On Structural Parameterizations for the 2-Club Problem[☆]

Sepp Hartung[a], Christian Komusiewicz[a], André Nichterlein[a], Ondřej Suchý[b,1]

[a] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, D-10587 Berlin, Germany
[b] Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic

Abstract

The NP-hard 2-Club problem is, given an undirected graph $G = (V, E)$ and $\ell \in \mathbb{N}$, to decide whether there is a vertex set $S \subseteq V$ of size at least ℓ such that the induced subgraph $G[S]$ has diameter at most two. We make progress towards a systematic classification of the complexity of 2-Club with respect to a hierarchy of prominent structural graph parameters. First, we present the following tight NP-hardness results: 2-Club is NP-hard on graphs that become bipartite by deleting one vertex, on graphs that can be covered by three cliques, and on graphs with *domination number* two and *diameter* three. Then, we consider the parameter *h-index* of the input graph. The study of this parameter is motivated by real-world instances and the fact that 2-Club is fixed-parameter tractable when parameterized by the larger parameter *maximum degree*. We present an algorithm that solves 2-Club in $|V|^{f(k)}$ time with k being the *h-index* of G. By showing W[1]-hardness for this parameter, we provide evidence that the above algorithm cannot be improved to a fixed-parameter algorithm. Furthermore, the reduction used for this hardness result can be modified to show that 2-Club is NP-hard if the input graph has constant *degeneracy*. Finally, we show that 2-Club is fixed-parameter tractable when parameterized by *distance to cographs*.

Keywords: clique relaxations, cohesive subnetworks, social network analysis, fixed-parameter tractability, parameter hierarchy, multivariate complexity analysis

1. Introduction

The identification of cohesive subnetworks is an important task in the analysis of social and biological networks, since these subnetworks are likely to represent communities or functional subnetworks within the large network. The natural cohesiveness requirement is to demand that the subnetwork is a complete graph, a clique. However, this requirement is often too restrictive and thus relaxed definitions of cohesive graphs such as s-cliques [1],

[☆] An extended abstract of this paper appeared in *Proceedings of the 39th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'13)*, Jan. 2013, volume 7741 of LNCS, pages 233-243, Springer, 2013 [21].

Email addresses: sepp.hartung@tu-berlin.de (Sepp Hartung), christian.komusiewicz@tu-berlin.de (Christian Komusiewicz), andre.nichterlein@tu-berlin.de (André Nichterlein), ondrej.suchy@fit.cvut.cz (Ondřej Suchý)

[1] The research was partially supported by the grant 14-13017P of the Czech Science Foundation.

s-plexes [34], and s-clubs [27] have been proposed. In this work, we study the problem of finding large s-clubs within the input network. An s-club is a vertex set that induces a subgraph of diameter at most s. Thus, s-clubs are distance-based relaxations of cliques, which are vertex sets inducing diameter-one graphs. For constant $s \geq 1$, the problem of finding s-clubs is defined as follows.

s-Club
Input: An undirected graph $G = (V, E)$ and $\ell \in \mathbb{N}$.
Question: Is there a vertex set $S \subseteq V$ of size at least ℓ such that $G[S]$ has diameter at most s?

In this work, we study the computational complexity of 2-Club, that is, the special case of $s = 2$. The restriction to this special case is further motivated by the following two considerations. First, 2-Club is an important special case concerning the applications: For biological networks, 2-clubs and 3-clubs have been identified as the most reasonable diameter-based relaxations of cliques [30]. Further, Balasundaram et al. [4] also proposed to compute 2-clubs and 3-clubs for analyzing protein interaction networks. 2-Club also has applications in the analysis of social networks [26]. Consequently, all experimental studies concentrate on finding 2- and 3-clubs [2, 4, 9, 11, 12, 20, 25]. Second, 2-Club is the most basic variant of s-Club that is different from the Clique problem which is equivalent to 1-Club. For example, being a clique is a hereditary graph property, that is, it is closed under vertex deletion. In contrast, being a 2-club is not hereditary, since deleting vertices can increase the diameter of a graph. Hence, it is interesting to spot differences in the computational complexity of the two problems.

In the spirit of multivariate algorithmics [17, 24, 29], we aim to describe how structural properties of the input graph determine the computational complexity of 2-Club. We want to determine sharp boundaries between tractable and intractable special cases of 2-Club, and whether some graph properties, especially those motivated by the structure of social and biological networks, can be exploited algorithmically. By arranging the parameters in a hierarchy (ranging from large to small parameters) we draw a border line between tractability and intractability to obtain a systematic view on "stronger parameterizations" (see Section 1.2 for a formal introduction). Most importantly, this hierarchy allows to transfer tractability and intractability results between parameters and thus helps in "navigating" through the parameters. Using the hierarchy we can deduce many (in)tractability results from relatively few algorithms and reductions. Hence, even if some results are obtained for seemingly uncommon or unmotivated parameters, they can imply results for several natural parameters. We refer to Komusiewicz and Niedermeier [24] for a further discussion of the parameter hierarchy and for examples of its usage to "decompose" the computational intractability of NP-hard problems. A similar approach was followed for other hard graph problems such as Odd Cycle Transversal [22] and for the computation of the pathwidth of a graph [7].

The structural properties that we consider, called structural graph parameters, are usually described by integers; well-known examples of such parameters are the maximum degree or the treewidth of a graph. Our results use the classic framework of NP-hardness as well as the framework of parameterized complexity to show (parameterized) tractability and intractability of 2-Club when parameterized by the structural graph parameters under consideration. That is, for some graph parameters we show that 2-Club becomes

NP-hard in case of constant parameter values, whereas for other graph parameters we show fixed-parameter (in)tractability.

1.1. Related Work

For all $s \geq 1$, s-Club is NP-complete on graphs of diameter $s+1$ [4]; 2-Club is NP-complete even on split graphs and, thus, also on chordal graphs [3]. In contrast, 2-Club is solvable in polynomial time on bipartite graphs, on trees, and on interval graphs [33]. Golovach et al. [19] consider the complexity of s-Club in special graph classes. For instance they prove polynomial-time solvability of s-Club on chordal bipartite, strongly chordal and distance hereditary graphs. Additionally, on a superclass of these graph classes, called weakly chordal graphs, s-Club is polynomial-time solvable for odd s and NP-hard for even s [19]. Mahdavi and Balasundaram [25] have shown that it is NP-hard to decide whether a given s-club is maximal for each fixed $s \geq 2$.

The s-Club problem is well-understood from the viewpoint of approximation algorithms [3]: It is NP-hard to approximate s-Club within a factor of $n^{\frac{1}{2}-\epsilon}$ for any $\epsilon > 0$. On the positive side, it has been shown that a largest set consisting of a vertex together with all vertices within distance $\lfloor \frac{s}{2} \rfloor$ is a factor $n^{\frac{1}{2}}$ approximation for even $s \geq 2$ and a factor $n^{\frac{2}{3}}$ approximation for odd $s \geq 3$. Several heuristics [8, 11, 12], integer linear programming formulations [2, 4, 9], fixed-parameter algorithms [20, 32], and branch-and-bound algorithms [9] have been proposed and experimentally evaluated [20, 25].

From the viewpoint of parameterized algorithmics, 1-Club is equivalent to Clique and thus W[1]-hard when parameterized by ℓ [14]. In contrast, for all $s \geq 2$, s-Club parameterized by ℓ is fixed-parameter tractable [32]. Furthermore, it is fixed-parameter tractable when parameterized by the treewidth of G [33]. Additionally, a search tree algorithm that branches into the two possibilities to delete one of two vertices with distance more than s achieves a running time of $O(2^{n-\ell} \cdot nm)$ for the *dual parameter* $n - \ell$ which measures the *distance to an s-club* [32].[2] This algorithm cannot be improved to $(2-\epsilon)^{n-\ell} \cdot n^{O(1)}$ for any $\epsilon > 0$ if the strong exponential time hypothesis is true [20]. Interestingly, Chang et al. [12] proved that the same search tree algorithm runs in $O(1.62^n)$ time where n is the number of vertices in G .

The main observation behind the fixed-parameter algorithm for ℓ is that the closed neighborhood $N[v]$ of any vertex v is an s-club for $s \geq 2$. Hence, the maximum degree Δ in non-trivial instances is less than $\ell - 1$. In yes-instances, however, it also holds that $\ell \leq \Delta^s + 1$. Thus, for constant s, fixed-parameter tractability when parameterized by ℓ also implies fixed-parameter tractability when parameterized by the maximum degree of G. Moreover, s-Club does not admit a polynomial kernel when parameterized by ℓ (unless NP $\subseteq$ coNP/poly) [32]. Interestingly, taking for each vertex the vertex itself together with all other vertices that are in distance at most s forms a so-called Turing-kernel with at most k^2-vertices for even s and at most k^3-vertices for odd s [32]. In companion work [20], we considered further structural parameters: We presented a fixed-parameter algorithm for the parameter treewidth and polynomial kernels for the parameters (size of a) *feedback edge set* and the *cluster editing number*. Additionally, we

[2] Schäfer et al. [32] considered finding an s-club of size *exactly* ℓ. The claimed fixed-parameter tractability when parameterized by $n - \ell$ however only holds for the problem of finding an s-club of size *at least* ℓ. The other fixed-parameter tractability results hold for both variants.

showed the non-existence of a polynomial kernel and that the simple search tree algorithm for the *dual parameter* $n - \ell$ is asymptotically optimal. Somewhat in contrast to this negative result, we showed that an implementation of the branching algorithm for the parameter $n - \ell$ combined with the Turing-kernelization is among the best-performing algorithms on real-world and on synthetic instances [20].

1.2. Structural Parameters

We next discuss structural parameters and formally define those that we consider in this work (see Figure 1 for an illustration of their relations). We remark that there is no formal definition of what is considered to be a structural parameterization. Intuitively, it is a quantitative measurement of some structure in the input I which is rather independent of the problem and of the question whether I is a yes- or no-instance.

We next define the structural parameters under consideration. For a set of graphs Π (for instance, the set of bipartite graphs) the parameter *distance to* Π measures the number of vertices that have to be deleted in the input graph in order to obtain a graph in Π. We denote by P_t the path on t vertices. Then the set of P_t-free graphs consists of all graphs not containing any P_t as induced subgraph. The P_3-free graphs are called cluster graphs. In these graphs, the vertex set of each connected component is a clique. The P_4-free graphs are called cographs. A graph where each connected component is an s-club is called *s-club cluster graph*. Observe that this is equivalent to requiring that every shortest path does not contain a P_{s+2} as subgraph. Hence, P_4-free graphs are 2-club cluster graphs. Deleting all vertices that are contained in an induced P_t is a factor-t approximation for the parameter *distance to P_t-free graphs*. Hence, we may assume that such a vertex deletion set is provided as an additional input for the corresponding algorithms.

