Graph functionality*

Bogdan Alecu† Aistis Atminas‡ Vadim Lozin§

Abstract

In the present paper, we introduce the notion of graph functionality, which generalises simultaneously several other graph parameters, such as degeneracy or clique-width, in the sense that bounded degeneracy or bounded clique-width imply bounded functionality. Moreover, we show that this generalisation is proper by revealing classes of graphs of unbounded degeneracy and clique-width, where functionality is bounded by a constant. We also prove that bounded functionality implies bounded VC-dimension, i.e., graphs of bounded VC-dimension extend graphs of bounded functionality, and this extension is also proper.

Keywords: Clique-width; Permutation graph; Hereditary class; Graph representation

1 Introduction

Let $G = (V, E)$ be a simple graph, i.e., an undirected graph without loops and multiple edges. We denote by $A = A_G$ the adjacency matrix of G and by $A(x, y)$ the element of this matrix corresponding to vertices $x, y \in V$, i.e., $A(x, y) = 1$ if x and y are adjacent, and $A(x, y) = 0$ otherwise.

Let us now introduce the central notion studied in this paper:

Definition 1. We say that a vertex $y \in V$ is *a function of* vertices $x_1, \dots, x_k \in V$ if there exists a Boolean function f of k variables such that for any vertex $z \in V - \{y, x_1, \dots, x_k\}$, we have $A(y, z) = f(A(x_1, z), \dots, A(x_k, z))$. The *functionality* $\mathrm{fun}(y)$ of a vertex y is the minimum k such that y is a function of k vertices.

In particular, the functionality of an isolated vertex is 0, and the same is true for a dominating vertex, i.e., a vertex adjacent to all the other vertices in the graph. More generally, the functionality of a vertex y does not exceed the number of its neighbours (the degree of y) and the number of its non-neighbours. One more simple example of vertices of small

*Some results presented in this paper appeared in the extended abstract [1] published in the proceedings of the 45th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2019.

†Mathematics Institute, University of Warwick, Coventry, CV4 7AL, UK. Email: B.Alecu@warwick.ac.uk

‡Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, 111 Ren'ai Road, Suzhou 215123, China. Email: Aistis.Atminas@xjtlu.edu.cn

§Mathematics Institute, University of Warwick, Coventry, CV4 7AL, UK. Email: V.Lozin@warwick.ac.uk

functionality is given by twins, i.e., vertices x and y that have the same set of neighbours different from x and y. Twins are functions of each other, and their functionality is (at most) 1. The same is true for anti-twins, i.e., vertices whose neighbourhoods complement each other.

From a practical point of view, representing vertex's adjacency as a function can be of interest in the area of graph learning and graph mining, since it makes graphs amenable to the techniques of Logical Analysis of Data [13], which is based on Boolean methods for pattern detection. This approach provides a tool for revealing dependencies that are hidden in the structure of the graph and for identifying alliances that are more complex than "friends" or "enemies".

From a theoretical point of view, the importance of this approach is due to the fact that it defines a new complexity measure, which we call *graph functionality*, and that we define as follows:

Definition 2. The *functionality* $\mathrm{fun}(G)$ of a graph G is

$$\max_{H} \min_{y \in V(H)} \mathrm{fun}(y),$$

where the maximum is taken over all induced subgraphs H of G.

Functionality is defined by analogy with degeneracy, which it generalises: if we replace $\mathrm{fun}(y)$ with $\mathrm{degree}(y)$ in the above definition, we obtain the degeneracy of G. Taking the maximum over induced subgraphs ensures that functionality never increases when taking induced subgraphs.

Similarly to many other graph parameters, the notion of graph functionality becomes valuable when its value is small, i.e., is bounded by a constant independent of the size of the graph. In particular, graphs of small functionality admit compact representation, as was shown in [2]. That paper does not formally define the notion of graph functionality, but the results proved there imply that graphs of bounded functionality can be represented by binary words of length $O(n \log_2 n)$. The same is true for various other parameters, such as vertex degree, degeneracy, arboricity, tree-width or clique-width. In this paper, we show that the notion of graph functionality provides a common generalisation of all of them in the sense that bounded vertex degree, degeneracy, arboricity, tree-width or clique-width implies bounded functionality. We prove this in Section 2. Moreover, in the same section we show that this generalisation is proper by revealing classes of graphs where functionality is bounded but the other parameters are not. This includes permutation graphs, line graphs and, more generally, the intersection graphs of 3-uniform hypergraphs. On the other hand, in Section 3 we show that bounded functionality implies bounded VC-dimension, i.e., graphs of bounded VC-dimension extend graphs of bounded functionality, and this extension also is proper.

Throughout the paper, we consider only simple graphs and use standard terminology and notation. In particular, for a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. The neighbourhood $N(v)$ of a vertex $v \in V(G)$ is the set of vertices of G adjacent to v, and the degree of v is $|N(v)|$. A vertex of degree 0 is called *isolated*. The closed neighbourhood of v is $N[v] = \{v\} \cup N(v)$. A chordless cycle of length n is denoted C_n. A graph H is an *induced subgraph* of a graph G if H can be obtained from G by vertex deletions. A class X of graphs is *hereditary* if it is closed under taking induced subgraphs. We also define a parameter to be *hereditary* if it never increases when taking induced subgraphs. Note that all parameters in this paper are hereditary, including functionality, as noted earlier.

2 Graphs of small functionality

From the discussion in the introduction, it follows that graphs of bounded functionality extend graphs of bounded vertex degree. More generally, they extend graphs of bounded degeneracy, where the *degeneracy* of G is the minimum k such that every induced subgraph of G has a vertex of degree at most k. A notion related to degeneracy is that of *arboricity*, which is the minimum number of forests into which the edges of G can be partitioned. The degeneracy of G is always between the arboricity and twice the arboricity of G and hence graphs of bounded functionality extend graphs of bounded arboricity too.

One more important graph parameter is *clique-width*. Many algorithmic problems that are generally NP-hard become polynomial-time solvable when restricted to graphs of bounded clique-width [7]. Clique-width is a relatively new notion and it generalises another important graph parameter, tree-width, studied in the literature for decades. Clique-width is stronger than tree-width in the sense that graphs of bounded tree-width have bounded clique-width. In Section 2.1, we show that functionality is stronger than clique-width by proving that graphs of bounded clique-width have bounded functionality. Then in Sections 2.3 and 2.4, we identify classes of graphs where functionality is bounded, but degeneracy and clique-width are not.

2.1 Graphs of bounded clique-width

The notion of clique-width of a graph was introduced in [6]. The clique-width of a graph G is denoted $\mathrm{cwd}(G)$ and is defined as the minimum number of labels needed to construct G by means of the following four graph operations:

- creation of a new vertex v with label i (denoted $i(v)$),
- disjoint union of two labelled graphs G and H (denoted $G \oplus H$),
- connecting vertices with specified labels i and j (denoted $\eta_{i,j}$) and
- renaming label i to label j (denoted $\rho_{i \to j}$).

