E_{ij}) = \{a_i^{ij}a_j^{ij}, a_j^{ij}w_{ij}, w_{ij}a_i^{ij}, b_i^{ij}b_j^{ij}, b_j^{ij}w_{ij}, w_{ij}b_i^{ij}\}$ (see Figure 4).

We now describe the construction of $\varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}}$ and $w_A : A(D) \to \mathbb{N}$. First, let us consider the construction of $\varphi|_{E_{ij}}$ and $w_A|_{E_{ij}}$, i.e., φ and w_A restricted to E_{ij}, respectively. The intuition behind the construction of $\varphi|_{E_{ij}}$ is to ensure that, in any valid solution, pairs selected for each arc in E_{ij} all correspond to a pair $u \in C_i^H$ and $v \in C_j^H$, where $uv \in E(H)$. Moreover, the construction of $w_A|_{E_{ij}}$ ensures that all pairs in an arc in E_{ij} sum to the same number. We assume that all bitstrings are initialized to $\mathbb{O}_{306t}$. That is, whenever we do not explicitly specify the value of a group (block) in a bitstring, it is set to all zeros. For each $u \in C_i^H$ and $v \in C_j^H$, where $uv \in E(H)$, and for each arc e in E_{ij}, we will add a pair of bitstrings $(S_{uv}(e), T_{uv}(e))$ to $\varphi(e)$, the construction for which is described, shortly. To this end, consider $u \in C_i^H$ and $v \in C_j^H$ such that $uv \in E(H)$. Next, for each arc $a \in A(E_{ij})$, we describe the construction of the pair $(S_{uv}(a), T_{uv}(a))$ and the weight $w_A(a)$.

- For $a_i^{ij}a_j^{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}\ a_j^{ij}))$ (which

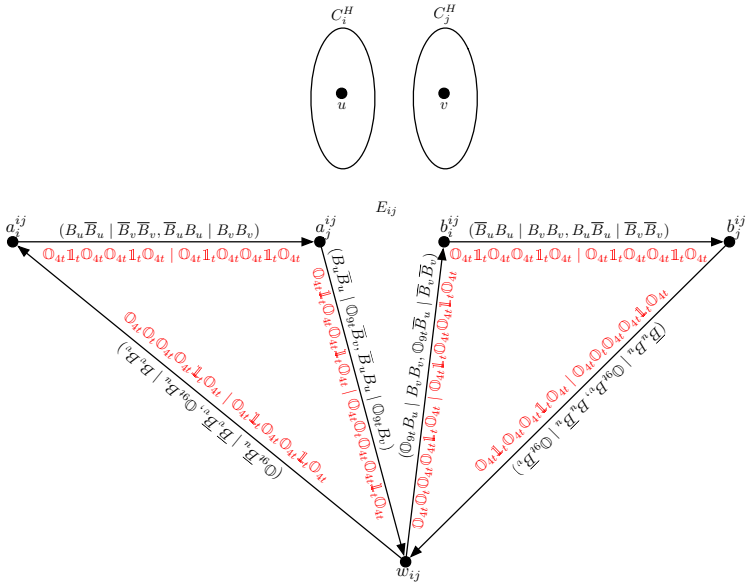

Figure 4: An illustration of the edge selection gadget E_{ij}, where $i < j$, and only the ith and jth block of each bitstring are shown (other bits are set to 0). Red bitstrings represent weights on the arcs, while black bitstrings represent pairs corresponding to $u \in C_i^H$ and $v \in C_j^H$.

is added to $\varphi(a_i^{ij}a_j^{ij})$), and the weight $w_A(a_i^{ij}a_j^{ij})$ is obtained by setting the following groups:
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(a_i^{ij}a_j^{ij})) = B_u\overline{B}_u \mid \overline{B}_v\overline{B}_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(a_i^{ij}a_j^{ij})) = \overline{B}_uB_u \mid B_vB_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(a_i^{ij}a_j^{ij})) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

- For $a_j^{ij}w_{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(a_j^{ij}w_{ij}), T_{uv}(a_j^{ij}w_{ij}))$ and $w_A(a_j^{ij}w_{ij})$ is obtained by setting the following groups:
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(a_j^{ij}w_{ij})) = B_u\overline{B}_u \mid \mathbb{O}_{9t}\overline{B}_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(a_j^{ij}w_{ij})) = \overline{B}_uB_u \mid \mathbb{O}_{9t}B_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(a_j^{ij}w_{ij})) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{O}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

- For $w_{ij}a_i^{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(w_{ij}a_i^{ij}), T_{uv}(w_{ij}a_i^{ij}))$ and $w_A(w_{ij}a_i^{ij})$ is obtained by setting the following groups:
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(w_{ij}a_i^{ij})) = \mathbb{O}_{9t}\overline{B}_u \mid \overline{B}_v\overline{B}_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(w_{ij}a_i^{ij})) = \mathbb{O}_{9t}B_u \mid B_vB_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(w_{ij}a_i^{ij})) = \mathbb{O}_{4t}\mathbb{O}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

- For $b_i^{ij}b_j^{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(b_i^{ij}b_j^{ij}), T_{uv}(b_i^{ij}b_j^{ij}))$ and $w_A(b_i^{ij}b_j^{ij})$ is obtained by setting the following groups:

$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(b_i^{ij}b_j^{ij})) = \overline{B}_u B_u \mid B_v B_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(b_i^{ij}b_j^{ij})) = B_u \overline{B}_u \mid \overline{B}_v \overline{B}_v$;
$\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(b_i^{ij}b_j^{ij})) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

- For $b_j^{ij}w_{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(b_j^{ij}w_{ij}), T_{uv}(b_j^{ij}w_{ij}))$ and $w_A(b_i^{ij}w_{ij})$ is obtained by setting the following groups:
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(b_j^{ij}w_{ij})) = \overline{B}_u B_u \mid \mathbb{O}_{9t} B_v$;
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(b_j^{ij}w_{ij})) = B_u \overline{B}_u \mid \mathbb{O}_{9t} \overline{B}_v$;
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(b_j^{ij}w_{ij})) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{O}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

- For $w_{ij}b_i^{ij} \in A(E_{ij})$, the pair of bitstrings $(S_{uv}(w_{ij}b_i^{ij}), T_{uv}(w_{ij}, b_i^{ij}))$ and $w_A(w_{ij}b_i^{ij})$ is obtained by setting the following groups:
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](S_{uv}(w_{ij}b_i^{ij})) = \mathbb{O}_{9t} B_u \mid B_v B_v$;
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](T_{uv}(w_{ij}, b_i^{ij})) = \mathbb{O}_{9t} \overline{B}_u \mid \overline{B}_v \overline{B}_v$;
  $\mathsf{group}[\mathsf{id}_i \mid \mathsf{id}_j](w_A(w_{ij}b_i^{ij})) = \mathbb{O}_{4t}\mathbb{O}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t} \mid \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.

By construction, for each $u \in C_i^H$ and v_j^H, where $uv \in E(H)$, and arc $a \in A(E_{ij})$, we have $S_{uv}(a) + T_{uv}(a) = w_A(a)$.

Compatibility between edge selection gadgets. We add edges between various edge selection gadgets to ensure that for each $i \in [\ell]$, the edges selected by the gadgets are incident on the same vertex in C_i^H. The selection of an edge by a gadget will be determined by the pair of numbers selected from $\varphi(a)$, where $a \in A(E_{ij})$ and $p_ip_j \in E(P)$. For each $p_i \in V(P)$, we have $|N_P(p_i)| = 4$, since P is a 4-regular graph. For $i \in [\ell]$, let $N_P(p_i) = \{p_{j_1}, p_{j_2}, p_{j_3}, p_{j_4}\}$, where we have a (fixed and cyclic) ordering on the vertices in $N_P(p_i)$ based on the ordering $\prec_P$. That is, we assume $p_{j_1} \prec_P p_{j_2} \prec_P p_{j_3} \prec_P p_{j_4} \prec_P p_{j_1}$. We will describe the set of arcs added between $E_{ij_1}, E_{ij_2}, E_{ij_3}$ and E_{ij_4}, we will call this set A_i. Similar to the construction of $\varphi(\cdot)$ for edge selection gadgets, here for each arc $a \in A_i$, for each $u \in C_i^H$, we will add a pair of bitstrings $(S_u(a), T_u(a))$ to $\varphi(a)$. Again, the intuition behind the construction of $\varphi|_{A_i}$ will be to ensure that, in any valid solution, pairs selected for each arc in A_i all correspond to a vertex $u \in C_i^H$. Moreover, the construction of $w_A|_{A_i}$ will ensure that all pairs in an arc in A_i sum to the same number.