A graph is a *co-cluster graph* if its complement graph is a cluster graph. The *clique cover number* is the minimum number of cliques in a graph that are needed to cover all vertices, that is, each vertex is contained in at least one of these cliques. The *domination number* of a graph is the minimum size of a dominating set. This is a set such that each vertex is contained in it or has at least one neighbor in it. An *independent set* is a vertex set inducing a graph without edges. A *vertex cover* is a vertex set whose deletions transforms G into a graph G' without edges, that is, the vertex set of G' is an independent set in G. A set of edge insertions and deletions is a *cluster editing set* if it transforms G into a cluster graph. A set of edges is a *feedback edge set* if its deletion results in an acyclic graph. A graph has *h-index* k, if k is the largest number such that the graph has at least k vertices of degree at least k. The *degeneracy* of a graph is the smallest number d such that each subgraph has at least one vertex of degree at most d. The *bandwidth* of a graph $G = (V, E)$ is the minimum $k \in \mathbb{N}$ such that there is a function $f : V \to \mathbb{N}$ with $|f(v) - f(u)| \leq k$ for all edges $\{u, v\} \in E$.

We now discuss how these structural parameters form a hierarchy as depicted in Figure 1. Intuitively, structural parameters are often related in such a way that parameter α is on all instances smaller than parameter β. For example, distance to P_4-free graphs is always at most as large as distance to P_3-free graphs. Formally, a parameter α is considered to be a stronger parameterization ("smaller") than β if there is a polynomial f (usually linear) such that $\alpha \leq f(\beta)$ for all instances. This stronger/weaker relationship between parameters allows to transform hardness results (from weaker to stronger) and tractability results (from stronger to weaker) between them. Then, "navigating" through

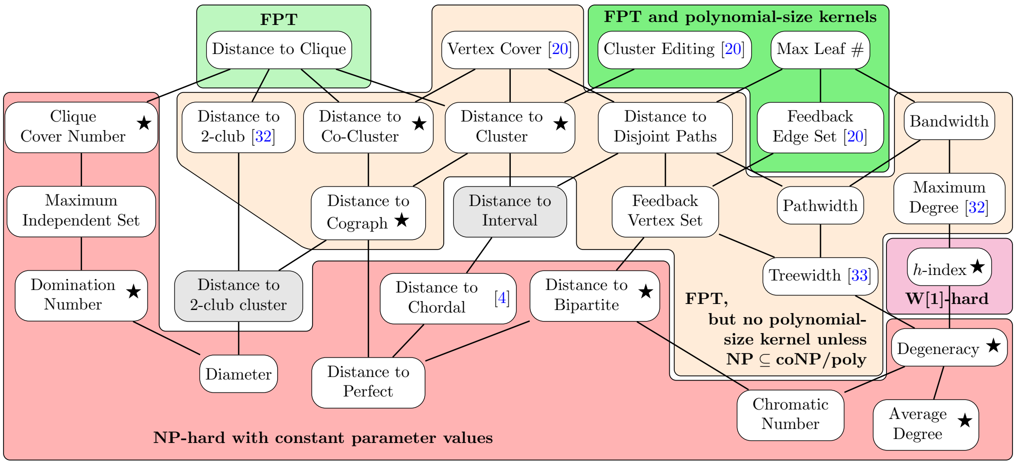

Figure 1: Overview of the relation between structural graph parameters (see Section 1.2) and of our results for 2-Club (marked with a ★). An edge from a parameter α to a parameter β below of α means that β is a stronger parameterization. The box containing the parameter (size of a) *vertex cover* on top, consists of parameters for which 2-Club becomes fixed-parameter tractable but does not admit a polynomial kernel [20]. The box consisting of *cluster editing*, *max leaf #*, and *feedback edge set* contains parameters admitting a fixed-parameter algorithm and a polynomial kernel [20]. The box at the bottom contains parameters where 2-Club remains NP-hard even for constant values. It is open whether 2-Club parameterized by *distance to interval* or *distance to 2-club cluster* is fixed-parameter tractable and whether it admits a polynomial kernel when parameterized by *distance to cliques*.

the corresponding parameter hierarchy allows for a systematic investigation of the parameter's impact on the problem complexity and helps to spot open research questions. For a detailed discussion of the relations depicted in Figure 1 we refer to Sasák [31], and Sorge and Weller [35].

1.3. Our Contribution

We make progress towards a systematic classification of the complexity of 2-Club when parameterized by structural graph parameters. Figure 1 gives an overview of our results and their implications. In Section 2, we consider the graph parameters *clique cover number*, *domination number*, and some related graph parameters. We show that 2-Club is NP-hard even if the *clique cover number* of G is three. In contrast, we show that if the *clique cover number* is two, then 2-Club is polynomial-time solvable. Then, we show that 2-Club is NP-hard even if G has a dominating set of size two. This result is tight in the sense that 2-Club is trivially solvable in case G has a dominating set of size one. In Section 3, we study the parameter *distance to bipartite graphs*. We show that 2-Club is NP-hard even if the input graph can be transformed into a bipartite graph by deleting only one vertex. This is somewhat surprising since 2-Club is polynomial-time solvable on bipartite graphs [33]. Then, in Section 4, we consider the graph parameter *h-index*. The study of this parameter is motivated by the fact that the h-index is usually small in social networks (see Section 4 for a more detailed discussion). On the positive side, we show that 2-Club is polynomial-time solvable for constant h-index. On the negative side, we show that 2-Club parameterized by the *h-index* k of the input graph is W[1]-hard. Hence, a running time of $f(k) \cdot n^{O(1)}$ is probably not achievable (unless W[1] = FPT).

Even worse, we prove that 2-Club becomes NP-hard even for constant degeneracy. Note that degeneracy of a graph is provably at most as large as its h-index.

Finally, in Section 5 we describe a fixed-parameter algorithm for the parameter *distance to cographs* and show that it can be slightly improved for the weaker parameter *distance to cluster graphs*. Interestingly, these are rare examples for structural graph parameters, that are unrelated to treewidth and still admit a fixed-parameter algorithm (see Figure 1). Notably, the fixed-parameter algorithm for treewidth and those for *distance to cograph* both have the same running time characteristic $2^{\Theta(2^k)} \cdot n^{O(1)}$ and this is, so far, also the best for the much "weaker" parameter *vertex cover* [20].

For the sake of completeness, we would like to mention that for the parameters *bandwidth* and *maximum degree*, taking the disjoint union of the input graphs is a composition algorithm that proves the non-existence of polynomial kernels [6], under the standard assumption that NP $\subseteq$ coNP/poly does not hold.

1.4. Preliminaries

We only consider undirected and simple graphs $G = (V, E)$ where $n := |V|$ and $m := |E|$. For a vertex set $S \subseteq V$, let $G[S]$ denote the *subgraph induced by S* and $G - S := G[V \setminus S]$. We use $\mathrm{dist}_G(u, v)$ to denote the *distance between u and v* in G, that is, the length of a shortest path between u and v. For a vertex $v \in V$ and an integer $t \geq 1$, denote by $N^G_t(v) := \{u \in V \setminus \{v\} \mid \mathrm{dist}_G(u, v) \leq t\}$ the set of vertices within distance at most t to v. Moreover, we set $N^G_t[v] := N^G_t(v) \cup \{v\}$, $N^G[v] := N^G_1[v]$ and $N^G(v) := N_1(v)$. If the graph is clear from the context, we omit the superscript G. Two vertices v and w are *twins* if $N(v) \setminus \{w\} = N(w) \setminus \{v\}$ and they are *twins with respect to a vertex set X* with $X \cap \{v, w\} = \emptyset$ if $N(v) \cap X = N(w) \cap X$. The twin relation is an equivalence relation; the corresponding equivalence classes are called *twin classes*. The following simple observation shows that either none or all vertices of a twin class are contained in a maximum-size s-club.

Observation 1. *Let S be an s-club in a graph $G = (V, E)$ and let $u, v \in V$ be twins. If $u \in S$ and $|S| > 1$, then $S \cup \{v\}$ is also an s-club in G.*

We briefly recall the relevant notions from parameterized complexity (see [14, 18, 28]). A problem is *fixed-parameter tractable* (FPT) with respect to a parameter k if there is a computable function f such that any instance (I, k) can be solved in $f(k) \cdot |I|^{O(1)}$ time. A problem is contained in XP if it can be solved in $|I|^{f(k)}$ time for some computable function f. A *kernelization algorithm* reduces any instance (I, k) in polynomial time to an equivalent instance (I', k') with $|I'| \leq g(k)$ and $k' \leq g(k)$ for some computable g. The instance (I', k') is called *kernel* of size g. If g is a polynomial it is called a *polynomial kernel*.

The problem class W[1] is a basic class of presumed parameterized *intractability*. A *parameterized reduction* maps an instance (I, k) in $f(k) \cdot |I|^{O(1)}$ time to an equivalent instance (I', k') with $k' \leq g(k)$ for some functions f and g. A parameterized reduction from a W[1]-hard problem L to a W[1]-hard problem L' proves W[1]-hardness of L' and thus makes fixed-parameter algorithms for L' unlikely.

Remark: Note that one can check in $O(nm)$ time whether a graph is an s-Club by applying a breadth-first search starting from each vertex.

1.5. Guess and Clean

When parameterizing 2-Club with a structural parameter that measures the (vertex-deletion) distance to some graph class Π (e.g. distance to cographs), the input (G, ℓ) is formally extended by a vertex subset X of G such that $G - X \in \Pi$. A common "preprocessing" step in most of our algorithms, which we call *guess and clean*, is to first *guess* the set $X \cap S$ by branching into all $2^{|X|}$ cases to select $X \cap S$ for a fixed maximum-size 2-club S in G. Then, in each branch we *clean* the graph G by first deleting all vertices in $X \setminus S$ and then, recursively, deleting all vertices that have (in the remaining graph) distance more than two to any vertex in $X \cap S$ (they cannot be contained in 2-club containing $X \cap S$). Afterwards, it remains to solve each of the $2^{|X|}$ "cleaned" instances. Such an instance consists of a graph $G' = (V', E')$ and the set $X \cap S$ where $G' - (X \cap S) \in \Pi$ and $\text{dist}_{G'}(u, v) \leq 2$ for all $u \in X$ and $v \in V'$. The task is to find a 2-club of size at least ℓ containing $X \cap S$. For each fixed-parameter algorithm that uses this scheme we thus only describe the algorithm that solves the cleaned instance. Many of these algorithms will again branch into cases to fix some vertices (in addition to $X \cap S$) to be contained in the desired 2-club. For each branch, we again perform "cleaning step", that is, all vertices that do not have distance at most two to all fixed vertices are recursively deleted. If a cleaning step deletes a fixed vertex, then the algorithm is aborted.

2. Clique Cover Number and Domination Number

In this section we provide several hardness proofs for parameters in the "red box" in Figure 1. Specifically, we prove that on graphs of *diameter* at most three, 2-Club is NP-hard even if either the *clique cover number* is three or the *domination number* is two. We first show that these bounds are tight. The size of a maximum independent set is at most the size of a *clique cover*. Moreover, since each maximal independent set is a dominating set, the *domination number* is also at most the size of a *clique cover*.

