1
Computational Complexity of Graphs

Stasys Jukna[1]

Computational complexity of graphs is the smallest number of union and intersection operations required to generate them when starting from simplest sets of edges: stars or cliques. An intriguing aspect of this measure is its connection to circuit complexity of Boolean functions and, in particular, with the **P** versus **NP** question. We survey this connection as well as known bounds on the complexity of explicit graphs.

1) University of Frankfurt, Dept. of Mathematics and Comput. Sci., Frankfurt a.M., Germany, and Vilnius University, Inst. of Mathematics and Informatics, Vilnius, Lithuania. Research supported by the DFG grant SCHN ~~503/5-1~~.

Contents

1.1 Introduction

Complexity is one of the crucial scientific phenomena of our times. In this chapter we consider the complexity of graphs. Motivated by specific applications, the complexity of a graph has been measured in several different ways.

For example, the complexity of a graph has been defined to be the number of its spanning trees [6, 11, 16]. Motivated by applications in biology, chemistry, and sociology, different notions of *graph entropy* were used to measure their complexity; see [12] for a survey. Motivated by the complexity of computing eigenspace projections, the *linear complexity* of graphs was introduced in [37]; this is the smallest number of arithmetic operations required to compute Ax, where A is the adjacency matrix of the graph. Motivated by the circuit complexity of boolean functions, the *star complexity* of graphs was introduced in [41]; this is the smallest number of union and intersection operations required to generate the graph when starting from stars. In this chapter we will consider this last measure.

In *computational complexity*, the measure of "complexity" of an object is understood as the smallest number of "elementary operations" that is enough to produce a given object starting from some "simplest" objects, called *generators*. Such a sequence of operations is called a *circuit*.

That is, a *circuit* for an object a is just a sequence $a_1, \ldots, a_t$ of objects such that $a_t = a$ and each a_i is obtained by applying an elementary operation to some previously obtained objects and generators. The *size* of the circuit is the number t of objects in it. Every circuit for an object a can be viewed as "code" of a. The larger the circuit must be (the more operations are required to produce the object), the more "complex" the object a is.

In this chapter we are mainly interested in the computational complexity of graphs and corresponding to them boolean functions. One can define the computational complexity of an n-vertex graph by considering its adjacency relation. Namely, one can encode the vertices by binary strings of length $l = \log_2 n$, and view a graph as its *adjacency function*: this is a boolean function of $2l$ variables which, given the codes of two vertices, outputs 1 iff these vertices are adjacent. One can then define the complexity of a given graph as the smallest number of AND, OR and NOT operations required to compute its adjacency function starting from variables and their negations. But in view of difficulties with proving lower bounds for boolean functions, this is a "dead-end" approach: so far, no explicit boolean function of $2l$ variables requiring more than $10l$ operations is known.

A more promising approach, initiated by Pudlák, Rödl and Savický in [41], is to view the graphs as *sets* of their edges, and define the complexity of a graph as the smallest number of the union ($\cup$) and intersection ($\cap$) operations needed to obtain the graph starting from some simplest graphs. In this chapter we

mainly consider the case when one takes *stars* as simplest graphs. A star is a set of edges joining one vertex with all remaining vertices. This results in the *star complexity* of graphs.

Of course, one may take other sets of "simplest" graphs as generators, like cliques, paths or matchings. The reason why we stick on stars as generators is that then the resulting measure for graphs is intimately related to the circuit complexity of boolean functions, and our main motivation is to prove lower bounds for boolean functions using graph complexity.

Counting arguments show that most of bipartite $n \times n$ graphs have star complexity about $n^2/\log n$. On the other hand, every *specific* graph of star complexity at least $5n$ would give us an specific boolean function requiring circuits of exponential size, and hence, resolve the fundamental problem of the entire computer science. Having found such a graph in **NP**, this would imply that $\mathbf{P} \neq \mathbf{NP}$. A graph belongs to **NP** if the adjacency in it can be decided by a nondeterministic Turing machine in time polynomial in $\log n$.

Actually, this is "bad news": we will not understand the star complexity of specific graphs until we resolve this widely open problem. Even worse, being "combinatorially complex" does not automatically imply that the graph is "computationally complex". In particular, there are combinatorially complex graphs—like Ramsey graphs—whose star complexity is small.

Still, "good news" is that we *are* able to prove non-trivial lower bounds on the star complexity of graphs in some restricted circuits models, like bounded-depth circuits with unbounded fanin gates. This already yields some new lower bounds for boolean functions, and opens alternative possibilities to approach some old problems in circuit complexity.