We set $A_i = \{a_i^{ij_1}b_i^{ij_2}, a_i^{ij_2}b_i^{ij_3}, a_i^{ij_3}b_i^{ij_4}, a_i^{ij_4}b_i^{ij_1}\}$. Consider an arc $a \in A_i$. We obtain $w_A(a)$, and for each $u \in C_i^H$, the pair $(S_u(a), T_u(a))$ (which is added to $\varphi(a)$) by setting the following groups (see Figure 5):
$\mathsf{group}[\mathsf{id}_i](S_u(a)) = \overline{B}_u \mathbb{O}_{9t}$;
$\mathsf{group}[\mathsf{id}_i](T_u(a)) = B_u \mathbb{O}_{9t}$;
$\mathsf{group}[\mathsf{id}_i](w_A(a)) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{9t}$.

This completes the description of the vertices and arcs of D, and the functions $w_A : A(D) \to \mathbb{N}$ and $\varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}}$. We now move to description of the functions $w_V : V(D) \to \mathbb{N}$.

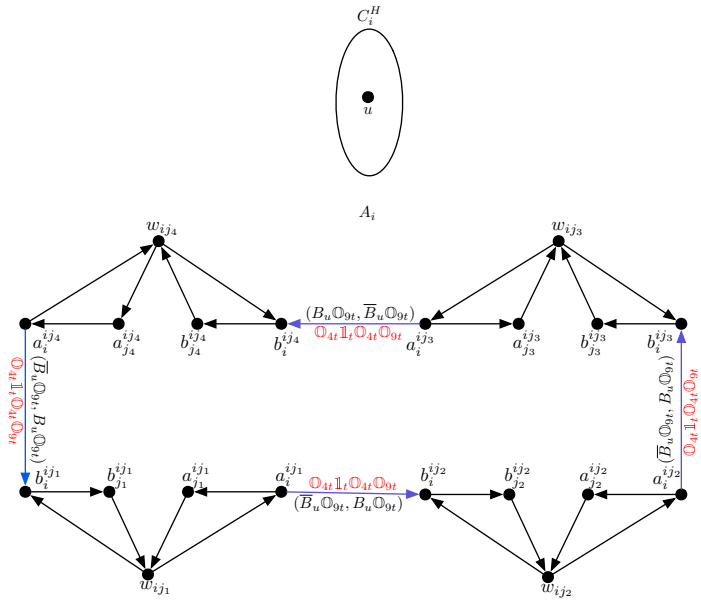

Figure 5: An illustration of edge selection gadgets and the additional edges between them.

The vertex weight function. For each $i, j \in [\ell]$, $i < j$, we set $w_V(\cdot)$ as follows.

- For all $u \in \{a_i^{ij}, a_j^{ij}, b_i^{ij}, b_j^{ij}, \}$, we set $w_V(u)$ to be the bitstring X_u of length $306 \log n$, where $\mathsf{group}[\mathsf{id}_i](X_u) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ and $\mathsf{group}[\mathsf{id}_j](X_u) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$.
- For w_{ij}, we set $w_V(w_{ij})$ to be the bitstring $X_{w_{ij}}$ of length $306 \log n$, which we construct as follows. We let Y be the bitstring of length t corresponding to the integer $2^t - 2$, i.e. a bitstring of length t with the last bit set to zero and all other bits set to one. Let Y' to be the bitstring of length $4t$ corresponding to the integer 1, i.e. the bitstring of length $4t$ with the last bit set to one and all other bits set to zero. We set $\mathsf{group}[\mathsf{id}_i](X_{w_{ij}}) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}Y'Y\mathbb{O}_{4t}$ and $\mathsf{group}[\mathsf{id}_j](X_{w_{ij}}) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}Y'Y\mathbb{O}_{4t}$.

This finishes the description of the instance $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ of SumCSP for a given instance $(P, H, \mathsf{col} : V(H) \to [\ell])$ of PSI. Below we state some propositions and lemmata that will be useful in establishing the equivalence of the two instances.

Proposition 2. *Let X, Y be two bitstrings of length $\log q$. Then $X + Y = 2^q - 1$ if and only if $\overline{X} = Y$.*

Proposition 3. *Let X and Y be two bitstrings each of length $34 \cdot 9 \cdot t$ and consisting of 17 groups, where $t = \log n$. Assume that, for each $i \in [17]$, group i*

in X consists of a bitstring of the form $X_i = \mathbb{O}_{4t}\mathbb{B}_x\mathbb{O}_{4t}$ and group i in Y consists of a bitstring of the form $Y_i = \mathbb{O}_{4t}\mathbb{B}_y\mathbb{O}_{4t}$, $x, y \in [n]$. Then, $X + Y$ is a bitstring of length $34 \cdot 9 \cdot t$ with the ith group is equal to $X_i + Y_i$, $i \in [17]$.

Lemma 1. *Let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be a yes-instance of* SumCSP *and $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ be a solution. Consider $p_i, p_{i'}, p_j, p_{j'} \in V(P)$ such that $a_i^{ij}b_i^{ij'}, a_j^{ij}b_j^{i'j} \in A(D)$. For $u \in C_i^H$ and $v \in C_j^H$, we have $\rho(a_i^{ij}a_j^{ij}) = (S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}a_j^{ij}))$ if and only if $\rho(a_i^{ij}b_i^{ij'}) = (S_u(a_i^{ij}b_i^{ij'}), T_u(a_i^{ij}b_i^{ij'}))$ and $\rho(a_j^{ij}b_j^{i'j}) = (S_v(a_j^{ij}b_j^{i'j}), T_v(a_j^{ij}b_j^{i'j}))$.*

Proof. Let $u \in C_i^H$ and $v \in C_j^H$ such that $\rho(a_i^{ij}b_i^{ij'}) = (S_u(a_i^{ij}b_i^{ij'}), T_u(a_i^{ij}b_i^{ij'}))$ and $\rho(a_j^{ij}, b_j^{i'j}) = (S_v(a_j^{ij}b_j^{i'j}), T_v(a_j^{ij}b_j^{i'j}))$. Also, let $u' \in C_i^H$ and $v' \in C_j^H$ such that $\rho(a_i^{ij}, a_j^{ij}) = (S_{u'v'}(a_i^{ij}a_j^{ij}), T_{u'v'}(a_i^{ij}a_j^{ij}))$. Observe that it is enough to show that $u = u'$ and $v = v'$. Recall that by construction we have $N_D^-(a_i^{ij}) = \{w_{ij}\}$, $N_D^+(a_i^{ij}) = \{a_j^{ij}, b_i^{ij'}\}$, and $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](w_V(a_i^{ij}))) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$, i.e. the first block of the id_ith group of $w_V(a_i^{ij})$ is $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$. Moreover, we have $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](T_{uv}(w_{ij}a_i^{ij}))) = \mathbb{O}_{9t}$, $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](S_{u'v'}(a_i^{ij}a_j^{ij}))) = B_{u'}$, and $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](S_{uv}(a_i^{ij}b_i^{ij'}))) = \overline{B}_u$. Combining Propositions 2 and 3 with the fact that $\mathbb{O}_{9t}$, $B_{u'}$, and $\overline{B}_u$ must sum to $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ implies that $u = u'$. An analogous argument can be given to show that $v = v'$. □

Lemma 2. *Let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be a yes-instance of* SumCSP *and $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ be a solution. Consider $p_i, p_{i'}, p_j, p_{j'} \in V(P)$ such that $a_i^{ij'}b_i^{ij}, a_j^{i'j}b_j^{ij} \in A(D)$. For $u \in C_i^H$ and $v \in C_j^H$, we have $\rho(b_i^{ij}b_j^{ij}) = (S_{uv}(b_i^{ij}b_j^{ij}), T_{uv}(b_i^{ij}b_j^{ij}))$ if and only if $\rho(a_i^{ij'}b_i^{ij}) = (S_u(a_i^{ij'}b_i^{ij}), T_u(a_i^{ij'}b_i^{ij}))$ and $\rho(a_j^{i'j}b_j^{ij}) = (S_v(a_j^{i'j}b_j^{ij}), T_v(a_j^{i'j}b_j^{ij}))$.*