Lemma 1. *For $s \geq 2$, s-Club can be solved in $O(nm)$ time on graphs where the size of a maximum independent set is at most two.*

Proof. Let $G = (V, E)$ be a graph where the size of maximum independent set is at most two. If a maximum independent set in G has size one or G has diameter at most s, then V is an s-club. Moreover, a connected graph with a maximum independent set size two has diameter at most three, because one could select an independent set of size three on a shortest path of length four. Hence we are left with the case of 2-Club and G having a maximum size-two independent set and diameter three. Thus there are two vertices $v, u \in V$ with $\text{dist}(v, u) = 3$. We next prove that $N_2[v]$ or $N_2[u]$ is a largest 2-club in G and thus can be determined in $O(nm)$ time by a breadth-first search starting from each vertex.

We first show that for each maximum 2-club S it holds that either $v \in S$ or $u \in S$. Assume that $v \notin S$, implying by the maximality of S that $S \cup \{v\}$ is not a 2-club. From this and from $N[v] \cup N[u] = V$ (by the maximality of the independent set $\{u, v\}$) it follows that there is a vertex in $w \in N[u] \cap S$ such that $\text{dist}_{G[S \cup \{v\}]}(v, w) > 2$. Thus w has distance (at least) two to every vertex in $N(v) \cap S$. This implies that all vertices in $N(v) \cap S$ have a neighbor in $N[u] \cap S$, as otherwise they would have distance three to w. Hence, $u \in S$ by the maximality of S.

Next, observe that the set $N[v]$ ($N[u]$) is a clique because two non-adjacent vertices in $N(v)$ ($N(u)$) together with u (with v) would form an independent set of size three, respectively. Since $N[v]$ and $N[u]$ are cliques, it follows that $N_2[v]$ and $N_2[u]$ are 2-clubs and, clearly, $N_2[v]$ is the largest 2-club containing v and, analogously, $N_2[u]$ is the largest 2-club containing u. Thus, $N_2[v]$ or $N_2[u]$ is a largest 2-club in G. □

Every graph with a clique cover number of two has a maximum independent set of size two. Hence, the above result directly implies a polynomial-time algorithm for graphs with a clique cover number of two. Observe that the clique cover number of a graph is exactly the chromatic number of its complement. Hence, 2-Club can be solved in polynomial time on bipartite graphs [33] and on complement graphs of bipartite graphs.

The following theorem shows that the bound on the maximum independent set size in Lemma 1 is tight.

Theorem 1. 2-Club *is NP-hard on graphs with* clique cover number *three and* diameter *three.*

Proof. We describe a reduction from Clique. Let $(G = (V, E), k)$ be a Clique instance and $n := |V|$. If $n = 1$, then we output $(P_{2+k}, 2 + k)$, which is an equivalent instance of 2-Club having the desired properties. Otherwise, we construct a graph $G' = (V', E')$ consisting of three disjoint vertex sets, that is, $V' = V_1 \cup V_2 \cup V_E$. Further, for $i \in \{1, 2\}$, let $V_i = V_i^V \cup V_i^{\text{big}}$, where V_i^V is a copy of V and V_i^{big} is a set of n^5 vertices. Let $u, v \in V$ be two adjacent vertices in G and let $u_1, v_1 \in V_1$, $u_2, v_2 \in V_2$ be the copies of u and v in G'. Then add the vertices e_{uv} and e_{vu} to V_E and add the edges $\{v_1, e_{vu}\}, \{e_{vu}, u_2\}, \{u_1, e_{uv}\}, \{e_{uv}, v_2\}$ to G'. Furthermore, add for each vertex $v \in V$ the vertex set $V_E^v = \{e_v^1, e_v^2, \dots, e_v^{n^3}\}$ to V_E and make v_1 and v_2 adjacent to all these new vertices. Finally, make the following vertex sets cliques: V_1, V_2, V_E, and $V_1^{\text{big}} \cup V_2^{\text{big}}$. Observe that G' has diameter three and that it has a *clique cover number* of three.

We now prove that G has a clique of size $k \Leftrightarrow G'$ has a 2-club of size at least $k' = 2n^5 + kn^3 + 2k + 2\binom{k}{2}$.

"$\Rightarrow$:" Let C be a clique of size k in G. Let $S_c \subseteq V_1^V \cup V_2^V$ contain all the copies of the vertices of C. Furthermore, let $S_E := \{e_{uv} \mid u_1 \in S_c \wedge v_2 \in S_c\}$ and $S_b := \{e_v^i \mid v \in C \wedge 1 \leq i \leq n^3\}$. We now show that $S' := S_c \cup S_E \cup S_b \cup V_1^{\text{big}} \cup V_2^{\text{big}}$ is a 2-club of size k'. First, observe that $|V_1^{\text{big}} \cup V_2^{\text{big}}| = 2n^5$ and $|S_c| = 2k$. Hence, $|S_b| = kn^3$ and $|S_E| = 2\binom{k}{2}$. Thus, S' has the desired size. To verify that S' is indeed a 2-club, for each pair of vertices from the sets S_C, S_E, S_b, V_1^{big}, V_2^{big} the following table shows whether they are adjacent (ad.) or it indicates in which of the sets they have a common neighbor.

	$S_C \cap V_1^V$	$S_C \cap V_2^V$	$S_E \cup S_b$	V_1^{big}	V_2^{big}
$S_C \cap V_1^V$	ad.	$S_E \cup S_b$	ad. or $S_E \cup S_b$	ad.	V_1^{big}
$S_C \cap V_2^V$	-	ad.	ad. or $S_E \cup S_b$	V_2^{big}	ad.
$S_E \cup S_b$	-	-	ad.	V_1^V	V_2^V
V_1^{big}	-	-	-	ad.	ad.
V_2^{big}	-	-	-	-	ad.

"$\Leftarrow$:" Let S' be a maximum-size 2-club in G' of size at least k'. Observe that G' consists of $|V'| = 2n^5 + 2n + 2m + n^4$ vertices. Since $|V| - k' \leq 2n + 2m + n^4 < n^5$ for $n \geq 2$, at least one vertex of V_1^{big} and of V_2^{big} is in S'. Since any two vertices in V_i^{big} are twins, for $i \in \{1, 2\}$, and S' is of maximum-size, by Observation 1 all vertices of $V_1^{\text{big}} \cup V_2^{\text{big}}$ are contained in S'. Analogously, it follows that at least k sets $V_E^{v^1}, V_E^{v^2}, V_E^{v^3}, \ldots, V_E^{v^k}$ are completely contained in S'. Since S' is a 2-club, the distance from vertices in V_i^{big} to vertices in $V_E^{v^j}$ is at most two. Hence, for each set $V_E^{v^j}$ in S' the two neighbors v_1^j and v_2^j of vertices in $V_E^{v^j}$ are also contained in S'. Since the distance of v_1^i and v_2^j for $v_1^i, v_2^j \in S'$ is also at most two, the vertices $e_{v^i v^j}$ and $e_{v^j v^i}$ are part of S' as well. Consequently, v^i and v^j are adjacent in G. Therefore, the vertices $v^1, \ldots, v^k$ form a size-k clique in G. ☐

Since a maximum independent set is also a dominating set, Theorem 1 implies that 2-Club is NP-hard on graphs with *domination number* three and *diameter* three. In contrast, for *domination number* one 2-Club is trivial. The following theorem shows that this cannot be extended.

Theorem 2. 2-Club *is NP-hard even on graphs with* domination number *two and* diameter *three.*

Proof. We present a reduction from Clique. Let $(G = (V, E), k)$ be a Clique instance and assume that G does not contain isolated vertices. We construct the graph G' as follows. First copy all vertices of V into G'. In G' the vertex set V will form an independent set. Now, for each edge $\{u, v\} \in E$ add an *edge vertex* $e_{\{u,v\}}$ to G' and make $e_{\{u,v\}}$ adjacent to u and v. Let V_E denote the set of edge vertices. Next, add a vertex set C of size $n+2$ to G' and make $C \cup V_E$ a clique. Finally, add a new vertex v^* to G' and make v^* adjacent to all vertices in V. Observe that v^* plus an arbitrary vertex from $V_E \cup C$ are a dominating set of G' and that G' has diameter three. We complete the proof by showing that G has a clique of size $k \Leftrightarrow G'$ has a 2-club of size at least $|C| + |V_E| + k$.

"$\Rightarrow$:" Let K be a size-k clique in G. Then, $S := K \cup C \cup V_E$ is a size-$|C| + |V_E| + k$ 2-club in G: First, each vertex in $C \cup V_E$ has distance two to all other vertices of S. Second, each pair of vertices $u, v \in K$ is adjacent in G and thus they have the common neighbor $e_{\{u,v\}}$ in V_E.

"$\Leftarrow$:" Let S be a 2-club of size $|C| + |V_E| + k$ in G'. Since $|C| > |V \cup \{v^*\}|$, it follows that there is at least one vertex $c \in S \cap C$. Since c and v^* have distance three, it follows that $v^* \notin S$. Now since S is a 2-club, each pair of vertices $u, v \in S \cap V$ has at least one common neighbor in S. Hence, V_E contains the edge vertex $e_{\{u,v\}}$. Consequently, $S \cap V$ is a size-k clique in G. ☐

3. Distance to Bipartite Graphs

A 2-club in a bipartite graph is a biclique (a complete bipartite graph). Finding a biclique with a maximum number of vertices can be done via matching in bipartite graphs, hence 2-Club is polynomial-time solvable on bipartite graphs [33]. However, we show that 2-Club is already NP-hard on graphs that become bipartite by deleting only one vertex.

Theorem 3. 2-Club *is NP-hard even on graphs with distance one to bipartite graphs.*

Proof. We reduce from the NP-hard Maximum 2-SAT problem.

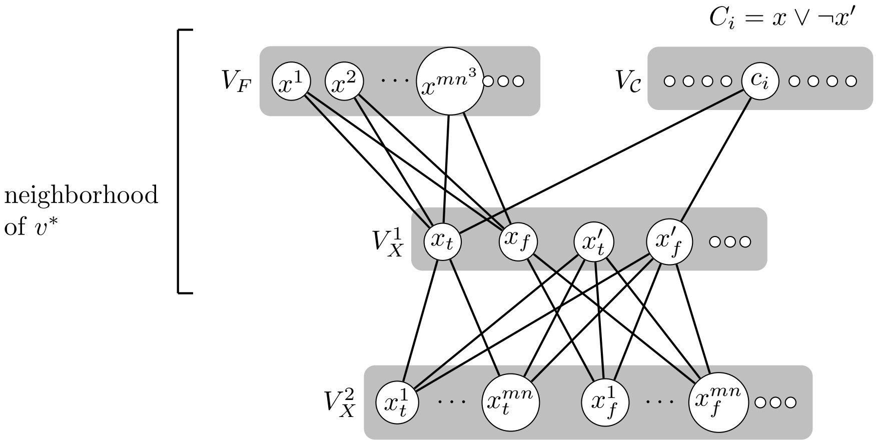

Figure 2: Schematic illustration of the reduction provided in the proof of Theorem 3. The main idea behind the construction is as follows. The size of the desired 2-club forces to contain the majority of the vertices in V_F and V_X^2. This has two consequences: For each $x \in X$ either x_t or x_f must be contained in a 2-club; otherwise two vertices from V_F and V_X^2 have distance three. Hence, the vertices from V_X^1 in the 2-club represent a truth assignment. Furthermore, since V_X^2 has only neighbors in V_X^1 and since the subgraph induced by $V_X^1 \cup V_X^2$ is bipartite, the subsets of V_X^1 and V_X^2 in a 2-club induce a complete bipartite graph. Finally, to fulfill the bound on the 2-club size, at least k vertices from $V_{\mathcal{C}}$ are in the 2-club; these vertices can only be added if the corresponding clauses are satisfied by the represented truth assignment.