Every graph can be defined by an algebraic expression using the four operations above. This expression is called a k-expression if it uses k different labels. For instance, the cycle C_5 on vertices a, b, c, d, e (listed along the cycle) can be defined by the following 4-expression:

$$\eta_{4,1}(\eta_{4,3}(4(e) \oplus \rho_{4\to 3}(\rho_{3\to 2}(\eta_{4,3}(4(d) \oplus \eta_{3,2}(3(c) \oplus \eta_{2,1}(2(b) \oplus 1(a)))))))).$$

Alternatively, any algebraic expression defining G can be represented as a rooted binary tree, whose leaves correspond to the operations of vertex creation, the internal nodes correspond to the $\oplus$-operations, and the root is associated with G. The operations η and ρ are assigned to the respective edges of the tree. Figure 1 shows the tree representing the above expression defining a C_5.

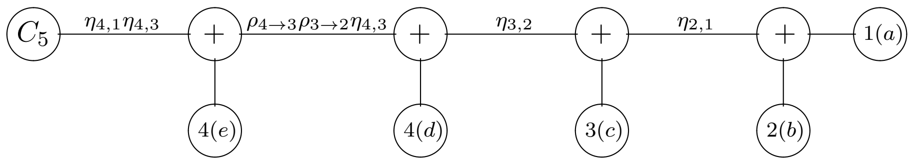


Figure 1: The tree representing the expression defining a C_5

Among various examples of graphs of bounded clique-width we mention distance-hereditary graphs. These are graphs of clique-width at most 3 [9]. Every graph in this class can be constructed from a single vertex by successively adding either a pendant vertex or a twin (true or false) [5]. From this characterization we immediately conclude that the functionality of distance-hereditary graphs is at most one. More generally, in the next theorem we show that functionality is bounded for all classes of graphs of bounded clique-width.

Theorem 1. *For any graph G,* $\mathrm{fun}(G) \leq 2\mathrm{cwd}(G) - 1$.

Proof. Let G be a graph of clique-width k and let T be a rooted tree corresponding to a k-expression that describes G. For a node v of the rooted tree T, let T_v be a subtree of T induced by the node v and all its descendants. We can choose v in such a way that T_v has more than k leaves, and neither of the two children of v has this property (if no such v exists, we are done, since G has at most k vertices). Since T_v has more than k leaves, at least two of them, say x and y, have the same label at node v. On the other hand, T_v has at most $2k$ leaves by the choice of v. Therefore, G contains at most $2k - 2$ vertices that distinguish x and y, since x and y are not distinguished outside of T_v. As a result, the functionality of both x and y is at most $2k - 1$.

It is known (see, e.g., [8]) that the clique-width of an induced subgraph of G cannot exceed the clique-width of G. Therefore, every induced subgraph of G has a vertex of functionality at most $2k - 1$. Thus, the functionality of G is at most $2k - 1$. □

In the proof of Theorem 1, the bound on functionality is achieved in a very specific way: in any graph of bounded clique width, there must exist two vertices whose neighbourhoods have small symmetric difference. In the next section, we formalise this idea by introducing a parameter which is intermediate between clique-width and functionality.

2.2 Graphs of bounded symmetric difference

Given a graph G and a pair of vertices x, y in G, let us denote by $\mathrm{sd}(x, y)$ the number of vertices different from x and y that are adjacent to exactly one of x and y. In other words, if x and y are non-adjacent, then $\mathrm{sd}(x,y)$ is the size of the symmetric difference of $N(x)$ and $N(y)$. We will refer to $\mathrm{sd}(x,y)$ as the size of the symmetric difference even if x and y are adjacent. Also, we write

$$\mathrm{sd}(G) = \max_{H} \min_{x,y \in V(H)} \mathrm{sd}(x,y),$$

where the maximum is taken over all induced subgraphs H of G. With some abuse of terminology we call $\mathrm{sd}(G)$ the *symmetric difference* of G.

Implicitly, this parameter was used in the proof of Theorem 1, which shows that bounded clique-width implies bounded symmetric difference. We also observed in the proof that bounded symmetric difference implies bounded functionality. Let us state both of these facts in our new terminology.

Theorem 2. *For any graph G, $\mathrm{sd}(G) \leq 2\,\mathrm{cwd}(G) - 2$.*

Proof. This follows immediately from the proof of Theorem 1. □

Theorem 3. *For any graph G, $\mathrm{fun}(G) \leq \mathrm{sd}(G) + 1$.*

Below, we show that both implications are proper. In particular, in the rest of this section we describe a class of graphs of bounded symmetric difference and unbounded clique-width. Also, in Sections 2.3 and 2.4 we describe classes of graphs of bounded functionality and unbounded symmetric difference.

Unit interval graphs have bounded symmetric difference

A unit interval graph is the intersection graph of intervals of the same length on the real line. In this class clique-width is unbounded [9], and so is degeneracy. We now prove that symmetric difference and hence functionality of unit interval graphs is bounded. This shows, in particular, that bounded symmetric difference does not imply bounded clique-width.

Theorem 4. *The symmetric difference of unit interval graphs is at most 1 and functionality is at most 2.*

Proof. Let G be a unit interval graph with n vertices and assume without loss of generality that G has no isolated vertices (by adding isolated vertices to a graph we increase neither its functionality nor symmetric difference). Take a unit interval representation for $G = (V, E)$ with the interval endpoints all distinct. We label the vertices $v_1, \ldots, v_n$ in the order in which they appear on the real line (from left to right), and denote the endpoints of interval I_i corresponding to vertex v_i by $a_i < b_i$. We will bound

$$S = \sum_{i=1}^{n-1} \mathrm{sd}(v_i, v_{i+1}).$$

Note that any neighbour of v_i which is not a neighbour of v_{i+1} needs to have its right endpoint between a_i and a_{i+1}. Similarly, any neighbour of v_{i+1} but not of v_i needs to have its left endpoint between b_i and b_{i+1}. In other words, $\mathrm{sd}(v_i, v_{i+1})$ is bounded above by the number of endpoints in $(a_i, a_{i+1}) \cup (b_i, b_{i+1})$ (we say bounded above and not equal, since it might happen that b_i lies between a_i and a_{i+1}, without contributing to the symmetric difference).

The key is now to note that any endpoint can be counted at most once in the whole sum S, since all (a_i, a_{i+1}) are disjoint (and the same applies to the (b_i, b_{i+1})), and the a's can only appear between b's (and vice-versa). In fact, a_1 and b_n are never counted in S, and if a_2 is between b_1 and b_2, then v_1 must be isolated, so a_2 is not counted either. The sum is thus at most $2n-3$. Since it has $n-1$ terms, one of the terms, say $\mathrm{sd}(v_t, v_{t+1})$, must be at most 1. Therefore, the functionality of both v_t and v_{t+1} is at most 2.

Since the class of unit interval graphs is hereditary, we conclude that the symmetric difference of any unit interval graph is at most 1 and the functionality is at most 2. □

We remark that 2 is the best bound on functionality we can obtain for this class, since we can easily construct unit interval graphs where each vertex has degree and co-degree greater than 1, and with no twins or anti-twins.