All in all, the star complexity of graphs is an interesting measure related to core problems of computer science. The goal of this survey is to motivate the reader to try to find graphs of large complexity.

1.2 Star complexity

We view graphs as *sets* of their edges. In what follows, $K_n = \binom{V}{2}$ denotes the set of all $\binom{n}{2}$ edges of a complete labeled graph on a fixed set V of $|V| = n$ vertices. By an n-vertex graph we will mean a subset $G \subseteq K_n$. Thus, $|G|$ will always denote the number of edges in G. A *star* around a vertex $v \in V$ is the set $S_v \subseteq K_n$ of all $n-1$ edges of K_n incident with v (Fig. 1.1).

Due to their direct connection with boolean functions, we will mainly consider *bipartite* graphs. A complete bipartite $n \times m$ graph is the set $K_{n,m} = L \times R$ of all nm edges, where $|L| = n$ and $|R| = m$; the sets L (left part) and R (right part) are sometimes called the *color classes*. A bipartite $n \times m$ graph is just

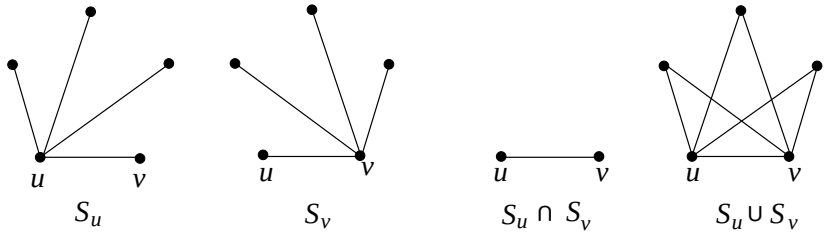


Fig. 1.1 Two stars S_u and S_v in K_5. Their intersection $S_u \cap S_v$ is just a single edge $\{u, v\}$, whereas their union $S_u \cup S_v$ is the complement of a complete subgraph K_3.

a subset $G \subseteq K_{n,m}$ of edges. A *star* around a vertex $v \in V = L \cup R$ is the set of all edges of $K_{n,m}$ incident with v. The *bipartite complement* of a bipartite graph $G \subseteq L \times R$ is the bipartite graph $\overline{G} = (L \times R) \setminus G$ with the same color classes L and R. The *adjacency matrix* of G is the $|L| \times |R|$ 0-1 matrix $A = (a_{u,v})$ such that $a_{u,v} = 1$ if and only if $(u, v) \in G$.

Definition 1.1 The *star complexity*, $\mathrm{Star}(G)$, of a graph G is the smallest number of fanin-2 union ($\cup$) and intersection ($\cap$) operations which is enough to produce the graph G starting from stars.

That is, we consider circuits whose generators (inputs) are stars, and elementary operations (gates) are $\cup$ and $\cap$. If not stated otherwise, we will assume that all gates have fanin 2.

Remark 1.2 Since there is a 1-1 correspondence between (labeled) bipartite graphs and 0-1 matrices, we can define the star complexity of a 0-1 matrix A as well. In this case, the AND and OR operations for matrices are performed componentwise. A *star matrix* is a 0-1 matrix consisting of exactly one all-1 row or of exactly one all-1 column, and having zeros elsewhere. It is easy to see that $\mathrm{Star}(G)$ is the smallest number of AND and OR operations that are enough to produce the adjacency matrix of G starting from star matrices.

Instead of circuits with set-theoretic gates $\cup$ and $\cap$, it will be more convenient to consider the standard model of monotone boolean circuits with boolean OR ($\vee$) and AND ($\wedge$) gates; such a circuit is monotone because it does not have negation gates $\neg f = 1 - f$. For this purpose, we associate a boolean variable x_v to each vertex $v \in V$, and consider circuits $F(X)$ on the set $X = \{x_v \colon v \in V\}$ of these variables. We say that a circuit $F(X)$ *represents* a given graph $G \subseteq K_n$ if for every two vertices $u \neq v$,

$$F(\mathrm{e}_u + \mathrm{e}_v) = 1 \text{ if and only if } u \text{ and } v \text{ are adjacent in } G; \tag{1.1}$$

here and throughout, $\mathrm{e}_u \in \{0,1\}^n$ is the unit vector of length n with exactly one 1 in the u-th position. If the graph $G \subseteq \binom{V}{2}$ is bipartite with a given bipartition $V = L \cup R$, then we only require that (1.1) holds for all $u \in L$ and $v \in R$.