> Maximum 2-SAT
> **Input:** A set $\mathcal{C} = \{C_1, \ldots, C_m\}$ of clauses over a variable set $X = \{x_1, \ldots, x_n\}$ where each clause C_i contains two literals.
> **Question:** Is there an assignment β for X that satisfies at least k clauses of $\mathcal{C}$?

Given an instance of Maximum 2-SAT, we construct an equivalent instance of 2-Club as follows. If $n \leq 4$ or $m \leq 2$, then the Maximum 2-SAT instance has constant size. We can thus solve it in constant time and output an equivalent instance of 2-Club having constant size and the desired properties. Otherwise, we construct an undirected graph $G = (V, E)$, where the vertex set V consists of the four disjoint vertex sets $V_{\mathcal{C}}$, V_F, V_X^1, V_X^2, and one additional vertex v^*. The construction of the four subsets of V is as follows (see Figure 2 for an illustration and a description of the main idea).

The vertex set $V_{\mathcal{C}}$ contains one vertex c_i for each clause $C_i \in \mathcal{C}$. The vertex set V_F contains for each variable $x \in X$ exactly mn^3 vertices $x^1, \ldots, x^{mn^3}$. The vertex set V_X^1 contains for each variable $x \in X$ two vertices: x_t which corresponds to assigning true to x and x_f which corresponds to assigning false to x. The vertex set V_X^2 is constructed similarly, but for every variable $x \in X$ it contains $2 \cdot mn$ vertices: the vertices $x_t^1, \ldots x_t^{mn}$ which correspond to assigning true to x, and the vertices $x_f^1, \ldots x_f^{mn}$ which correspond to assigning false to x.

Next, we describe the construction of the edge set E. The vertex v^* is made adjacent to all vertices in $V_{\mathcal{C}} \cup V_F \cup V_X^1$. Each vertex $c_i \in V_{\mathcal{C}}$ is made adjacent to the two vertices in V_X^1 that correspond to the two literals in C_i. Each vertex $x^i \in V_F$ is made adjacent to x_t and x_f, that is, the two vertices of V_X^1 that correspond to the two truth assignments for the variable x. Finally, each vertex $x_t^i \in V_X^2$ is made adjacent to all vertices of V_X^1 except to the vertex x_f. Similarly, each $x_f^i \in V_X^2$ is made adjacent to all vertices of V_X^1 except to x_t. This completes the construction of G which can clearly be performed in polynomial time. Observe that the removal of v^* makes G bipartite: each of the four vertex sets is an independent set and the vertices of $V_{\mathcal{C}}$, V_F, and V_X^2 are only adjacent to vertices of V_X^1. It remains to prove that $(\mathcal{C}, k)$ is a yes-instance of Maximum 2-Sat $\Leftrightarrow$ G has a 2-club of size at least

$$\underbrace{mn^4}_{|V_F|} + \underbrace{mn^2}_{|V_X^2|/2} + \underbrace{n}_{|V_X^1|/2} + k + 1.$$

"$\Rightarrow$": Let β be an assignment for X that satisfies k clauses $C_1, \ldots, C_k$ of $\mathcal{C}$. Consider the vertex set S that consists of V_F, v^*, the vertex set $\{c_1, \ldots, c_k\} \subseteq V_{\mathcal{C}}$ that corresponds to the k satisfied clauses, and for each $x \in X$ of the vertex set $\{x_t, x_t^1, \ldots, x_t^{mn}\} \subseteq V_X^1 \cup V_X^2$ if $\beta(x) = \text{true}$ and the vertex set $\{x_f, x_f^1, \ldots, x_f^{mn}\} \in V_X^1 \cup V_X^2$ if $\beta(x) = \text{false}$. Clearly, $|S| = mn^4 + mn^2 + n + k + 1$. In the following, we show that S is a 2-club. Herein, let $S_X^1 := V_X^1 \cap S$, $S_X^2 := V_X^2 \cap S$, and $S_{\mathcal{C}} := V_{\mathcal{C}} \cap S$.

First, v^* is adjacent to all vertices in $S_{\mathcal{C}} \cup V_F \cup S_X^1$. Hence, all vertices of $S \setminus S_X^2$ are within distance two in $G[S]$. By construction, the vertex sets S_X^1 and S_X^2 form a complete bipartite graph in G: A vertex $x_t^i \in S_X^2$ is adjacent to all vertices in V_X^1 except x_f which is not contained in S_X^1. The same argument applies to some $x_f^i \in S_X^2$. Hence, the vertices of S_X^2 are neighbors of all vertices in S_X^1. This also implies that the vertices of S_X^2 are in $G[S]$ within distance two from v^* and from every vertex in V_F since each vertex of $V_F \cup \{v^*\}$ has at least one neighbor in S_X^1. Finally, since the k vertices in $S_{\mathcal{C}}$ correspond to clauses that are satisfied by the truth assignment β, each of these vertices has at least one neighbor in S_X^1. Hence, every vertex in S_X^2 has in $G[S]$ distance at most two to every vertex in $S_{\mathcal{C}}$.

"$\Leftarrow$": Let S be a 2-club of size at least $mn^4 + mn^2 + n + k + 1$, and let $S_X^1 := V_X^1 \cap S$, $S_X^2 := V_X^2 \cap S$, $S_F := V_F \cap S$ and $S_{\mathcal{C}} := V_{\mathcal{C}} \cap S$. Since $|V_{\mathcal{C}}| + |V_X^1| + |V_X^2| + 1 \leq m + 2n + 2mn^2 + 1 < mn^3$ for $n \geq 5$ and $m \geq 3$, S contains more than $mn^4 - mn^3$ vertices from V_F. Consequently, for each $x \in X$ there is an index $1 \leq i \leq mn^3$ such that $x^i \in S_F$. Similarly, since $|V_{\mathcal{C}}| + |V_X^1| + |V_F| + 1 \leq m + 2n + mn^4 + 1 < mn^4 + mn^2$ for $n \geq 5$ and $m \geq 3$, we have $S_X^2 \neq \emptyset$.

We next show that for each $x \in X$ it holds that either x_t or x_f is contained in S_X^1. Since S is a 2-club, every vertex pair $x^i \in S_F$ and $u \in S_X^2$ has at least one common neighbor in S. By construction, this common neighbor is a vertex of S_X^1 and thus it is either x_t or x_f. Moreover, by the observation above for each $x \in X$ at least one x^i is contained in S_F. Thus, for each $x \in X$ at least one of x_t and x_f is contained in S_X^1.

Now observe that, $G[S_X^1 \cup S_X^2]$ is a complete bipartite graph, since S_X^1 and S_X^2 are independent sets and S_X^2 has only neighbors in S_X^1. This implies that if for some $x \in X$ there exist indices $1 \leq i, j \leq mn$ such that x_t^i and x_f^j are in S_X^2, then x_t and x_f are *not* in S_X^1. This contradicts the above observation that at least one of x_t and x_f is in S_X^1. Moreover, since $|V_{\mathcal{C}}| + |V_X^1| + 1 \leq m + 2n + 1 < mn$ for $n \geq 5$ and $m \geq 3$ and $|S \setminus V_F| > mn^2$, we have $|S_X^2| > mn^2 - mn$. It follows that for each $x \in X$ there is an index $1 \leq i \leq mn$ such that either $x_t^i \in S_X^2$ or $x_f^i \in S_X^2$. Finally, this implies that either x_t or x_f is not contained in S_X^1.

Summarizing, S has at most mn^4 vertices from V_F, at most mn^2 vertices belonging to S_X^2, exactly n vertices belonging to S_X^1, and thus there are $k+1$ vertices in $S_\mathcal{C} \cup \{v^*\}$. Since S is a 2-club that has non-empty S_X^2, every one of the at least k vertices from $S_\mathcal{C}$ has at least one neighbor in S_X^1. Because for each $x \in X$ either x_f or x_t is in S_X^1, the n vertices from S_X^1 correspond to an assignment β of X. By the above observation, this assignment satisfies at least k clauses of $\mathcal{C}$. $\square$

4. Average Degree, Degeneracy, and h-Index

2-Club is fixed-parameter tractable for the parameter *maximum degree* which can easily be shown for the algorithm of Schäfer et al. [32]. It has been observed that in large-scale biological [23] and social networks [5] the degree distribution often follows a power law, implying that there are some high-degree vertices while most vertices have low degree. This suggests considering stronger parameters such as *h-index*, *degeneracy*, and *average degree* [15]. For any graph it holds that avg. degree $\leq 2 \cdot$ degeneracy $\leq 2 \cdot h$-index, see also Figure 1 for other relationships. Furthermore, analyzing the coauthor network derived from the DBLP dataset[3] with more than 715,000 vertices, maximum degree 804, h-index 208, degeneracy 113, and average degree 7 shows that also in real-world social networks these parameters are considerably smaller than the maximum degree (see [20] for an analysis of these parameters on a broader dataset).

Unsurprisingly, 2-Club is NP-hard on graphs of constant *average degree*.

Proposition 1. *For any constant $\alpha > 2$, 2-Club is NP-hard on connected graphs with* average degree *at most α.*

Proof. Let (G, ℓ) be an instance of 2-Club where Δ is the maximum degree of G. We can assume that $\ell > \Delta + 2 \geq 3$ since, as shown for instance in the proof of Theorem 1, 2-Club remains NP-hard in this case. We add a path P to G and an edge from an endpoint of P to an arbitrary vertex $v \in V$, resulting in the graph G'. Thereby, putting at least $\lceil \frac{2m}{\alpha-2} - n \rceil$ vertices in P ensures that G' has average degree at most α.

We next prove the (G, ℓ) is a yes-instance if and only if (G', ℓ) is a yes-instance. Clearly, any 2-club in G is also a 2-club in G'. Reversely, let S' be a 2-club in G' of size at least ℓ. Since the degree of v in G' is at most $\Delta + 1$, S' contains at least one vertex, say u, that is not a neighbor of v. In case of $u \in P$, by construction it would follow that $S' \subseteq P \cup \{v\}$ and thus S' is a path with at most three vertices. This implies that $S' \cap P = \emptyset$ and thus S' is also a 2-club in G. $\square$

We remark that the bound provided in Proposition 1 is tight: Consider a connected graph G with average degree at most two, that is, $\frac{1}{n} \sum_{v \in V} \deg(v) \leq 2$. Because $\sum_{v \in V} \deg(v) = 2m$, it follows that $n \geq m$ and, thus, the feedback edge set of G contains at most one edge. As 2-Club is fixed-parameter tractable when parameterized by the (size of a) *feedback edge set* [20], it follows that 2-Club can be solved in polynomial time on connected graphs with *average degree* at most two.