2.3 Permutation graphs

Let π be a permutation of the elements in $\{1, 2, \ldots, n\}$. The permutation graph of π is a graph with vertex set $\{1, 2, \ldots, n\}$ in which two vertices i and j are adjacent if and only if $(i-j)(\pi(i)-\pi(j)) < 0$. Clique-width is known to be unbounded in the class of permutation graphs [9], and so is degeneracy.

For the purpose of this section, we associate a permutation π with its plot, i.e., the set of points $(i, \pi(i))$ in the plane. We label those points by $\pi(i)$ and define the *geometric neighbourhood* of a point k to be the union of two regions in the plane: the one above and to its left, and the one below and to its right. Then it is not difficult to see that the set of points of the permutation lying in the geometric neighbourhood of k is precisely the set of neighbours of vertex k in the permutation graph of π.

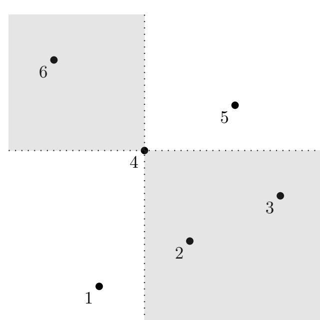


Figure 2: Geometric representation of $\pi = 614253$, with the neighbourhood of 4 shaded

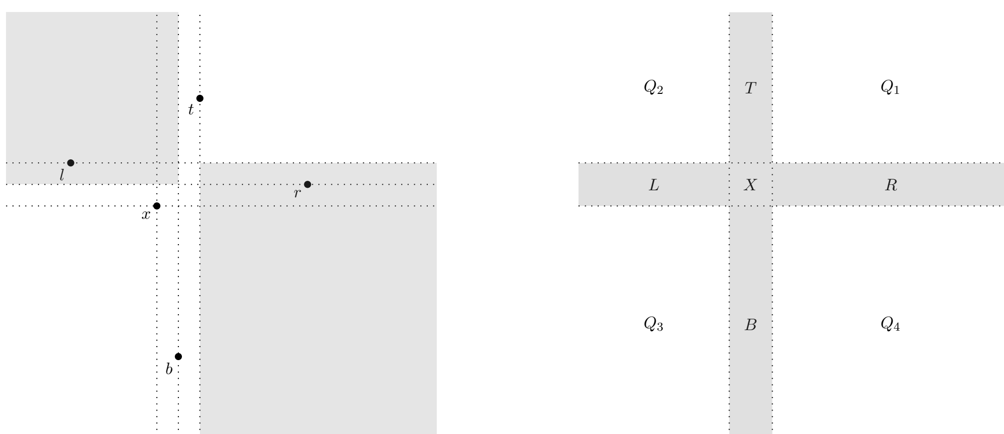
Theorem 5. *The functionality of permutation graphs is at most 8.*

Proof. Since the class of permutation graphs is hereditary, it suffices to show that every permutation graph contains a vertex of functionality at most 8. Let G be a permutation graph corresponding to a permutation π. The proof will be given in two steps: first, we show that if there is a vertex with a certain property in G (yet to be specified), then this vertex is a function of 4 other vertices. Second, we show how to find vertices that are "close enough" to having that property.

Step 1: Consider the plot of π. Among any 3 horizontally consecutive points, one is vertically between the two others. We call such a point *vertical middle* (in the permutation from Figure 2, the vertical middle points are 4, 2 and 3). Similarly, among any 3 vertically consecutive points, one is horizontally between the two others, and we call this point *horizontal middle* (in Figure 2, the horizontally middle points are 2, 5 and 4).

Now let us suppose that π has a point x that is simultaneously a horizontal and a vertical middle point. Then x is part of a triple x, b, t (not necessarily in that order) of horizontally consecutive points, where b is the bottom point (the lowest in the triple) and t is the top point (the highest in the triple). Also, x is part of a triple x, l, r (not necessarily in that order) of vertically consecutive points, where l is the leftmost and r is the rightmost point in the triple (see Figure 3a for an illustration).

In general, x can be at any of the 9 intersection points of pairs of 3 consecutive vertical and horizontal lines, i.e., x is somewhere in X (see Figure 3b). We also have $l \in L$, $r \in R$, $t \in T$ and $b \in B$ for the surrounding points (see Figure 3b). The important thing to note is that, since the points are consecutive, those are the *only* points of the permutation lying in the shaded area $X \cup L \cup R \cup T \cup B$. Any point different from x, l, r, t, b lies in one of Q_1, Q_2, Q_3 or Q_4.



(a) The geometric neighbourhood corresponding to $(N(r) \cap N(b)) \cup (N(l) \cap N(t))$

(b) Partition of the plot

Figure 3: A middle point x and its four surrounding points

It is not difficult to see that the geometric neighbourhood corresponding to $(N(r) \cap N(b)) \cup (N(l) \cap N(t))$ (see Figure 3a) will always contain Q_2 and Q_4, and will never intersect

Q_1 or Q_3. Therefore, the function that describes how x depends on $\{l, r, t, b\}$ can be written as follows:

$$f(x_r, x_b, x_l, x_t) = x_r x_b \vee x_l x_t,$$

where x_r, x_b, x_l, x_t are Boolean variables corresponding to points r, b, l, t, respectively, and the Boolean AND is simply denoted by juxtaposition of the variables. In other words, a vertex $y \notin \{x, l, r, t, b\}$ is adjacent to x if and only if

$$f(A(y, r), A(y, b), A(y, l), A(y, t)) = 1.$$

Step 2: Let us relax the simultaneous middle point condition to the following one: amongst every 5 vertically (respectively horizontally) consecutive points, call the middle three *weak horizontal* (respectively *vertical*) *middle points*. For instance, in Figure 2, the weak horizontal middle points are 4, 2 and 5 and the weak vertical ones are 4, 2, 5 and 3. Note that if the number of points is divisible by 5, at least $\frac{3}{5}$ of them are weak vertical and at least $\frac{3}{5}$ of them are weak horizontal middle points. Using this observation it is not hard to deduce that if there are at least 13 points, then more than half of them are weak vertical and more than half of them are weak horizontal middle points. Therefore, there must exist a point x that is simultaneously both. We can deal with this case only, as the functionality of any graph on at most 12 vertices is at most 6, which is due to the fact that every vertex has at most 6 neighbours or non-neighbours. If x is simultaneously a weak vertical and a weak horizontal middle point, then there must exist quintuples l, x, m_1, m_2, r and t, x, m_3, m_4, b (not necessarily in that order), where x is a simultaneous weak middle point in both directions, while m_1, m_2, m_3 and m_4 are the other weak middle points in their respective quintuples. By removing m_1, m_2, m_3 and m_4 from the graph, we find ourselves in the configuration of Step 1 and conclude that x is a function of $\{l, r, t, b\}$ in the reduced graph. Therefore, in the original graph x is a function of $\{l, r, t, b, m_1, m_2, m_3, m_4\}$, concluding the proof. □

Permutation graphs have unbounded symmetric difference

In this section, we show that symmetric difference in the class of permutation graphs is unbounded. Together with Theorem 5 this proves that bounded functionality does not imply bounded symmetric difference, and together with Theorem 1 this gives an alternative proof of the known fact that permutation graphs have unbounded clique-width.