Proof. Let $u \in C_i^H$ and $v \in C_j^H$ such that $\rho(a_i^{ij'}b_i^{ij}) = (S_u(a_i^{ij'}b_i^{ij}), T_u(a_i^{ij'}b_i^{ij}))$ and $\rho(a_j^{i'j}b_j^{ij}) = (S_v(a_j^{i'j}b_j^{ij}), T_v(a_j^{i'j}b_j^{ij}))$. Also, let $u' \in C_i^H$ and $v' \in C_j^H$ such that $\rho(b_i^{ij}b_j^{ij}) = (S_{u'v'}(b_i^{ij}b_j^{ij}), T_{u'v'}(b_i^{ij}b_j^{ij}))$. It is enough to show that $u = u'$ and $v = v'$. Recall that by construction we have $N_D^-(b_i^{ij}) = \{w_{ij}, a_i^{ij'}\}$, $N_D^+(b_i^{ij}) = \{b_j^{ij}\}$, and $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](w_V(a_i^{ij}))) = \mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ Moreover, we have $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](T_{uv}^{(w_{ij},b_i^{ij})})) = \mathbb{O}_{9t}$, $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](S_{u'v'}(b_i^{ij}b_j^{ij}))) = \overline{B}_{u'}$, and $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_i](T_{uv}(a_i^{ij'}b_i^{ij}))) = B_u$. Combining Propositions 2 and 3 with the fact that $\mathbb{O}_{9t}$, $\overline{B}_{u'}$, and B_u must sum to $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ implies that $u = u'$. An analogous argument can be given to show that $v = v'$. □

Lemma 3. *Let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be a yes-instance of* SumCSP *and $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ be a solution. Let $i, j \in [\ell]$, where $i < j$ and $p_ip_j \in E(P)$, and let $u \in C_i^H$ and $v \in C_j^H$. Then, the following three statements are equivalent:*

(1) $\rho(a_i^{ij}a_j^{ij}) = (S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}a_j^{ij}))$;

(2) $\rho(w_{ij}a_i^{ij}) = (S_{uv}(w_{ij}a_i^{ij}), T_{uv}(w_{ij}a_i^{ij}))$;

(3) $\rho(a_j^{ij}w_{ij}) = (S_{uv}(a_j^{ij}w_{ij}), T_{uv}(a_j^{ij}w_{ij}))$.

Proof. Let $u \in C_i^H$, $v \in C_j^H$, and $\rho(a_i^{ij}a_j^{ij}) = (S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}a_j^{ij}))$. Moreover, let $\rho(w_{ij}a_i^{ij}) = (S_{u'v'}(w_{ij}a_i^{ij}), T_{u'v'}(w_{ij}a_i^{ij}))$, where $u' \in C_i^H$ and $v' \in C_j^H$. To prove that statement (1) holds if and if only statement (2) holds, it is enough to show that $u = u'$ and $v = v'$. Recall that by construction we have $N_D^-(a_i^{ij}) = \{w_{ij}\}$ and $N_D^+(a_i^{ij}) = \{a_j^{ij}, b_i^{ij'}\}$, for some $b_i^{ij'} \in V(D)$ where $p_{j'}$ is a neighbor of p_i which comes after p_j in the fixed cyclic ordering of the neighbors of p_i. Moreover, we have $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](T_{u'v'}(w_{ij}a_i^{ij}))) = B_{u'}$, $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](S_{uv}(a_i^{ij}a_j^{ij}))) = \overline{B}_u$, and $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](T_{u''v''}(a_i^{ij}b_i^{ij'}))) = \mathbb{O}_{9t}$, where $u'' \in C_i^H$, $v'' \in C_j^H$, and $\rho(b_i^{ij}a_i^{ij'}) = (S_{u''v''}(b_i^{ij}a_i^{ij'}), T_{u''v''}(b_i^{ij}a_i^{ij'}))$. This together with the fact that the second block in the id_ith group of $w_V(a_i^{ij})$ is $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ (and Propositions 2 and 3) implies that $u = u'$. An analogous argument can be given to show that $v = v'$. Using a symmetric argument, it can be shown that statement (1) holds if and only if statement (3) holds. □

Lemma 4. *Let* $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ *be a yes-instance of* SumCSP *and* $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ *be a solution. Let* $i, j \in [\ell]$, *where* $i < j$ *and* $p_ip_j \in E(P)$, *and let* $u \in C_i^H$ *and* $v \in C_j^H$. *Then, the following three statements are equivalent:*

(1) $\rho(b_i^{ij}b_j^{ij}) = (S_{uv}(b_i^{ij}b_j^{ij}), T_{uv}(b_i^{ij}b_j^{ij}))$

(2) $\rho(w_{ij}b_i^{ij}) = (S_{uv}^{(w_{ij},b_i^{ij})}, T_{uv}^{(w_{ij},b_i^{ij})})$;

(3) $\rho(b_j^{ij}w_{ij}) = (S_{uv}(b_j^{ij}w_{ij}), T_{uv}(b_j^{ij}w_{ij}))$.

Proof. For $u \in C_i^H$ and $v \in C_j^H$, let $\rho(b_i^{ij}b_j^{ij}) = (S_{uv}(b_i^{ij}b_j^{ij}), T_{uv}(b_i^{ij}b_j^{ij}))$. Moreover, let $\rho(w_{ij}b_i^{ij}) = (S_{u'v'}(w_{ij}b_i^{ij}), T_{u'v'}(w_{ij}b_i^{ij}))$, where $u' \in C_i^H$ and $v' \in C_j^H$. To prove that statement (1) holds if and only if statement (2) holds, it is enough to show that $u = u'$ and $v = v'$. Recall that by construction we have $N_D^+(a_i^{ij}) = \{b_j^{ij}\}$ and $N_D^-(b_i^{ij}) = \{w_{ij}, a_i^{ij'}\}$, for some $b_i^{ij'} \in V(D)$ where $p_{j'}$ is a neighbor of p_i which comes after p_j in the fixed cyclic ordering of the neighbors of p_i. Moreover, we have $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](T_{u'v'}(w_{ij}a_i^{ij}))) = \overline{B}_{u'}$, $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](S_{uv}(b_i^{ij}b_j^{ij}))) = B_u$, and $\mathsf{block}[2](\mathsf{group}[\mathsf{id}_i](T_{u''v''}(b_i^{ij}a_i^{ij'}))) = \mathbb{O}_{9t}$, where $u'' \in C_i^H$, $v'' \in C_j^H$, and $\rho(b_i^{ij}a_i^{ij'}) = (S_{u''v''}(b_i^{ij}a_i^{ij'}), T_{u''v''}(b_i^{ij}a_i^{ij'}))$. This together with the fact that second block in the id_ith group of $w_V(b_i^{ij})$ is $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ (and Propositions 2 and 3) implies that $u = u'$. An analogous argument can be given to show that $v = v'$. Using a symmetric argument, it can be shown that statement (1) holds if and only if statement (3) holds. □

Lemma 5. *Let* $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ *be a yes-instance of* SumCSP *and* $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ *be a solution. Let* $i, j \in [\ell]$, $i < j$, $p_ip_j \in E(P)$, *and* $u \in C_i^H$ *and* $v \in C_j^H$. *Then,* $\rho(a_i^{ij}a_j^{ij}) = (S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}a_j^{ij}))$ *if and only if* $\rho(b_i^{ij}b_j^{ij}) = (S_{uv}(b_i^{ij}b_j^{ij}), T_{uv}(b_i^{ij}b_j^{ij}))$.