Proposition 1 suggests considering "weaker" parameters such as *degeneracy* or *h-index* of G (see Figure 1). Recall that having h-index k means that there are at most k vertices

[3]The dataset and a corresponding documentation are available online (http://dblp.uni-trier.de/xml/). Accessed Feb. 2012

with degree greater than k. Since social networks have small h-index [20], fixed-parameter tractability when parameterized by the h-index would be desirable. Unfortunately, we show that 2-Club is W[1]-hard when parameterized by the h-index and NP-hard with constant *degeneracy*. Following this result, we show that there is "at least" an XP-algorithm implying that 2-Club is polynomial-time solvable for constant h-index.

We reduce from the W[1]-hard Multicolored Clique problem [16]. Therein, given a coloring $c : V \to \{1, \dots, k\}$ of the vertices V of a graph, a subset of $C \subseteq V$ is called a *multicolored clique* if C is a clique and $c(v) \neq c(v')$ for all $\{v, v'\} \subseteq C$ with $v \neq v'$.

Multicolored Clique
Input: An undirected graph $G = (V, E)$, $k \in \mathbb{N}$, and a (vertex) coloring $c : V \to \{1, \dots, k\}$.
Question: Is there a multicolored clique of size k in G?

Lemma 2. *There are two polynomial-time computable reductions that compute for any instance (G, c, k) of* Multicolored Clique *an equivalent* 2-Club*-instance (G', ℓ) such that G' has* diameter *three and, additionally, in reduction i) G' has* h-index *at most $k + 7$ and in reduction ii) G' has* degeneracy *six.*

Proof. The only difference between both reductions is the construction of a so-called *coloring gadget*. We first describe the common part.

Let (G, c, k) with $G = (V, E)$, $n := |V|$ and $c : V \to \{1, \dots, k\}$ be an instance of Multicolored Clique. We may assume without loss of generality that there is no edge between equally-colored vertices in G (otherwise they can be safely removed). If $n \leq 4$ then solve the instance (in constant time) and output a constant size equivalent instance of 2-Club having the desired properties. Otherwise, we construct a graph G' and choose $\ell \in \mathbb{N}$ such that (G', ℓ) is a yes-instance for 2-Club if and only if (G, c, k) is a yes-instance for Multicolored Clique. Eventually, G' consists of a *vertex gadget* for each vertex in V, an *edge vertex* for each edge in E, an *anchor gadget*, and a coloring gadget. We first provide the description of the vertex gadgets and the edge vertices.

Vertex Gadget & Edge Vertex: For each vertex $v \in V$ create a *vertex gadget* by adding the *α-vertices* $\{\alpha_1^v, \dots, \alpha_n^v\}$, the *$\beta$-vertices* $\{\beta_1^v, \dots, \beta_{n+1}^v\}$, and the *$\gamma$-vertices* $\{\gamma_1^v, \dots, \gamma_n^v\}$, and $\{\omega_\alpha^v, \omega_\gamma^v\}$. Add edges such that $(\alpha_1^v, \beta_1^v, \gamma_1^v, \alpha_2^v, \beta_2^v, \gamma_2^v, \dots, \alpha_n^v, \beta_n^v, \gamma_n^v, \omega_\alpha^v, \beta_{n+1}^v, \omega_\gamma^v, \alpha_1^v)$ induces a cycle. Add the three vertices $U = \{u_\alpha, u_\beta, u_\gamma\}$ (set U exists only once and it belongs to the anchor gadget) and add edges from all α- (β-, γ-)vertices to u_α (u_β, u_γ), respectively. Add the edges $\{\omega_\alpha^v, u_\alpha\}$ and $\{\omega_\gamma^v, u_\gamma\}$ for the *ω-vertices* $\omega_\alpha^v, \omega_\gamma^v$.

Furthermore, for a fixed ordering $V = \{v_1, \dots, v_n\}$ add for each edge $\{v_i, v_j\} \in E$ an *edge vertex* $e_{i,j}$ that is adjacent to each of $\{\alpha_j^{v_i}, \beta_j^{v_i}, \gamma_i^{v_j}\}$ (the α- and the γ-vertex neighbor are in different vertex gadgets). The following property holds:

1. Each α-, β-, γ-, ω-vertex is adjacent to exactly one vertex in U and has only one common neighbor with each of the other two vertices in U. Furthermore, for each edge vertex $e_{i,j}$ it holds that its neighbor $\alpha_j^{v_i}$ ($\beta_j^{v_i}$, $\gamma_i^{v_j}$) is the only common neighbor with u_α (u_β, u_γ), respectively.

Basic Idea: The idea of the construction is that U will be forced to be contained in any 2-club S of size at least ℓ (by the anchor gadget). Hence by Property 1 it follows

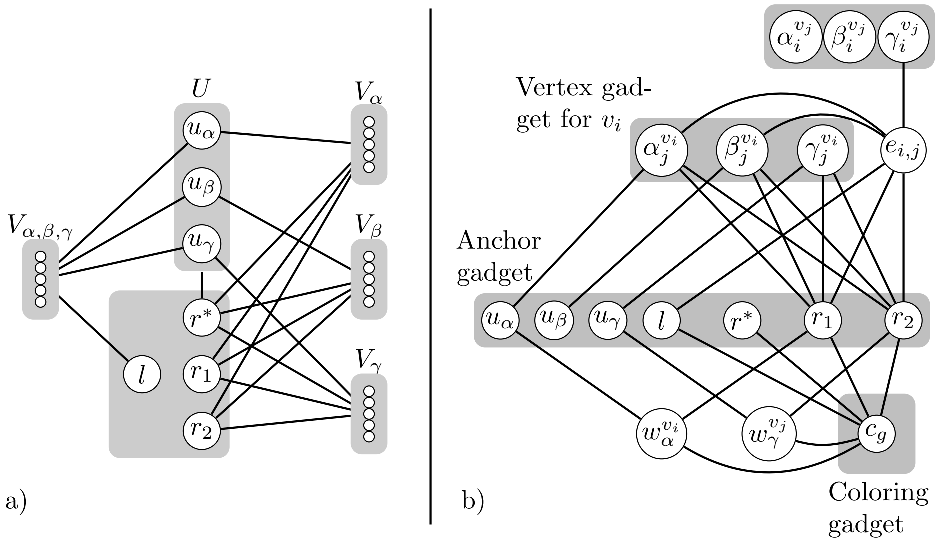

Figure 3: Schematic illustration of the reduction in the proof of Lemma 2. Part a) shows the anchor gadget and Part b) shows the rest of the construction. Part a): The vertices $\{u_\alpha, u_\beta, u_\gamma, l, r^*, r_1, r_2\}$ are the only vertices that have neighbors outside the anchor gadget and $\{l, r^*, r_1, r_2\}$ forms a clique. All edges are drawn from a vertex to a gray-colored box, meaning that, this vertex is adjacent to all vertices within the box. All vertices from one of the sets $\{V_{\alpha,\beta,\gamma}, V_\alpha, V_\beta, V_\gamma\}$ are twins and u_α (u_β, u_γ) is the only common neighbor between $V_{\alpha,\beta,\gamma}$ and V_α (V_β, V_γ), respectively. Together with the choice of ℓ this forces the vertices in $U = \{u_\alpha, u_\beta, u_\gamma\}$ to be contained in all 2-clubs of size at least ℓ. Part b): All edges between the anchor gadget vertices $\{u_\alpha, u_\beta, u_\gamma, l, r^*, r_1, r_2\}$ and all other vertices are depicted. Coloring gadget i) is illustrated by vertex c_g, assuming that v_i is of color g and v_j is of different color.

that if an α-vertex is contained in S, then the unique β- and γ-vertex in its neighborhood has to be contained in S as well (here, ω-vertices behave like "normal" α/γ-vertices). Since this argument symmetrically holds for β- and γ-vertices, it follows that either all or none of the vertices from a vertex gadget are contained in S. Analogously, each edge vertex $e_{i,j} \in S$ needs to have a common neighbor with each vertex of U. Thus, by Property 1 from $e_{i,j} \in S$ it follows that the vertex gadgets corresponding to v_i and v_j are completely contained in S. By connecting the ω-vertices appropriately we will ensure that S cannot contain two vertex gadgets that correspond to equally-colored vertices in G. Furthermore, we choose the value of ℓ such that S contains exactly $\binom{k}{2}$ edge vertices and thus by Property 1 also the k corresponding vertex gadgets, implying that the corresponding vertices in G form a multicolored clique. To complete the construction we next add the *anchor gadget* and the *coloring gadget*.

Anchor Gadget: The construction is as follows (see Figure 3 a)): Add the vertices $\{l, r^*, r_1, r_2\}$ which are together with U the only vertices having neighbors outside the anchor gadget. Add four vertex sets $V_\alpha, V_\beta, V_\gamma, V_{\alpha,\beta,\gamma}$ each of size n^3 and add edges from each vertex in $V_{\alpha,\beta,\gamma}$ to each in $U \cup \{l\}$. Additionally, add edges from each vertex in V_α (V_β, V_γ) to u_α (u_β, u_γ) and add an edge from each of $V_\alpha \cup V_\beta \cup V_\gamma$ to each of $\{r^*, r_1, r_2\}$. Finally, add edges such that $\{l, r^*, r_1, r_2\}$ is a clique and an edge from r^* to each vertex in U.

Denoting by V_A the set of all anchor gadget vertices, the following property follows directly from the anchor gadget construction.

2. Any vertex that is adjacent to one of $\{l, u_\alpha, u_\beta, u_\gamma\}$ and to one of $\{r^*, r_1, r_2\}$ has distance at most two to all vertices in $V_A \setminus U$.

Connecting Vertex and Anchor Gadget: Recall that so far only U has neighbors outside the anchor gadget. We describe via properties how to connect the anchor gadget to the vertex gadgets and the edge vertices.

3. ω_α^v is adjacent to r_1 and ω_γ^v is adjacent to r_2 for all $v \in V$.

4. All α-, β-, and γ-vertices are adjacent to r_1 and r_2. All edge vertices are adjacent to each of $\{l, r_1, r_2\}$.

Coloring Gadget: We next construct the so-called coloring gadget that guarantees that only those vertex pairs $\{\omega_\alpha^v, \omega_\gamma^{v'}\}$ have a common neighbor (and thus can be contained in any 2-club) for which $c(v) \neq c(v')$. We will give two different constructions of the coloring gadget where the first guarantees an h-index of at most $k+7$ and the second guarantees degeneracy six. Denoting coloring gadget vertices by V_C we will prove that both constructions fulfill the following properties:

5. Each vertex in V_C is adjacent to each of $\{l, r^*, r_1, r_2\}$.

6. Any pair $\{\omega_\alpha^v, \omega_\gamma^{v'}\}$, $v \neq v'$, has a common neighbor in V_C if and only if $c(v) \neq c(v')$.

Coloring gadget i): For each color $i \in \{1, \ldots, k\}$ add a vertex c_i and let $V_C = \{c_1, \ldots, c_k\}$ the vertex set containing these vertices. Add an edge between a vertex ω_α^v and c_i if $c(v) = i$ and an edge from ω_γ^v to c_i if $c(v) \neq i$ (Property 6). Finally, add edges such that each vertex in V_C is adjacent to each vertex in $\{l, r^*, r_1, r_2\}$ (Property 5).