Theorem 6. *For any $t \in \mathbb{N}$, there is a permutation graph G with $\mathrm{sd}(G) \geq t$.*

Proof. Given two vertices x_1 and x_2 of a permutation graph G, the symmetric difference of their neighbourhoods can be represented geometrically as an area in the plane (see Figure 4). More precisely, a vertex different from x_1 and x_2 lies in the symmetric difference of their neighbourhoods if and only if the corresponding point of the permutation lies in the shaded area.

In order to prove the theorem, it suffices, for each $t \in \mathbb{N}$, to exhibit a set S_t of points in the plane (with no two on the same vertical or horizontal line) such that for any pair

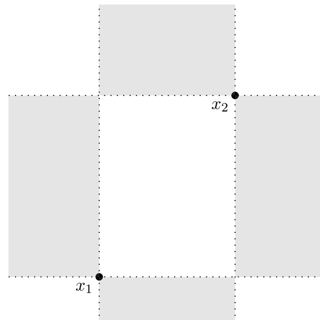


Figure 4: Geometric symmetric difference of two points x_1 and x_2

$x_1, x_2 \in S_t$, there are at least t other points of S_t lying in the geometric symmetric difference of x_1 and x_2. Such a construction immediately gives rise to a permutation and thus a permutation graph where the symmetric difference of the neighbourhoods of any pair of vertices is at least t.

We construct sets S_t in the following way (see Figure 5 for an example):

- start with all the points with integer coordinates between 0 and t inclusive;
- apply to those points the counterclockwise rotation about the origin sending the vector $(1, 0)$ to the unit vector with direction $(t + 1, 1)$ (applying this rotation ensures none of the points share a horizontal or a vertical line).

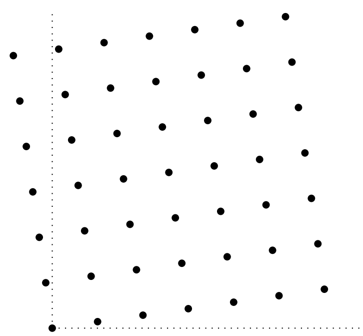


Figure 5: The set S_6

To see that these sets have indeed the desired property, let $x_1, x_2 \in S_t$. For simplicity, we will use the coordinates of the points before the rotation. Suppose $x_1 = (a_1, b_1)$ and $x_2 = (a_2, b_2)$. There are four possible cases (after switching x_1 and x_2 if necessary):

- If $a_1 = a_2$ and $b_1 < b_2$, then the t points (k, b_2), (l, b_1) with $k < a_1 < l$ are in the symmetric difference.

- Similarly, if $b_1 = b_2$ and $a_1 < a_2$, then the t points (a_1, k), (a_2, l) with $k < b_1 < l$ are in the symmetric difference.
- If $a_1 < a_2$ and $b_1 < b_2$, the following points all lie in the symmetric difference of x_1 and x_2:

  (1) Points (a_1, k) with $k < b_1$ (in the bottom region).

  (2) Points (a_1, k) with $b_1 < k \leq b_2$ (in the left region).

  (3) Points (a_2, k) with $b_2 < k$ (in the top region).

  (4) Points (a_2, k) with $b_1 \leq k < b_2$ (in the right region).

  In particular, (1) and (3) account for at least $b_1 + t - b_2$ points, while (2) and (4) account for $2(b_2 - b_1)$ others. We conclude that in total, at least $t + (b_2 - b_1) > t$ points lie in the symmetric difference of x_1 and x_2.
- If $a_1 < a_2$ and $b_1 > b_2$, the following points all lie in the symmetric difference of x_1 and x_2:

  (1) Points (k, b_2) with $a_1 \leq k < a_2$ (in the bottom region).

  (2) Points (k, b_1) with $k < a_1$ (in the left region).

  (3) Points (k, b_1) with $a_1 < k \leq a_2$ (in the top region).

  (4) Points (k, b_2) with $a_2 < k$ (in the right region).

  Summing up, we find again at least t points in the symmetric difference of x_1 and x_2.

□

2.4 Intersection graphs

The *line graph* of a graph G is the intersection graph of its edges. It is known (see, e.g., [11]) that clique-width is unbounded in the class of line graphs. The same is true for degeneracy, since line graphs contain arbitrarily large cliques.

Theorem 7. *The functionality of line graphs is at most 6.*

Proof. Let G be a graph and H be the line graph of G. Since the class of line graphs is hereditary, it suffices to prove that H has a vertex of functionality at most 6. We will prove a stronger result showing that *every* vertex of H has functionality at most 6.

Let x be a vertex in H, i.e., an edge in G. We denote the two endpoints of this edge in G by a and b. Assume first that both the degree of a and the degree of b are at least 4. Let $Y = \{y_1, y_2, y_3\}$ be a set of any three edges of G different from x that are incident to a, and let $Z = \{z_1, z_2, z_3\}$ be a set of any three edges of G different from x that are incident to b.

We claim that a vertex $v \notin \{x\} \cup Y \cup Z$ is adjacent to x in H if and only if it is adjacent to every vertex in Y or to every vertex in Z. Indeed, if v is adjacent to x in H, then the edge v intersects the edge x in G. If the intersection consists of a, then v is adjacent to every vertex in Y in the graph H, and if the intersection consists of b, then v is adjacent to every vertex in Z in the graph H. Conversely, let v be adjacent to every vertex in Y, then v must intersect the edges y_1, y_2, y_3 in G at vertex a, in which case v is adjacent to x in H. Similarly, if v is adjacent to every vertex in Z, then v intersects the edges z_1, z_2, z_3 in G at vertex b and hence v is adjacent to x in H.

Therefore, in the case when both a and b have degree at least 4 in G, the function that describes how x depends on $\{y_1, y_2, y_3, z_1, z_2, z_3\}$ in the graph H can be written as follows: $f(y_1, y_2, y_3, z_1, z_2, z_3) = y_1y_2y_3 \vee z_1z_2z_3$.

If the degree of a is less than 4, we include in Y all the edges of G distinct from x which are incident to a (if there are any) and remove the term $y_1y_2y_3$ from the function. Similarly, if the degree of b is less than 4, we include in Z all the edges of G distinct from x which are incident to b (if there are any) and remove the term $z_1z_2z_3$ from the function. If both terms have been removed, the function is defined to be identically 0, i.e., no vertices are adjacent to x in H, except for those in $Y \cup Z$. □

Having proved that the intersection graph of edges, i.e., the intersection graph of a family of 2-subsets, has bounded functionality, it is natural to ask whether the intersection graph of a family of k-subsets has bounded functionality for $k > 2$. This question is substantially harder and we answer it only for $k = 3$.

Line graphs of 3-uniform hypergraphs have bounded functionality

We will denote a 3-uniform hypergraph with the ground set V by $(V, \mathcal{S})$, where $\mathcal{S}$ is a set of 3-element subsets of V (the *hyperedges*). We will use variables $s, s', s_1, s_2, \ldots$ to denote hyperedges, i.e., the elements of $\mathcal{S}$, and variables $v, v_1, v_2, \ldots$ to denote the elements of V. We will say that two hyperedges s and s' intersect if $s \cap s' \neq \emptyset$. We start with a preparatory result.