Proof. Let $i, j \in [\ell]$, $i < j$, and let $(p_i, p_j) \in E(P)$. Let $u, u' \in C_i^H$ and $v, v' \in C_j^H$ such that $\rho(a_i^{ij}a_j^{ij}) = (S_{uv}(a_i^{ij}a_j^{ij}), T_{uv}(a_i^{ij}a_j^{ij}))$ and $\rho(b_i^{ij}b_j^{ij}) = (S_{u'v'}(b_i^{ij}b_j^{ij}), T_{u'v'}(b_i^{ij}b_j^{ij}))$. We need to show that $u' = u$ and $v' = v$. From Lemmas 3 and 4, we know that $\rho(w_{ij}a_i^{ij}) = (S_{uv}(w_{ij}a_i^{ij}), T_{uv}(w_{ij}a_i^{ij}))$, $\rho(a_j^{ij}w_{ij}) = (S_{uv}(a_j^{ij}w_{ij}), T_{uv}(a_j^{ij}w_{ij}))$, $\rho(b_j^{ij}w_{ij}) = (S_{uv}(b_j^{ij}w_{ij}), T_{uv}(b_j^{ij}w_{ij}))$, and $\rho(w_{ij}b_i^{ij}) = (S_{u'v'}(w_{ij}b_i^{ij}), T_{u'v'}(w_{ij}b_i^{ij}))$. Moreover, we have $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_j](S_{uv}\,(w_{ij}a_i^{ij}))) = \overline{B}_v$, $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_j](S_{u'v'}(w_{ij}b_i^{ij}))) = B_{v'}$, $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_j]\,(T_{uv}(a_j^{ij}w_{ij}))) = \mathbb{O}_{9t}$, and $\mathsf{block}[1](\mathsf{group}[\mathsf{id}_j](T_{u'v'}(b_j^{ij}w_{ij}))) = \mathbb{O}_{9t}$. This together with the fact that first block in the id_jth group of $w_V(w_{ij})$ is $\mathbb{O}_{4t}\mathbb{1}_t\mathbb{O}_{4t}$ (and Propositions 2 and 3) implies that $v = v'$. An analogous argument shows that $u = u'$. □

Lemma 6. $(P, H, \mathit{col} : V(H) \to [\ell])$ *is a yes-instance of* PSI *if and only if* $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ *is a yes-instance of* SumCSP.

Proof. In the forward direction, let $(P, H, \mathsf{col} : V(H) \to [\ell])$ be a yes-instance of PSI and $\phi : V(P) \to V(H)$ be an injective function such that for each $i \in [\ell]$, $\mathsf{col}(\phi(p_i)) = i$, and, for each $(p_i, p_j) \in E(P)$, we have $(\phi(p_i), \phi(p_j)) \in E(H)$. For $i \in [\ell]$, let $h_i^* = \phi(p_i)$. We now define $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ such that for each $e \in A(D)$, we have $\rho(e) \in \varphi(e)$ and for each $v \in V(D)$, $\sum_{u\in N^+(v)} \mathsf{fir}(\rho((v, u))) + \sum_{u\in N^-(v)} \mathsf{sec}(\rho((u, v))) = w_V(v)$. For $i \in [\ell]$ and for each $e \in E_i$, we set $\rho(e) = (S^e_{\phi(p_i)}, T^e_{\phi(p_i)})$. For $i, j \in [\ell]$, $i < j$ such that $(p_i, p_j) \in E(P)$ and for each $e \in A(E_{ij})$, we set $\rho(e) = (S^e_{\phi(p_i)\phi(p_j)}, T^e_{\phi(p_i)\phi(p_j)})$. Recall that by construction, ρ satisfies all the desired properties.

In the reverse direction let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be a yes-instance of SumCSP and $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ be a solution. From Lemmas 1 to 5, it follows that for each $i, j \in [\ell]$, $i < j$ with $(p_i, p_j) \in E(P)$, there exists $u \in C_i^H$ and $v \in C_j^H$ such that for all $e \in A(E_{ij})$, $\rho(e) = (S^e_{uv}, T^e_{uv})$, for all $e \in A(E_i)$, $\rho(e) = (S^e_u, T^e_e)$ and, for all $e \in A(E_j)$, $\rho(e) = (S^e_v, T^e_v)$. For $i \in [\ell]$, let h_i^* be the vertex such that for all $e \in A(E_i)$, we have $\rho(e) = (S^e_{h_i^*}, T^e_{h_i^*})$. We show that $\phi : V(P) \to V(H)$ such that for $i \in [\ell]$, $\phi(p_i) = h_i^*$, is a solution for PSI.

For $i \in [\ell]$ and $k \in [4]$, let p_{j_k} be neighbors of p_i in P such that for each $k' \in [3]$, $j_{k'} < j_{k'+1}$. Further, let h_i^* be the vertex such that $\rho((a_i^{ij_1}, b_i^{ij_2})) = (S^{(a_i^{ij_1}, b_i^{ij_2})}_{h_i^*}, T^{(a_i^{ij_1}, b_i^{ij_2})}_{h_i^*})$. We will show that $\phi : V(P) \to V(H)$ such that for $i' \in [\ell]$, $\phi(p_i) = h_i^*$ is a solution for PSI. From construction it follows that for each $i \in [\ell]$, $\mathsf{col}(h_i^*) = i$. Consider an edge $(p_i, p_j) \in E(P)$, where $i < j$. From construction of D we know that we have E_{ij} as sub-digraph of D. Furthermore,

we also have $(h_i^*, h_j^*) \in E(H)$ since for each $e \in A(E_{ij})$ we have $(S^e_{h_i^*}, T^e_{h_j^*}) \in \varphi(e)$ and we added such a pair only if h_i^* and h_j^* are adjacent. □

Theorem 7. SumCSP *is W[1]-hard when parameterized by the number of arcs in the directed graph, even when all numbers appearing in an instance are bounded by some polynomial.*

Proof. Let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be the constructed instance of SumCSP for the given instance $(P, H, \mathsf{col} : V(H) \to [\ell])$ of PSI. An easy trace of the construction shows that it can be accomplished in time polynomial in $|V(H)|$ and that all the numbers appearing in the construction are bounded by $|V(H)|^{\mathcal{O}(1)}$. Moreover, note that P is 4-regular and therefore $|E(P)| = \mathcal{O}(|V(P)|)$. Since (by construction) the number of arcs in D is linear in the number of edges in P, we have $|A(D)| = \mathcal{O}(|E(P)|) = \mathcal{O}(|V(P)|)$. Combining all of the above with Lemma 6 and the W[1]-hardness of PSI (parameterized by $|V(P)|$) completes the proof. □

4. W[1]-hardness of CNC

Recall that, in an instance (G, k, μ) of the CNC problem, we are given an undirected graph G and integers k and μ. The goal is to determine whether there exists a set $S \subseteq V(G)$ of size (exactly) k such that $\sum_{C\in\mathcal{C}(G-S)} \binom{C}{2} \leq \mu$, where $\mathcal{C}(G - S) = \{C_1, \ldots, C_\ell\}$ denotes the set of connected components in $G - S$. We let $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ be an instance of SumCSP. We let $w_{vmax} = \mathsf{max}_{v\in V(D)}(w_V(v))$ denote the maximum weight of a vertex in D. We let $w_{amax} = \mathsf{max}_{a\in A(D)}(w_A(a))$ denote the maximum weight of an arc in D. We let $w_{big} = (w_{amax} \cdot w_{vmax})^{100}$. We assume, without loss of generality, that the number of arcs in D is greater than some constant, say $|A(D)| \geq 50$, and that $w_{vmax} > 2|A(D)|$ (otherwise we can increase all numbers in the SumCSP instance appropriately). Moreover, we let $W^\star = (k+3)(w_{big}+w_{vmax}+2)$, where $k = 2|A(D)|$. For each vertex $v \in V(D)$, we define a quantity $W_v = W^\star - (k+3)(w_V(v)+2) = (k+3)(w_{big}+w_{vmax}-w_V(v))$. We shall create an instance (G, k, μ) of CNC, where $k = 2|A(D)|$ (note that k is even), $\mu = |V(D)| \cdot \binom{W^\star}{2}$, and $\mathsf{tw}(G) = k^{\mathcal{O}(1)}$. We now proceed to the construction of the graph G.