Note that the h-index of G' is at most $|V_C| + |U| + |\{l, r^*, r_1, r_2\}| = k + 7$, as the vertices in $V_C \cup U \cup \{l, r^*, r_1, r_2\}$ are the only ones that might have degree at least $k+7$.

Coloring gadget ii): For each pair $\{\omega_\alpha^v, \omega_\gamma^{v'}\}$ with $c(v) \neq c(v')$ add a vertex $c_{v,v'}$ that is adjacent to each of $\{\omega_\alpha^v, \omega_\gamma^{v'}\}$ (Property 6). Finally, denoting all these new vertices by V_C we add an edge from each vertex in V_C to each vertex in $\{l, r^*, r_1, r_2\}$ (Property 5).

We next prove that G' has degeneracy six by giving an elimination order, that is, an order of how to delete vertices of degree at most six that results in an empty graph: In the anchor gadget each of the vertices in $V_\alpha, V_\beta, V_\gamma, V_{\alpha,\beta,\gamma}$ has maximum degree four and hence they can be deleted. Then, delete all α-, β-, γ-, ω-vertices and edge vertices as each of them has degree six. In the remaining graph all vertices in V_C have neighborhood $\{l, r^*, r_1, r_2\}$ and thus can be deleted. Then, delete the degree-one vertices $\{u_\alpha, u_\beta\}$ and the clique $\{l, r^*, r_1, r_2\}$.

Having described the construction of G' we finally set

$$\ell = \underbrace{k(3n+3)}_{k \text{ vertex gadgets}} + \underbrace{4n^3+7}_{\text{anchor gadget}} + \underbrace{\binom{k}{2}}_{\binom{k}{2} \text{ edge vertices}} + \underbrace{|V_C|}_{\text{coloring gadget}} . \tag{1}$$

Correctness: It remains to prove that (G, c, k) is a yes-instance of Multicolored Clique $\Leftrightarrow$ (G', ℓ) is a yes-instance of 2-Club.

"⇒" Let C be a multicolored clique in G of size k. We construct a set $S \subseteq V'$ of size ℓ and prove that it is a 2-club in G'. The set S contains each vertex gadget that corresponds to some vertex in C, the coloring gadget, the anchor gadget, and any edge vertex $e_{i,j}$ with $v_i, v_j \in C$. See Equation (1) to verify that $|S| = \ell$.

It remains to prove that S is a 2-club and thus consider the distances in $G'[S]$. We first prove that all vertices in V_C have distance at most two to all other vertices in S. This follows from the fact that each vertex gadget vertex, each edge vertex, and each vertex in V_A has a neighbor in $\{l, r^*, r_1, r_2\}$ and thus by Property 6 has distance at most two to all vertices in V_C.

We next prove that all vertices in V_A have distance at most two to all vertices in S. Since all vertex gadget vertices are adjacent to one vertex in U and to one of $\{r_1, r_2\}$ it follows by Property 2 that they have distance at most two to all vertices in $S \cap (V_A \setminus U)$. Similarly, as all edge vertices are adjacent to $\{l, r_1, r_2\}$ (Property 4), again by Property 4 it follows that edge vertices have distance at most two to all vertices in $S \cap (V_A \setminus U)$. Furthermore, since S contains each vertex gadget entirely and thus also the set $\{\alpha_j^{v_i}, \beta_j^{v_i}, \gamma_i^{v_j}\}$ for each edge vertex $e_{i,j}$, it follows that all vertex gadget vertices and all edge vertices have distance at most two to all vertices in V_A. It remains to show that each vertex pair from V_A has distance at most two: Each vertex in V_A has a neighbor in the clique $\{l, r^*, r_1, r_2\}$ and thus they all have distance at most two to all these clique vertices. Since the vertices in each of the sets $V_{\alpha,\beta,\gamma}, V_\alpha, V_\beta, V_\gamma$ are twins it follows that any vertex pair from one of these sets has distance at most two. Moreover, vertices $\{u_\alpha, u_\beta, u_\gamma, r^*\}$ ensure that, in fact, each vertex pair from $V_{\alpha,\beta,\gamma} \cup V_\alpha \cup V_\beta \cup V_\gamma$ has distance at most two. Additionally, each vertex in $V_{\alpha,\beta,\gamma} \cup V_\alpha \cup V_\beta \cup V_\gamma$ is a neighbor of all of $\{u_\alpha, u_\beta, u_\gamma\}$ or is adjacent to r^* which is a neighbor of $\{u_\alpha, u_\beta, u_\gamma\}$, proving that they all have distance at most two to all vertices in V_A. Finally, observe that u_α, u_β, and u_γ have the common neighbor r^*.

Finally, we consider the vertex gadget vertices and edge vertices. By Property 4 all α-, β-, γ-vertices and edge vertices have two common neighbors $\{r_1, r_2\}$. Additionally, ω_α^v is adjacent to r_1 and ω_γ^v is adjacent to r_2, implying that each pair, except those consisting of two ω-vertices, have the common neighbor r_1 or r_2. Finally, since C is a clique and S contains the corresponding vertex gadgets and edge vertices, it follows that each pair of ω-vertices in S either have a common neighbor because they are in the same vertex gadget or they have common neighbor in V_C (Property 6). This proves that S is a 2-club of size at least ℓ.

"⇐": Let S be a size at least ℓ 2-club in G'. We first prove that $V_A \subseteq S$. Towards this, we first prove that S contains from each of the vertex sets $V_\alpha, V_\beta, V_\gamma, V_{\alpha,\beta,\gamma}$ at least one vertex. The maximum number of vertices in G' that is not contained in S is

$$\begin{aligned} |V'| - \ell = {} & \underbrace{n(3n+3)}_{n \text{ vertex gadgets}} + \underbrace{4n^3+7}_{\text{anchor gadget}} + \underbrace{|E|}_{\text{edge vertices}} + \underbrace{|V_C|}_{\text{coloring gadget}} - \ell \\ \overset{(1)}{=} {} & (n-k)(3n+3) + |E| - \binom{k}{2} < n^3 \text{ (since } n > 4). \end{aligned}$$

Since each of $V_\alpha, V_\beta, V_\gamma, V_{\alpha,\beta,\gamma}$ has size n^3 it follows that at least one vertex from each of them is contained in S. From this it directly follows that $U \subseteq S$, since u_α (u_β, u_γ) is the only common neighbor of a vertex in $V_{\alpha,\beta,\gamma}$ and one in V_α (V_β, V_γ), respectively.

From Equation 1 it follows that at least $k(3n+3)+\binom{k}{2}$ vertices from the union of the vertex gadget vertices and edge vertices are contained in S. Since by Property 1 either all or none of the vertices of a vertex gadget are contained in S, from Property 6 it follows that S does not contain two vertex gadget vertices corresponding to two different but equally-colored vertices v and v' in G (the vertices $\{\omega_\alpha^v$ and $\omega_\gamma^{v'}\}$ would have no common neighbor). Hence, S contains vertices from at most k different vertex gadget vertices. However, since by Property 1 for each edge vertex $e_{i,j} \in S$ it holds that the vertex gadget corresponding to v_i and also the one for v_j is contained in S, it follows that S contains exactly k vertex gadgets and, correspondingly, $\binom{k}{2}$ edge vertices. Finally, from Property 1 it follows that the edges corresponding to the $\binom{k}{2}$ edge vertices in S must have their endpoints in the k vertices corresponding to the k vertex gadgets in S. Hence, the vertices in G corresponding to the vertex gadgets in S form a clique of size k in G. □

Lemma 2 has several consequences.

Corollary 1. 2-Club *is NP-hard on graphs with* degeneracy *six.*

Corollary 2. 2-Club *parameterized by* h-index *is W[1]-hard.*

The reduction in Lemma 2 is from Multicolored Clique and the new parameter is linearly bounded in the old one. Combining this with the results of Chen et al. [13] leads to the following stronger running time bound.

Corollary 3. 2-Club *cannot be solved in* $n^{o(k)}$*-time on graphs with* h-index k *unless the exponential time hypothesis fails.*

We next prove that there is an XP-algorithm for the parameter h-index. Therein, we mainly exploit the fact that if a graph G has h-index k, then there is a set X of at most k vertices such that $G - X$ has maximum degree at most k. Since 2-Club is fixed-parameter tractable when parameterized by the maximum degree, one can find largest 2-clubs in the connected components of $G - X$ and when combining them to a larger 2-club S one needs to ensure that they share common neighbors in $X \cap S$.

Theorem 4. 2-Club *can be solved in* $O(2^{k^4+k} \cdot n^{2^k} \cdot n^2 m)$ *time where k is the* h-index *of the input graph.*

Proof. We give an algorithm that, given a guessed and cleaned input (G, ℓ, X), finds in $O(2^{k^4} \cdot n^{2^k} \cdot n^2 m)$ time a maximum 2-club in G that contains X where k denotes the h-index of G. Therein, X is the set of vertices in G with degree greater than k. By definition of the h-index, $|X| \le k$. For the proof of correctness fix any maximum 2-club S in G with $X \subseteq S$. Throughout the algorithm via branching we will guess some vertices contained in S and we will collect them in the set P. We initialize P with X.

Consider the at most 2^k twin classes of the vertices in $V \setminus X$ with respect to X. The algorithm first branches into the $O(n^{2^k})$ cases to guess for each twin class T any vertex from $T \cap S$, called the *center* of T. Clearly, if $T \cap S = \emptyset$, then there is no center and we delete all vertices in T. Add all the centers to P and clean the graph.

Two twin classes T and T' are in *conflict* if $N^G(T) \cap N^G(T') \cap X = \emptyset$. Now, the crucial observation is that, if T and T' are in conflict, then all vertices in $(T \cup T') \cap S$ are contained in the same connected component of $G[S \setminus X]$, since otherwise they would not have pairwise distance at most two. However, this implies that all vertices in $T \cap S$ have pairwise distance at most four in $G[S \setminus X]$: In $G[S]$ there is a length at most two path for each pair $v \in T$ and $w \in T'$ not containing any vertex from X (by definition of T and T' being in conflict). Thus there is a length at most four path for any pair from T (T') in $G[S \setminus X]$.