Lemma 1. *Let $(V, \mathcal{S})$ be a 3-uniform hypergraph and $v \in V$. Then one of the following holds:*

- *There are 3 hyperedges s_1, s_2, s_3 such that $s_i \cap s_j = \{v\}$ for all $1 \leq i < j \leq 3$.*
- *There are 4 vertices v_1, v_2, v_3, v_4 such that each hyperedge $s \in \mathcal{S}$ that contains v also contains at least one of the v_1, v_2, v_3 or v_4.*

Proof. Consider the set $\mathcal{E} = \{s \backslash \{v\} : s \in \mathcal{S}, v \in s\}$. This is the set of pairs of vertices that are obtained by removing vertex v from the hyperedges that contain v. Therefore, $(V, \mathcal{E})$ can be viewed as a graph. The lemma now says that either this graph contains a matching with 3 edges (as a subgraph) or it contains 4 vertices that any edge is adjacent to (vertex cover of size 4). The proof of this is now easy. One can take a maximal matching M, and if it has at least 3 edges, then we are done. In the other case, when the maximal matching M has at most two edges, take v_i's to be the vertices of the matching. If needed, add arbitrary vertices to obtain a set of 4 vertices. By maximality of the matching, every hyperedge contains at least one of the vertices selected, hence we are done as well. □

The following two easy observations will be needed in the course of the proof.

Observation 1. Let $(V, \mathcal{S})$ be a 3-uniform hypergraph. Suppose hyperedges $s_1, s_2, s_3 \in \mathcal{S}$ pairwise intersect at exactly one vertex, say $\{v\} = s_1 \cap s_2 = s_2 \cap s_3 = s_3 \cap s_1$. In other words, $s_1 = \{v, v_1, v_2\}$, $s_2 = \{v, v_3, v_4\}$, $s_1 = \{v, v_5, v_6\}$, for some distinct vertices $v_1, v_2, \ldots, v_6$. Let $F' = \{(v_i, v_j, v_k) : 1 \leq i \leq 2, 3 \leq j \leq 4, 5 \leq k \leq 6\}$ be the set of 8 hyperedges that intersect each of s_1, s_2, s_3 in exactly one vertex that is different from v. Then one can easily determine whether a given edge $s' \in \mathcal{S} \backslash F'$ contains vertex v or not by looking at the intersection of s' with s_1, s_2, s_3. Indeed, s' contains v if and only if s' intersects each of s_1, s_2 and s_3.

Observation 2. Let $(V, \mathcal{S})$ be a 3-uniform hypergraph. Suppose hyperedges $s_1, s_2, s_3 \in \mathcal{S}$ pairwise intersect at exactly 2 vertices. In other words, $s_1 = \{v_1, v_2, v_3\}$, $s_2 = \{v_1, v_2, v_4\}$ and $s_3 = \{v_1, v_2, v_5\}$, for some distinct vertices $v_1, v_2, v_3, v_4, v_5 \in V$. Let F' be the set containing the hyperedge $\{v_3, v_4, v_5\}$. Then one can easily determine whether a given edge $s' \in \mathcal{S} \backslash F'$ contains at least one of the vertices v_1, v_2 or not by looking at the intersection of s' with s_1, s_2, s_3. Indeed, s' contains v_1, v_2 or both if and only if s' intersects each of s_1, s_2 and s_3.

Definition 3. Let $(V, \mathcal{S})$ be a 3-uniform hypergraph and let $v_1, v_2 \in V$. We will call the pair $\{v_1, v_2\}$ *thick* if there are at least 32 hyperedges in $\mathcal{S}$ that contain $\{v_1, v_2\}$.

We will split our analysis into two cases. In the first lemma we will show that the intersection graphs of 3-uniform hypergraphs without thick pairs have bounded functionality. In the second lemma we will provide a structural theorem about hypergraphs containing thick pairs, from which a bounded functionality result follows easily as well. We note that in the case without thick pairs, we provide a bound on functionality for *any* vertex of the intersection graph. Meanwhile, in the case of hypergraphs with thick pairs, for any given bound M one can find a hypergraph and a hyperedge such that corresponding vertex in the intersection graph has functionality at least M. Thus a structural result is needed in this case, to show that we can find a *particular* hyperedge in any given hypergraph with thick pairs, such that the functionality of the vertex corresponding to the hyperedge is bounded by a constant, that does not depend on the hypergraph.

We start with the case when there are no thick pairs.

Lemma 2. *Let $(V, \mathcal{S})$ be a 3-uniform hypergraph without thick pairs. Then for any hyperedge $s \in \mathcal{S}$ there is a set of hyperedges $F \subset \mathcal{S} \backslash \{s\}$ of size $|F| \leq 462$ such that for any $s' \in \mathcal{S} \backslash (F \cup \{s\})$, one can determine whether s' intersects s by looking at the intersections of s' with the hyperedges of F.*

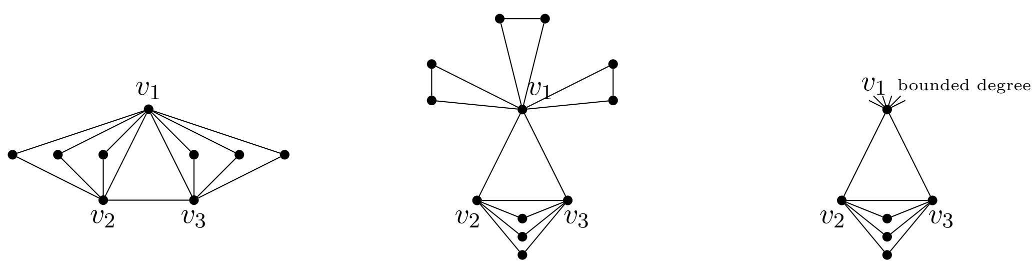
Proof. Let s be any hyperedge in the hypergraph. Since we assume that there are no thick pairs, there are at most 30×3 hyperedges in $(V, \mathcal{S})$ that intersect s in exactly 2 vertices. We denote this set of at most 90 hyperedges by F_1. Let $v \in s$, and consider the hyperedges in $(V, \mathcal{S} \backslash (F_1 \cup \{s\}))$ that contain vertex v. By Lemma 1 we can distinguish between the following two cases.

- Assume there exist 3 hyperedges s_1, s_2, s_3 in $(V, \mathcal{S} \backslash (F_1 \cup \{s\}))$ that pairwise intersect at vertex v only. In this case, we denote by F_2 the set of at most 11 hyperedges consisting of s_1, s_2, s_3 and all the hyperedges in $\mathcal{S}$ that have exactly one vertex in each of $s_1 \backslash \{v\}$, $s_2 \backslash \{v\}$ and $s_3 \backslash \{v\}$. According to Observation 1 we can determine whether a given hyperedge $s' \in \mathcal{S} \backslash (F_1 \cup F_2 \cup \{s\})$ contains v or not by looking at the intersection of s' with s_1, s_2, s_3.