Construction. For each vertex $v \in V(D)$, we create a clique K_v of size $2(k + 3)$ and an independent set I_v of size W_v. We add all edges between vertices in K_v and vertices in I_v. For each arc $a = uv \in A(D)$, we create a *chain* H_{uv} (which will connect K_u and K_v) as follows. H_{uv} consists of $w_A(a) + 1$ *bridging pairs* of vertices $\mathcal{P}_{uv} = \{p_0, \ldots, p_{w_A(uv)}\}$, where each pair $p_i \in \mathcal{P}_{uv}$ consists of two (independent) vertices $\{p_i^1, p_i^2\}$. Moreover, we have $w_A(a)$ *border walls* $\mathcal{B}_{uv} = \{b_1, \ldots, b_{w_A(a)}\}$, each of size $k + 1$, i.e. each wall consists of $k + 1$ (independent) vertices. We add all edges between K_u and pair p_0 and we add all edges between K_v and $p_{w_A(uv)}$. Next, we add all edges between p_{i-1} and b_i and all edges between b_i and p_i, for $i \in [w_A(a)]$. We call the pair p_0 the *first pair*

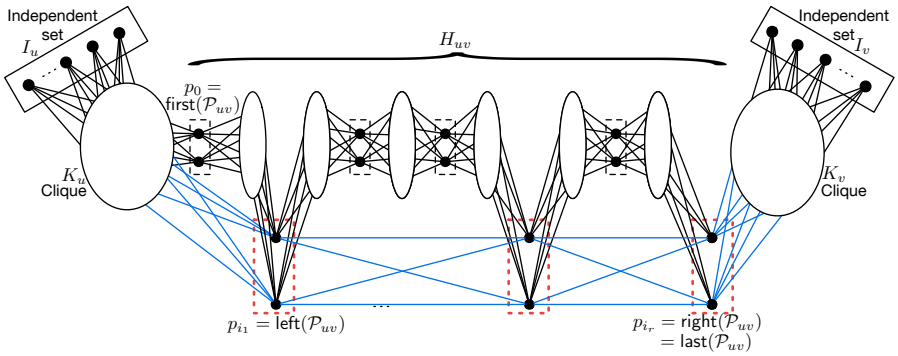

Figure 6: An illustration of parts of the construction of the graph G. Border walls (of size $k+1$) are connected via bridging pairs. Cliques are of size $2(k+3)$ and independent sets I_u and I_v are of size W_u and W_v, respectively.

of $\mathcal{P}_{uv}$ and denote it by $\mathsf{first}(\mathcal{P}_{uv})$. Similarly, we call the pair $p_{w_A(uv)}$ the *last pair* of $\mathcal{P}_{uv}$ and denote it by $\mathsf{last}(\mathcal{P}_{uv})$. Then, we sort all entries $(i,j) \in \varphi(a)$ in increasing order based on the first coordinate. Let $\{(i_1,j_1),(i_2,j_2),\ldots,(i_r,j_r)\}$ denote the resulting sorted set. We assume, without loss of generality, that the set contains no duplicate pairs (as they can be safely deleted) and no two pairs have the same first entry. This assumption is justified by the fact that for all $(i,j),(i',j') \in \varphi(a)$ we have $i+j = i'+j' = w_A(a)$. We add all edges (if they do not already exist) between K_u and vertices $\{p^1_{i_1}, p^2_{i_1}\}$ and all edges between K_v and vertices $\{p^1_{i_r}, p^2_{i_r}\}$. We call the pair p_{i_1} the *left pair* of $\mathcal{P}_{uv}$ and denote it by $\mathsf{left}(\mathcal{P}_{uv})$. Similarly, we call the pair p_{i_r} the *right pair* of $\mathcal{P}_{uv}$ and denote it by $\mathsf{right}(\mathcal{P}_{uv})$. Finally, for each two consecutive entries (i,j) and (i',j') we add all edges between $\{p^1_i, p^2_i\}$ and $\{p^1_{i'}.p^2_{i'}\}$. This completes the construction of the graph G (see Figure 6).

We first show that the treewidth of G is bounded by a polynomial in k. To do so, we need to recall the notion of a *mixed search game* [13]. In a mixed search game, a graph G represents a "system of tunnels". Initially, all edges are contaminated by a gas. An edge is cleared by placing searchers at both its endpoints simultaneously or by sliding a searcher along the edge. A cleared edge is re-contaminated if there is a path from an uncleared edge to the cleared edge without any searchers on its vertices or edges. A search is a sequence of operations that can be of the following types:

- placement of a new searcher on a vertex;
- removal of a searcher from a vertex;
- sliding a searcher on a vertex along an incident edge and placing the searcher on the other end.

A search strategy is winning if after its termination all edges are cleared. The *mixed search number* of a graph G, denoted by $\mathsf{ms}(G)$, is the minimum number

of searchers required for a winning strategy of mixed searching on G. It is well-known [14] that $\mathsf{tw}(G) \leq \mathsf{pw}(G) \leq \mathsf{ms}(G) \leq \mathsf{pw}(G) + 1$.

Proposition 4. $\mathsf{tw}(G) = k^{\mathcal{O}(1)}$.

Proof. We give a mixed search strategy to clean the graph G using $k^{\mathcal{O}(1)} + \mathcal{O}(1)$ searchers. First, we put $2(k+3)|V(D)|$ searchers on all vertices in K_v, for all $v \in V(D)$. Since $k = 2|A(D)|$ and $|V(D)| = \mathcal{O}(|A(D)|) = \mathcal{O}(k)$, we have $2(k+3)|V(D)| = k^{\mathcal{O}(1)}$. Those searchers will remain fixed until the end of the cleaning process. Note that vertices in I_v, for any $v \in V(D)$, have no neighbors outside of K_v. Moreover, vertices in a chain H_{uv}, for each arc $a = uv \in A(D)$, have no neighbors outside of $V(H_{uv}) \cup V(K_u) \cup V(K_v)$. Therefore, it remains to show how to clean each H_{uv} using a constant number of additional searchers; edges between I_v and K_v can be cleaned using one additional searcher.

We search the rest of the graph in phases, one phase per H_{uv} and reusing the searchers from the previous phase. For each H_{uv}, we reuse $\mathcal{O}(1)$ searchers. We start the phase by placing searchers on $\{p^1_{i_1}, p^2_{i_1}, p^1_{i_2}, p^2_{i_2}\}$. Since vertices inside border walls are independent and any two border walls are separated by a bridging pair, using an additional 5 searchers we can clean the graph induced by all bridging pairs $\{p_i \mid 0 \leq i \leq i_2\}$ and all border walls $\{b_i \mid 1 \leq i \leq i_2\}$. We now proceed sequentially (from left to right) as follows. We remove the two searchers on $\{p^1_{i_1}, p^2_{i_1}\}$, place them on $\{p^1_{i_3}, p^2_{i_3}\}$, and clean the graph induced by all bridging pairs $\{p_i \mid i_2 \leq i \leq i_3\}$ and all border walls $\{b_i \mid i_2 \leq i \leq i_3\}$. We repeat until H_{uv} is cleaned. It is not hard to see that after the last round all the graph is cleaned. Since $\mathsf{tw}(G) \leq \mathsf{pw}(G) \leq \mathsf{ms}(G) \leq \mathsf{pw}(G) + 1$, the proposition follows. □

Below we prove a series of lemmas that allows us to transform any solution S to a constructed instance (G, k, μ) of CNC into an "equally good" solution S' having some "nice" structural properties. We let $\mathcal{C}(G-S) = \{C_1, \ldots, C_\ell\}$ denote the set of connected components in $G - S$. We classify a connected component $C \in \mathcal{C}(G-S)$ into one of three types. We say C is a *small* component whenever C does not contain any vertices from K_v or I_v, for all $v \in V(D)$. We say C is a *large* component whenever C intersects with at least two cliques K_u and K_v, $u, v \in V(D)$. We say that C is a *medium* component otherwise. Note that, for any $v \in V(D)$, any solution of size k cannot separate $G[V(I_v) \cup V(K_v)]$ into two or more components. Therefore, if $\mathcal{C}(G-S)$ consists of only medium components then $|\mathcal{C}(G-S)|$ is exactly $|V(D)|$ and S includes exactly one bridging pair from each chain H_{uv}, $uv \in A(D)$.