Hence, for each twin class T with center c that is in conflict to any other twin class it holds that $T \cap S \subseteq N_4^{G-X}[c]$ and since $G - X$ has maximum degree at most k, one can guess $N_4^S[c] := N_4^{G-X}[c] \cap S$ by branching into at most 2^{k^4} cases. Delete all vertices in T guessed to be not contained in $N_4^S[c]$, add $N_4^S[c]$ to P, and clean the graph. Note that the remaining graph is a 2-club, since P contains X and the intersection of S with each twin class that is in conflict to any other twin class. By definition of twin classes that are in conflict, it holds that all other twin classes share a common neighbor in X. $\square$

5. Distance to (Co-)Cluster Graphs and Cographs

In this section we present fixed-parameter algorithms for 2-Club parameterized by *distance to co-cluster graphs*, by *distance to cluster graphs*, and by *distance to cographs*. All these algorithms have running time $2^{\Theta(2^k)} \cdot n^{O(1)}$ which is roughly similar to the one obtained for treewidth [20]. For the weaker parameters the constants in the exponential part of the running time are smaller. Hence, none of the algorithms "dominates" one of the other algorithms even with *distance to cographs* being a stronger parameter than *distance to cluster graphs* or *distance to co-cluster* graphs (see Figure 1). As already mentioned, even for the considerably weaker parameter *vertex cover* the best known algorithm has running time $2^{\Theta(2^k)} \cdot n^{O(1)}$ [20]. In contrast, the parameter *distance to clique* which is unrelated to *vertex cover* admits a $O(2^k \cdot nm)$-time algorithm, even in case of the general s-Club problem: Each clique is a 2-club. Hence, the parameter *distance to clique* is at most as large as the dual parameter $n - \ell$. Thus, the $O(2^{n-\ell} \cdot nm)$-time algorithm [32] has the claimed running time.

5.1. Distance to Co-Cluster Graphs and Distance to Cluster Graphs

We first present an algorithm for 2-Club parameterized by the *distance to co-cluster graphs*. A graph is a co-cluster graph if its complement graphs is a cluster graph. Hence, these graphs are disjoint unions of independent sets and between different independent sets all possible edges are present.

Theorem 5. *2-Club is solvable in $O(2^k \cdot 2^{2^k} \cdot nm)$ time where k denotes the* distance to co-cluster graphs.

Proof. Let (G, ℓ, X) be a 2-Club instance where X has $|X| = k$ and $G - X$ is a co-cluster graph. Note that the co-cluster graph $G - X$ is either a connected graph or it does not contain any edge. If $G - X$ is an independent set, the set X is a vertex cover and we thus apply the algorithm we gave in companion work [20] to solve the instance in $O(2^k \cdot 2^{2^k} \cdot nm)$ time.

Hence, assume that $G - X$ is connected. Since $G - X$ is a co-cluster graph, this implies that $G - X$ is a 2-club. Thus, if $\ell \leq n - k$, then we can trivially answer yes. Hence, assume that $\ell > n - k$ or, equivalently, $k > n - \ell$. Then, applying the $O(2^{n-\ell} nm)$-time algorithm [32] directly solves the problem in $O(2^k nm)$ time. $\square$

Next, we present a fixed-parameter algorithm for the parameter *distance to cluster graphs.*

Theorem 6. 2-Club *is solvable in* $O(2^k \cdot 4^{2^k} \cdot nm)$ *time where* k *denotes* distance to cluster graphs.

Proof. Let $(G = (V, E), X, \ell)$ be guessed and cleaned input instance with $G - X$ being a cluster graph and $|X| = k$. Let $\mathcal{T} = T_1, \ldots, T_p$ be the set of twin classes of $V \setminus X$ with respect to X and let $C_1, \ldots, C_q$ denote the clusters of $G - X$. Two twin classes T and T' are in *conflict* if $N(T) \cap N(T') \cap X = \emptyset$. The three main observations exploited in the algorithm are the following: First, if two twin classes T_i and T_j are in conflict, then all vertices of T_i that are in a 2-club and all vertices from T_j that are in a 2-club must be in the same cluster of $G - X$. Second, every vertex from $G - X$ can reach all vertices in X only via vertices of X or via vertices in its own cluster. Third, if one 2-club-vertex $v \in S$ is in a twin class $v \in T_i$ and in a cluster $v \in C_j$, then all vertices that are in T_i and in C_j can be added to S without violating the 2-club property.

We exploit these observations in a dynamic programming algorithm. In this algorithm, we create a two-dimensional table $\mathcal{A}$ where an entry $\mathcal{A}[i, \mathcal{T}']$ stores the maximum size of a set $Y \subseteq \bigcup_{1 \leq j \leq i} C_j$ such that the twin classes of Y are *exactly* $\mathcal{T}' \subseteq \mathcal{T}$ and all vertices in Y have in $G[Y \cup X]$ distance at most two to each vertex from $Y \cup X$.

Before filling the table $\mathcal{A}$, we calculate a value $s(i, \mathcal{T}')$ that stores the maximum number of vertices we can add from C_i that are from the twin classes in $\mathcal{T}'$ and fulfill the requirements in the previous paragraph. This value is defined as follows. Let $C_i^{\mathcal{T}'}$ denote the maximal subset of vertices from C_i whose twin classes are exactly $\mathcal{T}'$. Then, $s(i, \mathcal{T}') = |C_i^{\mathcal{T}'}|$ if $C_i^{\mathcal{T}'}$ exists and every pair of non-adjacent vertices from $C_i^{\mathcal{T}'}$ and from X have a common neighbor. Otherwise, set $s(i, \mathcal{T}') = -\infty$. Note that as a special case we set $s(i, \emptyset) = 0$. Furthermore, for two subsets $\mathcal{T}''$ and $\tilde{\mathcal{T}}$ define the predicate $\operatorname{conf}(\mathcal{T}'', \tilde{\mathcal{T}})$ as true if there is a pair of twin classes $T_i \in \mathcal{T}''$ and $T_j \in \tilde{\mathcal{T}}$ such that T_i and T_j are in conflict, and as false, otherwise.

Using these values, we now fill $\mathcal{A}$ with the following recurrence:

$$\mathcal{A}[i, \mathcal{T}'] = \max\Big(\{-\infty\} \cup \{\mathcal{A}[i-1, \tilde{\mathcal{T}}] + s(i, \mathcal{T}'') \mid (\tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}') \wedge \neg \operatorname{conf}(\tilde{\mathcal{T}}, \mathcal{T}'')\}\Big)$$

This recurrence considers all cases of combining a set Y for the clusters C_1 to C_{i-1} with a solution Y' for the cluster C_i. Herein, a positive table entry is only obtained when the twin classes of $Y \cup Y'$ are exactly $\mathcal{T}'$ and the pairwise distances between $Y \cup Y'$ and $Y \cup Y' \cup X$ in $G[Y \cup Y' \cup X]$ are at most two. The latter property is ensured by the definition of the $s()$ values and by the fact that we consider only combinations that do not put conflicting twin classes in different clusters.

Now, the table entry $\mathcal{A}[q, \mathcal{T}']$ contains the size of a maximum vertex set Y such that in $G[Y \cup X]$ every vertex from Y has distance two to all other vertices. It remains to ensure that the vertices from X are within distance two from each other. This can be done by only considering a table entry $\mathcal{A}[q, \mathcal{T}']$ if each non-adjacent vertex pair $x, x' \in X$ has either a common neighbor in X or in one twin class contained in $\mathcal{T}'$. The maximum size of a 2-club in G is then the maximum value of all table entries that fulfill this condition.

The running time can be bounded roughly as $O(2^k \cdot 4^{2^k} \cdot nm)$: We try all 2^k partitions of X and for each of these partitions, we fill a dynamic programming table with $2^{2^k} \cdot n$

entries. The number of overall table lookups and updates is $O(4^{2^k} \cdot n)$ since there are 4^{2^k} possibilities for the sets $\mathcal{T}''$, $\tilde{\mathcal{T}}$. Since each C_i is a clique, the entry $s(i, \mathcal{T}')$ is computable in $O(nm)$ time and the overall running time follows. □

5.2. Distance to Cographs

We now describe a fixed-parameter algorithm for 2-Club parameterized by *distance to cographs*. Recall that cographs are exactly the P_4-free graphs. Hence, any connected component of a cograph is a 2-club.

Theorem 7. 2-Club *is solvable in* $O(8^k \cdot 4^{2^k} \cdot n^4)$ *time where* k *denotes the* distance to cographs.

Proof. Let (G, ℓ, X) be a guessed and cleaned 2-Club instance with $G - X$ being a cograph and $|X| = k$. For our correctness proof we fix a maximum 2-club S in G with $X \subseteq S$.

Before describing the algorithm we first introduce the following characterization of cographs [10]: A graph is a cograph if it can be constructed from single vertex graphs by a sequence of parallel and series compositions. Given t vertex disjoint graphs $G_i = (V_i, E_i)$, the *series composition* is the graph $(\bigcup_{i=1}^t V_i, \bigcup_{i=1}^t E_i \cup \{\{u, v\} \mid (u \in G_i) \wedge (v \in G_j) \wedge (1 \leq i < j \leq t)\}$ and the *parallel composition* is $(\bigcup_{i=1}^t V_i, \bigcup_{i=1}^t E_i)$. The corresponding *cotree* of a cograph G is the tree whose leaves correspond to the vertices in G and each inner node represents a series or parallel composition of its children up to a root which represents G. Furthermore, the cotree can be computed in linear time.

We next describe a dynamic programming algorithm that proceeds in a bottom-up manner on the cotree of $G - X$ and finds a maximum 2-club in G that contains X. We may assume that $t = 2$ for all series and parallel compositions, as otherwise we can simply split up the corresponding nodes in the cotree. For each node Υ in the cotree let $V(\Upsilon) \subseteq V \setminus X$ be the vertices corresponding to the leaves of the subtree rooted in Υ. Furthermore, consider the (at most 2^k many) twin classes of $V \setminus X$ with respect to X and for a subset of twin classes $\mathcal{T}$ let $V(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} T$ denote the union of all vertices in the twin classes of $\mathcal{T}$. We compute a table Γ where for any subset of twin classes $\mathcal{T}$ and any node Υ of the cotree the entry $\Gamma(\Upsilon, \mathcal{T})$ is the size of a largest set $L \subseteq V(\Upsilon) \cap V(\mathcal{T})$ that fulfills the following properties:

1. for all $T \in \mathcal{T} : T \cap L \neq \emptyset$ and
2. for all $v \in L \cup X$ and $u \in L$: $\text{dist}_{G[L \cup X]}(u, v) \leq 2$

The intention of the definition above is that the graph $G[L \cup X]$ is a "2-club-like" structure that contains a vertex from each twin class in $\mathcal{T}$ (Property 1) and for any pair of vertices, except those where both vertices are from X, have distance at most two (Property 2). Denoting the root of the cotree by r and by $\mathcal{T}_s$ the set of all twin classes that have a non-empty intersection with S, $\Gamma(r, \mathcal{T}_s) \geq |S \setminus X|$ as $S \setminus X$ trivially fulfills all properties. Reversely, for any subset of twin class $\mathcal{T}$ that contains for each pair of vertices $\{u, v\} \in X$ with $\text{dist}_{G[X]}(u, v) > 2$ a twin class $T \in \mathcal{T}$ with $\{u, v\} \subseteq N(T)$, any set corresponding to $\Gamma(r, \mathcal{T})$ forms together with X a 2-club.

We now describe the dynamic programming algorithm. Let Υ be a leaf node of the cotree with $V(\Upsilon) = \{x\}$ and let $\mathcal{T}$ be any subset of twin classes. The two sets

$\{x\}$ and $\emptyset$ are the only candidates for L. Hence we set $\Gamma(\Upsilon, \mathcal{T}) := 1$ if x fulfills both properties, $\Gamma(\Upsilon, \emptyset) := 0$ ($\emptyset$ fulfills both properties), and $\Gamma(\Upsilon, \mathcal{T}) = -\infty$ otherwise.