- Suppose now that there exists a set of vertices $\{v_1, v_2, v_3, v_4\}$ such that every hyperedge in $(V, \mathcal{S} \backslash (F_1 \cup \{s\}))$ that contains v also contains at least one of v_1, v_2, v_3 or v_4. In this case, we denote by F_2 the set of all the hyperedges that contain at least one of the pairs $\{v, v_1\}$, $\{v, v_2\}$, $\{v, v_3\}$ or $\{v, v_4\}$. By our assumption on no thick pairs, the set F_2 contains at most $31 \times 4 = 124$ edges. Observe that no hyperedge $s' \in \mathcal{S} \backslash (F_1 \cup F_2 \cup \{s\})$ intersects v.

By analogy with building the set F_2 for the vertex v, we build two more sets F_3 and F_4 for the other two vertices contained in the hyperedge s, i.e., for the vertices in $s \backslash \{v\}$. Now it is easy to see that the set $F = F_1 \cup F_2 \cup F_3 \cup F_4$ allows us to determine whether a given hyperedge $s' \in S \backslash (F \cup \{s\})$ intersects s or not. Note that F has size at most $90 + 3 \times 124 = 462$. □

In our next result, we will show that a 3-uniform hypergraph with a thick pair contains one of the structures presented in Figure 6, which we call "fly", "windmill", and "broken windmill" (the hyperedges are represented by triangles).



(a) A "fly" (b) A "windmill" (c) A "broken windmill"

Figure 6: Substructures that appear in a 3-uniform hypergraph with a thick pair

To prove the result about these three structures, we need the following observation.

Observation 3. Let $(V, \mathcal{S})$ be a 3-uniform hypergraph and let $v \in V$ be a vertex that does not belong to any thick pair. Then one of the following holds:

- Either there are 3 hyperedges s_1, s_2, s_3 that pairwise intersect only at vertex v.

- Or vertex v is contained in at most 124 hyperedges of $(V, \mathcal{S})$.

Proof. From Lemma 1, it follows that either there are 3 hyperedges s_1, s_2, s_3 that pairwise intersect only at vertex v, or there are 4 vertices v_1, v_2, v_3 and v_4 such that every hyperedge that contains v also contains at least one of v_1, v_2, v_3 or v_4. Note that in the second case, since neither of $\{v, v_1\}, \{v, v_2\}, \{v, v_3\}$ and $\{v, v_4\}$ is thick, there are at most 31 hyperedges containing one of these pairs. Therefore, there are at most $31 \times 4 = 124$ hyperedges that contain v. This finishes the proof of the observation. $\square$

Lemma 3. *Let $(V, \mathcal{S})$ be a 3-uniform hypergraph that contains a thick pair. Then it contains one of the following:*

- *A "fly", which is a hyperedge $s = \{v_1, v_2, v_3\}$ together with hyperedges s_1, s_2, s_3, s_4, s_5, s_6 such that s_1, s_2, s_3 intersect s at $\{v_1, v_2\}$ and s_4, s_5, s_6 intersect s at $\{v_1, v_3\}$.*
- *A "windmill", which is a hyperedge $s = \{v_1, v_2, v_3\}$ together with hyperedges s_1, s_2, s_3, s_4, s_5, s_6 such that s_1, s_2, s_3 intersect s at $\{v_2, v_3\}$ and such that the pairwise intersection of s_4, s_5, s_6 is vertex v_1.*
- *A "broken windmill", which is a hyperedge $s = \{v_1, v_2, v_3\}$ together with hyperedges s_1, s_2, s_3 such that s_1, s_2, s_3 intersect s at $\{v_2, v_3\}$ and there are only at most 124 hyperedges in $(V, \mathcal{S}\backslash\{s\})$ that contain vertex v_1.*

Proof. Let us call a hyperedge $s = \{v_1, v_2, v_3\} \in \mathcal{S}$ thick if it contains a thick pair, i.e. if at least one of $\{v_1, v_2\}$, $\{v_2, v_3\}$ and $\{v_1, v_3\}$ is a thick pair. Let E denote the set of all thick pairs, $T \subseteq \mathcal{S}$ denote the set of all thick hyperedges, and $W \subseteq V$ denote the set of all vertices belonging to thick hyperedges.

First of all, we note that if there is a thick hyperedge $s = \{v_1, v_2, v_3\} \in \mathcal{S}$ containing two thick pairs $\{v_1, v_2\}$ and $\{v_1, v_3\}$, then we can choose three hyperedges, different from s containing $\{v_1, v_2\}$ and three hyperedges different from s containing $\{v_1, v_3\}$, and hence we obtain a "fly". We also note that each vertex of W belongs to some thick pair, or else a "windmill" or a "broken windmill" appears. Indeed, assume $s = \{v_1, v_2, v_3\}$ is a hyperedge with a thick pair $\{v_2, v_3\}$ and v_1 does not belong to any thick pair. Then, by Observation 3, either there are at most 124 hyperedges containing vertex v_1 in $(V, \mathcal{S}\backslash\{s\})$ or there are three hyperedges in $(V, \mathcal{S}\backslash\{s\})$ that pairwise intersect only at vertex v_1. Together with any three hyperedges that contain the vertices of the thick pair $\{v_2, v_3\}$ and are different from s, this gives us either a "windmill" or a "broken windmill".

Thus, from now onwards, let us assume that every vertex of W belongs to a thick pair and that each thick hyperedge contains exactly one thick pair, as otherwise we are done by the previous paragraph. As any thick pair belongs to at least 32 thick hyperedges, the inequality $|T| \geq 32|E|$ follows. Further, as each thick pair contains exactly two vertices of W and each vertex of W is contained in some thick pair, we have $|W| \leq 2|E|$. We conclude that $|T| \geq 32|E| \geq 16|W|$. Now, as each hyperedge $s \in T$ contains exactly one thick pair, we can define a function $f : T \to W$ sending $s \in T$ to $v \in W$ such that $s\backslash\{v\}$ is the thick pair contained in s. As $|T| \geq 16|W|$, by pigeonhole principle, there will be a vertex $v \in W$ such that $|f^{-1}(v)| \geq 16$. In other words, some vertex v forms a hyperedge in T with 16 different thick pairs $e_1, e_2, \ldots, e_{16}$, i.e., $\{v\} \cup e_i \in T$ for all $i \in 1, 2, \ldots, 16$.