Lemma 8. *Let S be a solution to (G, k, μ) and let $\mathcal{C}(G-S) = \{C_1, \ldots, C_\ell\}$. If $|S \cap \bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))| > 0$ then there exists a solution S' such that $|S'| = |S|$, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$, and $|S' \cap \bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))| = |S \cap \bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))| - 1$.*

Proof. Let w be a vertex in $S \cap (V(I_u) \cup V(K_u))$, for some $u \in V(D)$. Note that (by construction) $|V(I_u)| > k$ and $|V(K_u)| > k$. Moreover, for all $w_1, w_2 \in V(I_u)$

(or $V(K_u)$), we have $N_G(w_1) = N_G(w_2)$. Therefore, since $|S| = k$, we have $|\mathcal{C}(G-S)| = |\mathcal{C}((G-S) \cup \{w\})|$. In other words, if $w \in V(I_u)$ or $w \in V(K_u)$ then there exists at least one vertex $w' \notin S$ such that $w' \in V(I_u)$ or $w' \in V(K_u)$, respectively. Let $C_{w'} \in \mathcal{C}(G-S)$ denote the component in $G-S$ containing w'. Note that $C_{w'}$ is either a medium or a large component, since a small component (by definition) does not intersect with $\bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))$. If $C_{w'}$ is a large component then it must contain a vertex w'' which belongs to some chain H_{uv}, for some $v \in V(D)$. We let $S' = (S \setminus \{w\}) \cup \{w''\}$. It is not hard to see that S' does in fact satisfy all the required properties (since $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} = \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$). If $C_{w'}$ is a medium component then we have two cases to consider. If we can find a w'' (belonging to some chain) then the same replacement argument as above holds. Otherwise, we know that the size of $V(C_{w'})$ is at most $|V(I_u)| + |V(K_u)| = W_u + 2(k+3) = (k+3)(w_{big} + w_{vmax} - w_V(u) + 2) \leq (k+3)(w_{big} + w_{vmax} + 2)$. However, since S does not include exactly two vertices from each chain, we know that $\mathcal{C}(G-S)$ must include at least one large component, say C'', of size at least $2(k+3)w_{big}$. Replacing w with a vertex $w'' \in V(C'') \cap H_{u'v'}$, for some $u'v' \in V(D)$, produces the required set S' (recall that we assume $w_{vmax} \geq 2|A(D)| + 1 = k + 1 \geq 101$ and therefore $(k+3)w_{big} > (k+3)(w_{vmax}+2)$). □

By repeated applications of Lemma 8, we can assume that a solution S does not intersect with $V(I_u) \cup V(K_u)$, for all $u \in V(D)$. In what follows, we always assume that S satisfies this property. Using similar arguments, we can show that S also does not intersect with any border walls.

Lemma 9. *Let S be a solution to (G, k, μ) and let $\mathcal{C}(G-S) = \{C_1, \ldots, C_\ell\}$. If $|S \cap \bigcup_{uv \in A(D)} \mathcal{B}_{uv}| > 0$ then there exists a solution S' such that $|S'| = |S|$, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$, $|S' \cap \bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))| = |S \cap \bigcup_{u \in V(D)}(V(I_u) \cup V(K_u))|$, and $|S' \cap \bigcup_{uv \in A(D)} \mathcal{B}_{uv}|$ is strictly less than $|S \cap \bigcup_{uv \in A(D)} \mathcal{B}_{uv}|$.*

Proof. Let w be a vertex in $S \cap b$, where $b \in \mathcal{B}_{uv}$ for some $uv \in A(D)$. Recall that (by construction) $|b| = k+1$ and, for all $w_1, w_2 \in b$, we have $N_G(w_1) = N_G(w_2)$. Therefore, since $|S| = k$, we have $|\mathcal{C}(G-S)| = |\mathcal{C}((G-S) \cup \{w\})|$. In other words, there exists at least one vertex $w' \notin S$ such that $w' \in b$. Let $C_{w'} \in \mathcal{C}(G-S)$ denote the component in $G-S$ containing w'. If $C_{w'}$ is a medium or large component then we can always find a vertex w'' from some bridging pair, i.e. $w'' \in \mathcal{P}_{uv}$, to replace w and obtain S'. If $C_{w'}$ is a small component then either $|V(C_{w'})| = 1$, in which case we can replace w by any bridging pair vertex, or $C_{w'}$ contains some vertex $w'' \in \mathcal{P}_{uv}$, as needed. □

Lemmas 8 and 9 imply that we can always assume that S includes vertices from bridging pairs only. We say that a solution S *splits* a bridging pair $\{p^1, p^2\}$ if $|S \cap \{p^1, p^2\}| = 1$. We now proceed to showing that S does not split any bridging pair. We use $\mathsf{split}(G, S)$ to denote the number of bridging pairs split by S in G.

Lemma 10. *Let S be a solution to (G, k, μ) such that $S \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$ and let $\mathcal{C}(G - S) = \{C_1, \ldots, C_\ell\}$. If $\mathsf{split}(G, S) > 0$ then there exists a solution S' such that $|S'| = |S|$, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$, $S' \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$, and $\mathsf{split}(G, S') < \mathsf{split}(G, S)$.*

Proof. Given that k is even, we know that $\mathsf{split}(G, S)$ must also be even. Assume that S splits two bridging pairs $\{p^1, p^2\}$ and $\{q^1, q^2\}$. Without loss of generality, we assume that $p^1, q^1 \in S$, $p^2, q^2 \notin S$, $p_2 \in V(C_p)$, $q_2 \in V(C_q)$, and $|V(C_p)| \leq |V(C_q)|$, where $C_p, C_q \in \mathcal{C}(G - S)$. We let $S' = (S \setminus \{p_1\}) \cup \{q_2\}$. It is not hard to see that regardless of whether $C_p = C_q$ or not, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$, as needed. □

Lemma 11. *Let S be a solution to (G, k, μ) such that $S \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$ and $\mathsf{split}(G, S) = 0$. Let $\mathcal{C}(G - S) = \{C_1, \ldots, C_\ell\}$. Assume that $|S \cap \mathcal{P}_{uv}| = 2x \geq 10$, for some $uv \in A(D)$, and hence there exists $u_1v_1, \ldots, u_{x-1}v_{x-1} \in A(D)$ such that $|S \cap \mathcal{P}_{u_iv_i}| = 0$, for $i \in [x - 1]$. Then, there exists S' such that $|S'| = |S|$, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$, $S' \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$, $\mathsf{split}(G, S') = 0$, $\mathsf{left}(\mathcal{P}_{uv}) \cup \mathsf{right}(\mathcal{P}_{uv}) \cup \mathsf{first}(\mathcal{P}_{uv}) \cup \mathsf{last}(\mathcal{P}_{uv}) \subseteq S'$, $|S' \cap \mathcal{P}_{uv}| = 8$, and $|S' \cap \mathcal{P}_{u_iv_i}| = 2$, for $i \in [x - 1]$.*

Proof. Recall the construction of $\mathcal{P}_{uv} = \{p_0, \ldots, p_{w_A(uv)}\}$. We add all edges between K_u and pair $p_0 = \mathsf{first}(\mathcal{P}_{uv})$ and we add all edges between K_v and $p_{w_A(uv)} = \mathsf{last}(\mathcal{P}_{uv})$. Next, we sort all entries $(i, j) \in \varphi(uv)$ in increasing order based on the first coordinate. Let $\{(i_1, j_1), (i_2, j_2), \ldots, (i_r, j_r)\}$ denote the resulting sorted set. We add all edges between K_u and pair $p_{i_1} = \mathsf{left}(\mathcal{P}_{uv})$ and all edges between K_v and pair $p_{i_r} = \mathsf{right}(\mathcal{P}_{uv})$. Finally, for each two consecutive entries (i, j) and (i', j'), we add all edges between p_i and $p_{i'}$. Let us assume that $0 \neq i_1 \neq i_r \neq w_A(uv)$, as the same arguments hold when that is not case. Since $|S \cap \mathcal{P}_{uv}| \geq 10$, we know that S includes at least 5 pairs from $\mathcal{P}_{uv}$. Let $S' = (S \setminus \mathcal{P}_{uv}) \cup \mathsf{left}(\mathcal{P}_{uv}) \cup \mathsf{right}(\mathcal{P}_{uv}) \cup \mathsf{first}(\mathcal{P}_{uv}) \cup \mathsf{last}(\mathcal{P}_{uv}) \cup \mathsf{left}(\mathcal{P}_{u_1v_1}) \cup \ldots \cup \mathsf{left}(\mathcal{P}_{u_{x-1}v_{x-1}})$. It remains to show that $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$. Note that in the graph $G[V(K_u) \cup V(K_v) \cup H_{uv}] - S'$ we have five connected components, where two of those components are exactly K_u and K_v. Hence, the total number of connected pairs that are introduced by removing some of the pairs in $S \cap \mathcal{P}_{uv}$ is at most $\binom{(k+3)w_{amax}}{2}$. However, for each pair $\mathsf{left}(\mathcal{P}_{u_iv_i})$ that we add to S' the number of connected pairs decreases by at least $(k + 3)w_{big} > \binom{(k+3)w_{amax}}{2}$ (recall that $w_{vmax} \geq k + 1$ and $k \geq 100$). □

Since k is even, and S includes vertices from bridging pairs only, and S does not split any bridging pair, we know (from Lemma 11 and the fact that $\mathsf{split}(G,S) = 0$) that, for all $uv \in A(D)$, $|S \cap \mathcal{P}_{uv}| \in \{0, 2, 4, 6, 8\}$ (as otherwise S would have to split at least one bridging pair). The next lemma shows that we can in fact guarantee that $|S \cap \mathcal{P}_{uv}| = 2$, for all $uv \in A(D)$.