Next we describe the dynamic programming algorithm for inner nodes of the cotree. Let Υ be any node of the cotree with children Υ_1, Υ_2 and let $\mathcal{T}$ be any subset of twin classes. We construct a graph $G^{\Upsilon,\mathcal{T}}$ by exhaustively deleting in $G[(V(\Upsilon) \cap V(\mathcal{T})) \cup X]$ all vertices from $V(\Upsilon) \cap V(\mathcal{T})$ that have distance more than two to any vertex in X. (Clearly, such a vertex has to be deleted because of Property 2.) If the resulting graph $G^{\Upsilon,\mathcal{T}}$ violates Property 1, then there is no set corresponding to $\Gamma(\Upsilon, \mathcal{T})$ and thus we set the entry to be $-\infty$. Additionally, if $G^{\Upsilon,\mathcal{T}}$ fulfills all properties, then set $\Gamma(\Upsilon, \mathcal{T}) = |V(G^{\Upsilon,\mathcal{T}})| - |X|$. To handle the remaining case where $G^{\Upsilon,\mathcal{T}}$ violates only Property 2 we make a case distinction on the node type of Υ.

Case 1: Υ is a series node.
Let $\{u, v\} \subseteq V(G^{\Upsilon,\mathcal{T}}) \setminus X$ be a vertex pair with $\mathrm{dist}_{G^{\Upsilon,\mathcal{T}}}(u, v) > 2$. Since a series composition introduces an edge between each vertex in $V(\Upsilon_1)$ and each vertex in $V(\Upsilon_2)$ and $V(G^{\Upsilon,\mathcal{T}}) \setminus X \subseteq V(\Upsilon) = V(\Upsilon_1) \cup V(\Upsilon_2)$, it follows that either $V(G^{\Upsilon,\mathcal{T}}) \cap V(\Upsilon_1) = \emptyset$ or $V(G^{\Upsilon,\mathcal{T}}) \cap V(\Upsilon_2) = \emptyset$. This implies that $\Gamma(\Upsilon, \mathcal{T}) = \max\{\Gamma(\Upsilon_1, \mathcal{T}), \Gamma(\Upsilon_2, \mathcal{T})\}$.

Case 2: Υ is a parallel node.
Consider any set L that corresponds to $\Gamma(\Upsilon, \mathcal{T})$. By the definition of a parallel node there is no edge between a vertex from $V(\Upsilon_1)$ and a vertex in $V(\Upsilon_2)$. Consequently, any pair of vertices in L with one vertex in $V(\Upsilon_1)$ and the other in $V(\Upsilon_2)$ have a common neighbor in X. Correspondingly, we say that two twin classes are *consistent* if they have at least one common neighbor in X and two sets of twin classes are consistent if any twin class of the first set is consistent with any twin class of the second set. Denoting by $\mathcal{T}_1^S$ ($\mathcal{T}_2^S$) the set of twin classes with a non-empty intersection with $L \cap V(\Upsilon_1)$ ($L \cap V(\Upsilon_2)$), by the argumentation above it follows that $\mathcal{T}_1^S$ is consistent with $\mathcal{T}_2^S$. Additionally, it is straightforward to verify that $L \cap V(\Upsilon_1)$ ($L \cap V(\Upsilon_2)$) fulfills all properties (except being a largest set) for the entry $\Gamma(\mathcal{T}_1^S, \Upsilon_1)$ ($\Gamma(\Upsilon_2, \mathcal{T}_2^S)$).

Reversely, for any two consistent sets of twin classes $\mathcal{T}_1, \mathcal{T}_2$ let L_1 (L_2) be any vertex set that corresponds to $\Gamma(\Upsilon_1, \mathcal{T}_1)$ ($\Gamma(\Upsilon_2, \mathcal{T}_2)$). It holds that $L_1 \cup L_2$ fulfills all properties for $\Gamma(\Upsilon, \mathcal{T}_1 \cup \mathcal{T}_2)$ and hence $\Gamma(\Upsilon, \mathcal{T}_1 \cup \mathcal{T}_2) \geq |L_1 \cup L_2|$. Hence it is correct to set $\Gamma(\mathcal{T}, \Upsilon)$ to be the largest value of $\Gamma(\Upsilon_1, \mathcal{T}_1) + \Gamma(\Upsilon_2, \mathcal{T}_2)$ where $\mathcal{T}_1, \mathcal{T}_2$ are consistent and $\mathcal{T}_1 \cup \mathcal{T}_2 = \mathcal{T}$. This completes the description of the algorithm.

The table Γ contains $O(n \cdot 2^{2^k})$ entries as there are at most 2^k twin classes. Each entry can be computed in $O(n^3 + 3^{|\mathcal{T}|}|\mathcal{T}|^2 k)$ time: Graph $G^{\Upsilon,\mathcal{T}}$ can be computed in $O(n^3)$ time. The running time of Case 2 dominates those of Case 1 and in Case 2 there are at most $3^{|\mathcal{T}|}$ possibilities to split $\mathcal{T}$ into two (not necessarily disjoint) sets $\mathcal{T}_1, \mathcal{T}_2$ and the check of consistency can be done in $O(|\mathcal{T}|^2 k)$. In total the time spent in Case 2 is $O(n \cdot \sum_{i=0}^{2^k} \binom{2^k}{i}(3^i i^2 k + n^3)) = O(4^{2^k} \cdot 4^k n^4)$. Together with the factor of 2^k needed to guess X in the guess and cleaning step, the running time of the above algorithm is $O(8^k \cdot 4^{2^k} \cdot n^4)$. □

6. Conclusion

We have resolved the complexity status of 2-Club for most of the parameters in the complexity landscape shown in Figure 1. Still, several open questions remain. First, there are obviously parameters for which the parameterized complexity is still open. For example, is 2-Club parameterized by *distance to interval graphs* or by *distance to 2-club cluster graphs* in XP or even fixed-parameter tractable? In this context, also parameter combinations could be of interest. Since a complete investigation of the parameter space is infeasible one should focus on practically relevant parameter combinations. One example could be the following question that is left open by the hardness results for *h-index* and *degeneracy*. Is 2-Club parameterized by the *h-index* still W[1]-hard if the input graph has *constant degeneracy*? Second, it remains open whether there is a polynomial kernel for the parameter *distance to clique* or to identify further non-trivial structural parameters for which polynomial kernels exist. Third, for many of the presented fixed-parameter tractability results it would be interesting to either improve the running times or to obtain tight lower bounds. For example, is it possible to solve 2-Club parameterized by *distance to clique* in $\delta^k \cdot n^{O(1)}$ time for some $\delta < 2$? Similarly, is it possible to solve 2-Club parameterized by *vertex cover* in $2^{o(2^k)} \cdot n^{O(1)}$ time? An answer to the latter question could be a first step towards improving the (also doubly exponential) running time of the algorithms for the parameters *treewidth* or *distance to cographs*. Finally, it would be interesting to see which results carry over to 3-Club [25, 30] or to the related 2-Clique problem [4].

References

[1] R. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3 (1):113–126, 1973.

[2] M. T. Almeida and F. D. Carvalho. Integer models and upper bounds for the 3-club problem. *Networks*, 60(3):155–166, 2012.

[3] Y. Asahiro, E. Miyano, and K. Samizo. Approximating maximum diameter-bounded subgraphs. In *Proc. 9th LATIN*, volume 6034 of *LNCS*, pages 615–626. Springer, 2010.

[4] B. Balasundaram, S. Butenko, and S. Trukhanovzu. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[5] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.

[6] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.

[7] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for structural parameterizations of pathwidth. In *Proc. 13th SWAT*, volume 7357 of *LNCS*, pages 352–363. Springer, 2012.

[8] J.-M. Bourjolly, G. Laporte, and G. Pesant. Heuristics for finding k-clubs in an undirected graph. *Computers & Operations Research*, 27(6):559–569, 2000.

[9] J.-M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21–28, 2002.

[10] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: a Survey*, volume 3 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 1999.

[11] F. D. Carvalho and M. T. Almeida. Upper bounds and heuristics for the 2-club problem. *European Journal of Operational Research*, 210(3):489–494, 2011.

[12] M.-S. Chang, L.-J. Hung, C.-R. Lin, and P.-C. Su. Finding large k-clubs in undirected graphs. *Computing*, 95(9):739–758, 2013.

[13] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005.

[14] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

[15] D. Eppstein and E. S. Spiro. The h-index of a graph and its application to dynamic subgraph statistics. *Journal of Graph Algorithms and Applications*, 16(2):543–567, 2012.

[16] M. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

[17] M. R. Fellows, B. M. P. Jansen, and F. A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013.

[18] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[19] P. A. Golovach, P. Heggernes, D. Kratsch, and A. Rafiey. Finding clubs in graph classes. *Discrete Applied Mathematics*, 174:57–65, 2014.

[20] S. Hartung, C. Komusiewicz, and A. Nichterlein. Parameterized algorithmics and computational experiments for finding 2-clubs. In *Proc. 7th IPEC*, volume 7535 of *LNCS*, pages 231–241. Springer, 2012.

[21] S. Hartung, C. Komusiewicz, and A. Nichterlein. On structural parameterizations for the 2-club problem. In *Proc. 39th SOFSEM*, volume 7535 of *LNCS*, pages 231–241. Springer, 2013.

[22] B. M. P. Jansen and S. Kratsch. On polynomial kernels for structural parameterizations of odd cycle transversal. In *Proc. 6th IPEC*, volume 7112 of *LNCS*, pages 132–144. Springer, 2011.

[23] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[24] C. Komusiewicz and R. Niedermeier. New races in parameterized algorithmics. In *Proc. 37th MFCS*, volume 7464 of *LNCS*, pages 19–30. Springer, 2012.

[25] F. Mahdavi and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9:84–97, 2012.

[26] N. Memon and H. L. Larsen. Structural analysis and mathematical methods for destabilizing terrorist networks using investigative data mining. In *Proc. 2nd ADMA*, volume 4093 of *LNCS*, pages 1037–1048. Springer, 2006.

[27] R. J. Mokken. Cliques, Clubs and Clans. *Quality and Quantity*, 13:161–173, 1979.

[28] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[29] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. 27th STACS*, volume 5 of *LIPIcs*, pages 17–32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

[30] S. Pasupuleti. Detection of protein complexes in protein interaction networks using n-Clubs. In *Proc. 6th EvoBIO*, volume 4973 of *LNCS*, pages 153–164. Springer, 2008.

[31] R. Sasák. Comparing 17 graph parameters. Master's thesis, University of Bergen, 2010. See https://robert.sasak.sk/gpp/index.php for the corresponding *Graph Parameter Project*.

[32] A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

[33] A. Schäfer. Exact algorithms for s-club finding and related problems. Diploma thesis, Friedrich-Schiller-Universität Jena, 2009.

[34] S. B. Seidman and B. L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.

[35] M. Sorge and M. Weller. The graph parameter hierarchy. Manuscript, 2013. URL http://fpt.akt.tu-berlin.de/msorge/parameter-hierarchy.pdf.