We finish the proof by considering whether some four of these pairs $e_1, e_2, \ldots, e_{16}$ share a vertex in common or not. Suppose first that four of these pairs, say e_1, e_2, e_3, e_4, have a vertex w in common. Denote these pairs by $e_1 = \{w, z\}$, $e_2 = \{w, u_1\}$, $e_3 = \{w, u_2\}$ and $e_4 = \{w, u_3\}$. Then $s = \{v, w, z\}$, with $s_1 = \{v, w, u_1\}$, $s_2 = \{v, w, u_2\}$, $s_3 = \{v, w, u_3\}$, together with any three hyperedges that contain the thick pair $e_1 = \{w, z\}$ and are different from s, gives us a "fly". Alternatively, assume that no four pairs of $e_1, e_2, \ldots, e_{16}$ share a vertex in common. Then a graph G with the edges set $\{e_1, e_2, \ldots, e_{16}\}$ has degree at most 3 and hence it must contain a matching of size at least 4. Indeed, by observing that each edge of G is incident to at most 4 other edges, one can pick any edge and remove all incident edges repeatedly at least 4 times to obtain the required matching of size 4. Now, as each of these four edges is a thick pair and forms a hyperedge with v, we can easily see that a "windmill" appears. In both cases we obtain a desired structure, hence we are done. □

Corollary 1. Let $(V, \mathcal{S})$ be a 3-uniform hypergraph that contains a thick pair. Then there is a hyperedge $s \in \mathcal{S}$ and a set of hyperedges $F \subset \mathcal{S} \backslash \{s\}$ of size $|F| \leq 128$ such that for any $s' \in \mathcal{S} \backslash (F \cup \{s\})$, one can determine whether s' intersects s by looking at the intersections of s' with the hyperedges of F.

Proof. We use the notation of the statement of Lemma 3. Let s be a hyperedge given by Lemma 3 belonging either to a "fly" or to a "windmill" or to a "broken windmill".

If s belongs to a "fly", then we can take F to consist of 6 hyperedges $s_1, s_2, \ldots, s_6$ together with 2 further possible hyperedges on the wings of the fly (if the hypergraph contains it) on vertices $(s_1 \cup s_2 \cup s_3) \backslash \{v_1, v_2\}$ and $(s_4 \cup s_5 \cup s_6) \backslash \{v_1, v_3\}$. It now follows from Observation 2 that intersecting any hyperedge $s' \in S \backslash (F \cup s)$, with s_1, s_2 and s_3 one can determine whether s' contains either v_1 or v_2. Similarly, intersecting with s_4, s_5, s_6, determines whether s' contains either v_1 or v_3. Hence, by looking at the intersection of the edges of F with s' we can determine whether s' intersects s or not.

If s belongs to a "windmill", we can take F to consist of 6 hyperedges $s_1, s_2, \ldots, s_6$ together with one possible hyperedge on $(s_1 \cup s_2 \cup s_3) \backslash \{v_2, v_3\}$ and 8 possible hyperedges on $(s_4 \cup s_5 \cup s_6) \backslash \{v_1\}$ that have one vertex in each wing of the "windmill". By Observation 2 intersection of s' with s_1, s_2, s_3 determines whether s' contains either v_2 or v_3, while Observation 1 allows us to determine whether s' contains v_1 by looking at the intersection of s' with s_4, s_5 and s_6. Thus, again we can determine whether s' intersects s.

If s belongs to a "broken windmill", we can take F to consist of all the hyperedges that contain vertex v_1, of which there are at most 124, also with s_1, s_2, s_3 and one further possible hyperedge on the set $(s_1 \cup s_2 \cup s_3) \backslash \{v_2, v_3\}$. Now F contains at most 128 edges and it is clear by Observation 2 that this set determines whether s' intersects s or not. □

From Lemma 2 and Corollary 1 we deduce our main result of this section.

Theorem 8. *Intersection graphs of 3-uniform hypergraphs have functionality bounded by 462.*

3 Graphs of large functionality

Knowing what is good without knowing what is bad is just half-knowledge. Therefore, in this section we turn to graphs of large functionality.

When we talk about graphs of large functionality we assume that we deal with an infinite family X of graphs, because in any finite collection of graphs functionality is bounded by a constant. Moreover, we can further assume that X is hereditary. Indeed, if X is not hereditary, we can extend it to a hereditary class by adding all induced subgraphs of graphs in X, and this extension has (un)bounded functionality if and only if X has, because functionality is hereditary. Any hereditary class of graphs of bounded functionality has $2^{O(n \log_2 n)}$ labelled graphs with n vertices, since graphs of bounded functionality can be represented by binary words of length $O(n \log_2 n)$ [2]. In the terminology of [4], these are classes with (at most) factorial speed of growth, or simply (at most) factorial classes. Therefore, in every superfactorial class functionality is unbounded. This is the case, for instance, for bipartite, co-bipartite and split graphs, since each of these classes contains at least $2^{n^2/4}$ labelled graphs with n vertices. This fact allows us to establish a relationship between functionality and yet another important graph parameter known as VC-dimension, which is of use in statistical learning theory.

A set system (X, S) consists of a set X and a family S of subsets of X. A subset $A \subseteq X$ is *shattered* if for every subset $B \subseteq A$ there is a set $C \in S$ such that $B = A \cap C$. The VC-dimension of (X, S) is the cardinality of a largest shattered subset of X.

The VC-dimension of a graph $G = (V, E)$ was defined in [3] as the VC-dimension of the set system (V, S), where S the family of closed neighbourhoods of vertices of G, i.e., $S = \{N[v] \ : \ v \in V(G)\}$. We denote the VC-dimension of G by $vc(G)$.

Theorem 9. *There exists a function f such that for any graph G, $vc(G) \leq f(fun(G))$.*

Proof. Fix a k and consider the class X_k of all graphs of functionality at most k. Clearly, X_k is hereditary. Assume X_k contains graphs of arbitrarily large VC-dimension and let $G_1, G_2, \ldots$ be an infinite sequence of graphs from X_k with strictly increasing values of the VC-dimension. Let Y be the hereditary class containing all these graphs and all their induced subgraphs. Then Y is a hereditary subclass of X_k with unbounded VC-dimension. It is was shown in [14] that the only minimal hereditary classes of graph of unbounded VC-dimension are bipartite, co-bipartite and split graphs. But then Y and hence X_k contains one of these three classes, which is a contradiction to the fact that functionality is unbounded in these classes. Therefore, there is a constant $f(k)$ bounding the VC-dimension of graphs in X_k, which proves the result. $\square$

Since large VC-dimension implies large functionality, it would be natural to construct graphs of large functionality through constructing graphs of large VC-dimension. The latter is an easy task. Indeed, consider the bipartite graph $D_n = (A, B, E)$ with two parts $|A| = n$ and $|B| = 2^n$. For each subset $C \subseteq A$ we create a vertex in B whose neighbourhood coincide with C. Clearly, the VC-dimension of D_n is n and hence with n growing the functionality of D_n grows as well.

However, this example is not very interesting in the sense that D_n contains vertices of low functionality (of low degree) and hence graphs of large functionality are hidden in D_n as proper induced subgraphs. A much more interesting task is constructing graphs where *all* vertices have large functionality. In what follows, we show that this is the case for hypercubes.

Let $V_n = \{0,1\}^n$ be the set of binary sequences of length n and let $v, w \in V_n$. The Hamming distance $d(v,w)$ between v and w is the number of positions in which the two sequences differ. A *hypercube* Q_n is the graph with vertex set $V_n = \{0,1\}^n$, in which two vertices are adjacent if and only if the Hamming distance between them equals 1.