Lemma 12. *Let S be a solution satisfying the following properties: (1) $S \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$; (2) $\mathsf{split}(G,S) = 0$; (3) $|S \cap \mathcal{P}_{uv}| \in \{0, 2, 4, 6, 8\}$, for all*

$uv \in A(D)$; (4) If $|S \cap \mathcal{P}_{uv}| = 8$, for $uv \in A(D)$, then $\mathsf{left}(\mathcal{P}_{uv}) \cup \mathsf{right}(\mathcal{P}_{uv}) \cup \mathsf{first}(\mathcal{P}_{uv}) \cup \mathsf{last}(\mathcal{P}_{uv}) \subseteq S$. Then, there exists a solution S' satisfying the following properties: (i) $S' \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$; (ii) $\mathsf{split}(G,S') = 0$; (iii) $|S' \cap \mathcal{P}_{uv}| = 2$, for all $uv \in A(D)$.

Proof. Let s_0, s_1, s_2, s_3, and s_4 denote the cardinality of $\{uv \in A(D) \mid |S \cap \mathcal{P}_{uv}| = 0\}$, $\{uv \in A(D) \mid |S \cap \mathcal{P}_{uv}| = 2\}$, $\{uv \in A(D) \mid |S \cap \mathcal{P}_{uv}| = 4\}$, $\{uv \in A(D) \mid |S \cap \mathcal{P}_{uv}| = 6\}$, and $\{uv \in A(D) \mid |S \cap \mathcal{P}_{uv}| = 8\}$, respectively. Note that $s_0 + s_1 + s_2 + s_3 + s_4 = |A(D)|$ and $2s_1 + 4s_2 + 6s_3 + 8s_4 = k$. Hence, $2(s_0 + s_1 + s_2 + s_3 + s_4) = 2s_1 + 4s_2 + 6s_3 + 8s_4$ and $s_0 = s_2 + 2s_3 + 3s_4$. In what follows, we assume that $s_0 = s_2 + 2s_3 + 3s_4 > 0$; as otherwise we are done.

We define s_0^u, s_1^u, s_2^u, s_3^u, and s_4^u as follows:

- $s_0^u = |\{vw \in A(D) \mid (u = v \vee u = w) \wedge |S \cap \mathcal{P}_{vw}| = 0\}|$;
- $s_1^u = |\{vw \in A(D) \mid (u = v \vee u = w) \wedge |S \cap \mathcal{P}_{vw}| = 2\}|$;
- $s_2^u = |\{vw \in A(D) \mid (u = v \vee u = w) \wedge |S \cap \mathcal{P}_{vw}| = 4\}|$;
- $s_3^u = |\{vw \in A(D) \mid (u = v \vee u = w) \wedge |S \cap \mathcal{P}_{vw}| = 6\}|$;
- $s_4^u = |\{vw \in A(D) \mid (u = v \vee u = w) \wedge |S \cap \mathcal{P}_{vw}| = 8\}|$.

Since $s_0 > 0$ and $\sum_{u \in V(D)} s_0^u = 2s_0$, we know that there exists at least one vertex $u \in V(D)$ such that $s_0^u > 0$. We claim that there exists a vertex $u \in V(D)$ such that $0 < s_0^u \leq s_2^u + 2s_3^u + 3s_4^u$. Assume otherwise. That is, assume that for all $u \in V(D)$ for which $s_0^u > 0$ we have $s_2^u + 2s_3^u + 3s_4^u > s_0^u$. Since $\sum_{u \in V(D)} s_i^u = 2s_i$, $i \in \{0,1,2,3,4\}$, we have $\sum_{u \in V(D)} s_2^u + \sum_{u \in V(D)} 2s_3^u + \sum_{u \in V(D)} 3s_4^u > \sum_{u \in V(D)} s_0^u$, which is equivalent to $2s_2 + 4s_3 + 6s_4 > 2s_0$, a contradiction to the fact that $s_0 = s_2 + 2s_3 + 3s_4$. Hence, the claim follows.

Now, we let $u \in V(D)$ such that $s_0^u \leq s_2^u + 2s_3^u + 3s_4^u$. We create a new solution S' as follows. We let $S' = S \setminus \bigcup_{vw \in A(D) \wedge (u = v \vee u = w)} \mathcal{P}_{vw}$. Then, for each $vw \in A(D)$, if $u = v$ we add $\mathsf{right}(\mathcal{P}_{uw})$ to S' and if $u = w$ we add $\mathsf{left}(\mathcal{P}_{vu})$ to S'. If we are left with $|S'| < k$ then we pick the remaining pairs from chains whose intersection with S' is empty. Note that, after this replacement, we reduce the number of large components by at least one. Therefore, $\sum_{C' \in \mathcal{C}(G-S')} \binom{C'}{2} \leq \sum_{C \in \mathcal{C}(G-S)} \binom{C}{2}$. In other words, the number of connected pairs increases by at most $\binom{(k+3)(w_{big} + w_{vmax} + 2) + k(k+3)w_{amax}}{2}$. However, since we reduce the number of large components, the number of connected pairs decreases by at least $\binom{2(k+3)w_{big}}{2}$. The lemma follows by repeating the replacement procedure as long as we can find a vertex u with $0 < s_0^u \leq s_2^u + 2s_3^u + 3s_4^u$. When no such vertex exists, it must be the case that $|S' \cap \mathcal{P}_{uv}| = 2$, for all $uv \in A(D)$. □

We are now ready to prove the correctness of the reduction, which is implied by Lemmas 13 and 14 below.

Lemma 13. *If $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N} \times \mathbb{N}})$ is a yes-instance of* SumCSP *then (G, k, μ) is a yes-instance of* CNC.

PROOF. Let $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ be a solution to the SUMCSP instance. We construct a solution S to the CNC instance by picking one bridging pair from each chain as follows. Initially, we set $S = \emptyset$. For each $uv \in A(D)$, we let $\mathcal{P}_{uv} = \{p_0, \ldots, p_{w_A(uv)}\}$, we let $\rho(uv) = (x_{uv}, y_{uv})$, and we set $S = S \cup p_{x_{uv}}$. It is not hard to see that $G - S$ consists of exactly $|V(D)|$ components (as we pick one bridging pair from each chain). We associate each component with some vertex $u \in V(D)$. The size of each component is exactly $|V(K_u)| + |V(I_u)| + \sum_{v \in N^+(u)}(k+3)x_{uv} + \sum_{v \in N^-(u)}(k+3)y_{vu} = |V(K_u)| + |V(I_u)| + (k+3)w_V(u) = 2(k+3) + (k+3)(w_{big} + w_{vmax} - w_V(v)) + (k+3)w_V(u) = (k+3)(w_{big} + w_{vmax} + 2) = W^\star$. □

Lemma 14. *If (G, k, μ) is a yes-instance of* CNC *then $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N}\times\mathbb{N}})$ is a yes-instance of* SUMCSP.