Theorem 10. *Functionality of the hypercube Q_n is at least $(n-1)/3$.*

Proof. By symmetry, it suffices to show that the vertex $v = 00\dots0 \in V_n$ has functionality at least $(n-1)/3$. Let v be a function of vertices in a set $S \subseteq V_n \backslash \{v\}$. To provide a lower bound on the size of S, and hence a lower bound on the functionality of v, for each $i = 1, 2, \dots, n$ consider the set $S_i = \{w \in S : d(w,v) = i\}$, i.e., the set of all binary sequences in S that contain exactly i 1s. Also, consider the following set:

$$I = \{i \in \{1,2,\dots,n\} : \exists z = z_1 z_2 \dots z_n \in S_1 \cup S_2 \cup S_3 \text{ with } z_i = 1\}.$$

Suppose $|I| \leq n-2$. Then there exist two positions i and j such that for any sequence $z = z_1 z_2 \dots z_n \in S_1 \cup S_2 \cup S_3$, we have $z_i = 0$ and $z_j = 0$. Consider the following two vertices:

- $u = u_1 u_2 \dots u_n$ with $u_k = 1$ if and only if $k = i$,
- $w = w_1 w_2 \dots w_n$ with $w_k = 1$ if and only if $k = i$ or $k = j$.

We claim that u and w are not adjacent to any vertex $z \in S$. First, it is not hard to see that for any $z \in S_1 \cup S_2 \cup S_3$ we have $d(z,u) \geq 2$ and $d(z,w) \geq 2$. Indeed, any $z \in S_1 \cup S_2 \cup S_3$ differs from u and w in position i, i.e., $z_i = 0$ and $u_i = w_i = 1$, and there must exist a $k \neq i, j$ with $z_k = 1$ and $u_k = w_k = 0$. Also, it is not difficult to see that $d(z,u) \geq 2$ and $d(z,w) \geq 2$ for any vertex $z \in S \backslash (S_1 \cup S_2 \cup S_3)$, because any such z has at least four 1s, while u and w have at most two 1s. Therefore, by definition, u and w are not adjacent to any vertex in S.

We see that the assumption that $|I| \leq n-2$ leads to the conclusion that there are two vertices $u, w \in Q_n \backslash (S \cup \{v\})$ which are non-adjacent to any vertex in S, but have different adjacencies to v. This contradicts the fact that v is a function of the vertices in S. So, we must conclude that I has size at least $n-1$. As each vertex in $S_1 \cup S_2 \cup S_3$ has at most three 1s, we conclude that $S_1 \cup S_2 \cup S_3$ must contain at least $|I|/3 = (n-1)/3$ vertices. This completes the proof of the theorem. □

Theorem 9 shows that graphs of bounded VC-dimension constitute an extension of graphs of bounded functionality, while Theorem 10 shows that this extension is proper, since the hereditary closure of hypercubes constitutes a proper subclass of bipartite graphs.

4 Concluding remarks and open problems

In this paper, we proved a number of results about graph functionality. However, many questions on this topic remain unanswered. Some of them are motivated by the results presented in the paper, for instance:

Problem 1. Is functionality bounded for interval graphs and for intersection graphs of k-uniform hypergraphs for $k > 3$?

Many other questions are motivated by related research. Of particular interest is the notion of *implicit representation* [12]. Similarly to bounded functionality, any hereditary class that admits an implicit representation is at most factorial. However, the question whether all factorial classes admit implicit representations, also known as the *implicit graph representation conjecture*, is widely open. Note that for non-hereditary classes the conjecture is not valid (see, e.g., [15]). We ask whether there is any relationship between the two notions in the universe of hereditary classes.

Problem 2. Does implicit representation of graphs in a hereditary class imply bounded functionality in that class and/or vice versa?

One more open question is inspired by a result in [2] showing that if the family of prime (with respect to modular decomposition) graphs in a hereditary class X is factorial, then the entire class X is factorial.

Problem 3. Is it true that if prime (with respect to modular decomposition) graphs in a hereditary class X have bounded functionality, then all graphs in X have bounded functionality?

We note that a similar question for implicit representations is open too. One more question is motivated by the fact that graphs of bounded functionality generalise both graphs of bounded degree and graphs of bounded rank-width, where isomorphism can be tested in polynomial time [10].

Problem 4. Can isomorphism be tested in polynomial time for graphs of bounded functionality?

Finally, we observe that polynomial-time algorithms available for graphs of bounded clique-width cannot, in general, be extended to graphs of bounded functionality, because many NP-hard problems remain intractable for graphs of bounded vertex degree. However, bounded degree is frequently helpful for designing fixed-parameter tractable algorithms.

Problem 5. Identify NP-hard problems that are fixed-parameter tractable on graphs of bounded functionality.

References

[1] B. Alecu, A. Atminas, V. Lozin, Graph Functionality, *Lecture Notes in Computer Science*, 11789 (2019) 135–147.

[2] A. Atminas, A. Collins, V. Lozin, and V. Zamaraev, Implicit representations and factorial properties of graphs, *Discrete Mathematics*, 338 (2015) 164–179.

[3] N. Alon, G. Brightwell, H. Kierstead, A. Kostochka, P. Winkler, Dominating sets in k-majority tournaments. *J. Combin. Theory Ser. B* 96 (2006) 374–387.

[4] J. Balogh, B. Bollobás, D. Weinreich, The speed of hereditary properties of graphs, *J. Combin. Theory Ser. B* 79 (2000) 131–156.

[5] H.-J. Bandelt, H.M. Mulder, Distance-hereditary graphs, *J. Combin. Theory Ser. B* 41 (1986) 182–208.

[6] B. Courcelle, J. Engelfriet and G. Rozenberg, Handle-rewriting hypergraph grammars, *Journal of Computer and System Sciences*, 46 (1993) 218–270.

[7] B. Courcelle, J.A. Makowsky and U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory Comput. Syst.* 33 (2000) 125–150.

[8] B. Courcelle and S. Olariu, Upper bounds to the clique-width of a graph, *Discrete Applied Math.* 101 (2000) 77–114.

[9] M.C. Golumbic, U. Rotics, On the clique-width of some perfect graph classes. *Internat. J. Found. Comput. Sci.* 11 (2000), no. 3, 423–443.

[10] Martin Grohe, Pascal Schweitzer, Isomorphism Testing for Graphs of Bounded Rank Width. *FOCS* 2015: 1010–1029.

[11] F. Gurski, E. Wanke, Line graphs of bounded clique-width, *Discrete Mathematics*, 307 (2007) 2734–2754.

[12] S. Kannan, M. Naor, S. Rudich, Implicit representation of graphs. *STOC* 1988: 334–343.

[13] M. Lejeune, V. Lozin, I. Lozina, A. Ragab, S. Yacout, Recent advances in the theory and practice of Logical Analysis of Data, *European J. Oper. Res.* 275 (2019) 1–15.

[14] V. Lozin, Graph Parameters and Ramsey Theory, *Lecture Notes in Computer Science*, 10765 (2018) 185–194.

[15] J.P. Spinrad, Efficient graph representations. Fields Institute Monographs, 19. American Mathematical Society, Providence, RI, 2003. xiii+342 pp.