PROOF. Let S be a solution to (G, k, μ). From Lemmas 8 to 12, we know that $\mathcal{C}(G-S)$ consists of only medium components. In other words, $S \subseteq \bigcup_{uv \in A(D)} \mathcal{P}_{uv}$, $\mathsf{split}(G,S) = 0$, and $|S \cap \mathcal{P}_{uv}| = 2$, for all $uv \in A(D)$. Hence, the number of components in $G - S$ is exactly $|V(D)|$. Let $\mathcal{C}(G - S) = \{C_1, \ldots, C_{|V(D)|}\}$. Recall that $W^\star = (k+3)(w_{big} + w_{vmax} + 2)$ and $\mu = |V(D)| \cdot \binom{W^\star}{2}$. Therefore, we have $\sum_{C \in \mathcal{C}(G-S)} \binom{C}{2} \leq |V(D)| \cdot \binom{W^\star}{2}$. Applying Proposition 1, we know that each component in $\mathcal{C}(G - S)$ must have $W^\star$ vertices. We associate each component with some vertex $u \in V(D)$. Note that K_u contains $2(k+3)$ vertices and I_u contains $(k+3)(w_{big} + w_{vmax} - w_V(v))$ vertices. Therefore, $W^\star - |V(K_u)| - |V(I_u)| = (k+3)w_V(u)$. Since each chain H_{uv} or H_{vu}, v is a neighbor of u, contributes $(k+3)x$ vertices, for some x, to the component associated with u, the sum of those contributions must equal $(k+3)w_V(u)$.

We claim that this implies that there exists $\rho : A(D) \to \mathbb{N} \times \mathbb{N}$ such that for each $uv, vu \in A(D)$, $\rho(uv) \in \varphi(uv)$, $\rho(vu) \in \varphi(vu)$ and $\sum_{v \in N^+(u)} \mathsf{fir}(\rho(uv)) + \sum_{v \in N^-(u)} \mathsf{sec}(\rho(vu)) = w_V(u)$. Let us recall the relevant parts of the construction. H_{uv} consists of $w_A(a)+1$ bridging pairs of vertices $\mathcal{P}_{uv} = \{p_0, \ldots, p_{w_A(uv)}\}$, where each pair $p_i \in \mathcal{P}_{uv}$ consists of two (independent) vertices $\{p_i^1, p_i^2\}$. Moreover, we have $w_A(a)$ border walls $\mathcal{B}_{uv} = \{b_1, \ldots, b_{w_A(a)}\}$, each consisting of $k+1$ (independent) vertices. We add all edges between K_u and pair p_0 and we add all edges between K_v and $p_{w_A(uv)}$. Next, we add all edges between p_{i-1} and b_i and all edges between b_i and p_i, for $i \in [w_A(a)]$. Then, we sort all entries $(i, j) \in \varphi(a)$ in increasing order based on the first coordinate. Let $\{(i_1, j_1), (i_2, j_2), \ldots, (i_r, j_r)\}$ denote the resulting sorted set. We add all edges (if they do not already exist) between K_u and vertices $\{p_{i_1}^1, p_{i_1}^2\}$ and all edges between K_v and vertices $\{p_{i_r}^1, p_{i_r}^2\}$. Finally, for each two consecutive entries (i, j) and (i', j') we add all edges between $\{p_i^1, p_i^2\}$ and $\{p_{i'}^1.p_{i'}^2\}$.

Consider a component in $\mathcal{C}(G - S)$ associated with some clique K_u. Since K_u and K_v belong to different component in $\mathcal{C}(G - S)$, $\mathsf{split}(G,S) = 0$, and $|S \cap \mathcal{P}_{uv}| = 2$, for all $uv \in A(D)$, it must be the case that for the bridging pair $p_{i_\star} = S \cap \mathcal{P}_{uv}$ we have $i_\star \in \{i_1, i_2, \ldots, i_r\}$; as otherwise K_u and K_v would belong to the same component in $G - S$. This implies that, with each

$uv, vu \in A(D)$, we can associate a pair $(i_\star, j_\star) \in \{(i_1, j_1), (i_2, j_2), \ldots, (i_r, j_r)\}$. Equivalently, we have $\rho(uv) \in \varphi(uv)$ and $\rho(vu) \in \varphi(vu)$. It remains to show that $\sum_{v \in N^+(u)} \mathsf{fir}(\rho(uv)) + \sum_{v \in N^-(u)} \mathsf{sec}(\rho(vu)) = w_V(u)$. We know that each outgoing edge uv contributes $(k+3)i_\star^v$ vertices to the component associated with u and each incoming edge vu contributes $(k+3)j_\star^v$ vertices to the component associated with u; each border wall contributes $k+1$ vertices and each bridging pair contributes 2 vertices. The total sum of those contributions must equal $(k+3)w_V(u)$. Therefore, $\sum_{v \in N^+(u)}(k+3)i_\star^v + \sum_{v \in N^-(u)}(k+3)j_\star^v = (k+3)w_V(u)$. Dividing both sides by $k+3$ completes the proof. □

Theorem 15. CNC *is W[1]-hard when parameterized by solution size and the treewidth of the input graph.*

Proof. Let (G, k, μ) be the constructed instance of CNC for the given instance $(D, w_V : V(D) \to \mathbb{N}, w_A : A(D) \to \mathbb{N}, \varphi : A(D) \to 2^{\mathbb{N} \times \mathbb{N}})$ of SumCSP. An easy trace of the construction shows that it can be accomplished in time polynomial in $|V(D)|$, $|A(D)|$, w_{vmax}, and w_{amax}. Recall that $w_{vmax} = \mathsf{max}_{v \in V(D)}(w_V(v))$ denotes the maximum weight of a vertex in D and $w_{amax} = \mathsf{max}_{a \in A(D)}(w_A(a))$ denotes the maximum weight of an arc in D. Since $k = 2|A(D)|$ and $\mathsf{tw}(G) = k^{\mathcal{O}(1)}$, combining all of the above with Lemmas 13 and 14 and the W[1]-hardness of SumCSP (parameterized by $|A(D)|$) completes the proof. □

5. Conclusion

We have showed that the Critical Node Cut problem is W[1]-hard when parameterized by solution size and the treewidth of the input graph, answering the remaining question left open by Hermelin et al.[1]. In doing so, we introduced the SumCSP problem, which we believe can be a useful starting point for showing hardness results of the same nature, i.e., when the treewidth of the graph is part of the parameter. Although we have not yet been able to find further applications of the SumCSP problem, it would be interesting to see whether known hardness results can have "simpler" proofs if SumCSP is used as a starting point.

References

[1] D. Hermelin, M. Kaspi, C. Komusiewicz, B. Navon, Parameterized complexity of critical node cuts, Theoretical Computer Science 651 (2016) 62–75.

[2] M. Ventresca, Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem, Computers & Operations Research 39 (2012) 2763–2775.

[3] M. Ventresca, D. Aleman, A derandomized approximation algorithm for the critical node detection problem, Computers & Operations Research 43 (2014) 261–270.

[4] M. Di Summa, A. Grosso, M. Locatelli, Complexity of the critical node problem over trees, Computers & Operations Research 38 (2011) 1766–1774.

[5] B. Addis, M. D. Summa, A. Grosso, Removing critical nodes from a graph: complexity results and polynomial algorithms for the case of bounded treewidth, Optimization online (www.optimization-online.org) (2011).

[6] R. G. Downey, M. R. Fellows, Parameterized complexity, Springer-Verlag, 1997.

[7] D. Marx, Can you beat treewidth?, Theory of Computing 6 (2010) 85–112.

[8] H. L. Bodlaender, D. Lokshtanov, E. Penninkx, Planar capacitated dominating set is $W[1]$-hard, in: IWPEC, volume 5917 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 50–60.

[9] R. Diestel, Graph Theory, 4th Edition, volume 173 of *Graduate texts in mathematics*, Springer, 2012.

[10] J. Flum, M. Grohe, Parameterized Complexity Theory, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2006.

[11] R. Niedermeier, Invitation to fixed-parameter algorithms, Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press, 2006.

[12] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.

[13] D. Lokshtanov, D. Marx, S. Saurabh, Known algorithms on graphs of bounded treewidth are probably optimal, in: Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2011, pp. 777–789. URL: `http://dl.acm.org/citation.cfm?id=2133036.2133097`.

[14] A. Takahashi, S. Ueno, Y. Kajitani, Mixed searching and proper-path-width, Theoretical Computer Science 137 (1995) 253–268.