It is easy to see that the smallest size of a monotone circuit representing a given graph is exactly the star complexity of that graphs:

$$\mathrm{Star}(G) = \text{minimum size of a monotone circuit representing } G.$$

This holds because: (i) stars are the only graphs represented by single variables (inputs of the circuit), and (ii) if two functions g and h represent graphs G and H, then $G \cap H$ is represented by $g \wedge h$, and $G \cup H$ is represented by $g \vee h$. Recall that all graphs are on the *same* set of vertices, that is, are subsets of pairs of vertices (edges) of the same fixed set of vertices; also, in the case of bipartite graphs, the bipartition is the same.

Remark 1.3 Note that the fact that a circuit represents a given graph only means that the circuit must behave correctly only on input vectors in $\{0,1\}^{|V|}$ with exactly two 1s—on the remaining input vectors the circuit can output arbitrary values!

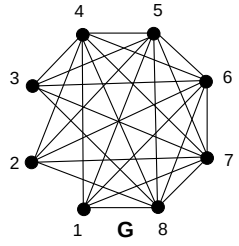
$$F(0,\ldots,0,\overset{u}{1},0,\ldots,0,\overset{v}{1},0,\ldots,0) = 1 \quad \text{if and only if} \quad \{u,v\} \in G\,.$$

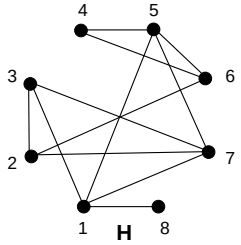
We will see that exactly this "freedom" makes the estimation of star complexity so difficult.

It will be sometimes more intuitive to view boolean functions and circuits as set-theoretic predicates $f : 2^{[n]} \to \{0,1\}$ accepting/rejecting *sets*: just identify every binary vector with the set of its 1-positions. In this set-theoretic setting, a circuit F represents a graph $G \subseteq K_n$ if F behaves correctly on 2-element sets $S = \{u,v\}$ (edges and non-edges): accepts such set if and only if u and v are adjacent in G. On sets S of size $|S| \neq 2$, the value $F(S)$ may be arbitrary!

Example 1.4. As mentioned above, in the case of non-bipartite graphs $G \subseteq \binom{V}{2}$, a circuit $F(x) = x_u$ consisting of single variable x_u represents the star $S_u = \{\{u,v\} \colon v \in V \setminus \{u\}\}$. A circuit $F(x) = \neg x_u$ consisting of a single negated variable represents the graph K_{n-1} obtained from K_n by removing all edges incident to u. An OR $F(x) = \vee_{u \in U} x_u$ of variables represents a union $\cup_{u \in U} S_u$ of stars, that is, the complement of the complete subgraph of K_n induced by $V \setminus U$. The AND $F(x) = x_u \wedge x_v$ represents the intersection $S_u \cap S_v$ of stars, that is, the graph consisting of just one edge $\{u,v\}$.

Example 1.5. Which of the following two graphs has large star complexity?





The graph G (on the left) is a complete graph K_8 with three edges of the triangle $\{1,2,3\}$ removed. One can verify that this graph is represented by the formula

$$F(x) = \Big(\bigvee_{v \not\in \{1,2\}} x_v \Big) \wedge \Big(\bigvee_{v \not\in \{1,3\}} x_v \Big) \wedge \Big(\bigvee_{v \not\in \{2,3\}} x_v \Big) .$$

To see this, take arbitrary two vertices u and v. First suppose that these vertices are adjacent in G. Then $\{u,v\} \not\subseteq \{1,2,3\}$. In this case the vector $\mathrm{e}_u + \mathrm{e}_v$ has at least one 1 in some position between 4 and 8. Thus, this vector must be accepted by all three ORs, implying that $F(\mathrm{e}_u + \mathrm{e}_v) = 1$. Now suppose that u and v are not adjacent in G. Then $\{u,v\} \subseteq \{1,2,3\}$, say, $u = 1$ and $v = 2$. In this case the vector $\mathrm{e}_u + \mathrm{e}_v$ has the form $(1,1,0,\ldots,0)$, and must be therefore rejected by the first OR. So, $F(\mathrm{e}_u + \mathrm{e}_v) = 0$ if u and v are not adjacent, as desired. Thus, the graph G (on the left) can be represented using just two fanin-2 AND gates and three large fanin OR gates. On the other hand, the graph H (on the right) does not seem to be represented with such a small number of gates.

Example 1.6. We give some examples in the case of bipartite graphs $G \subseteq L \times R$. Then a circuit $F(x) = x_w$ consisting of a single variable x_w for $w \in L \cup R$ represents the star $\{w\} \times R$ if $w \in L$, and the star $L \times \{w\}$ if $w \in R$ (see Fig. 1.2). An OR $F(x) = \vee_{w \in A \cup B} x_w$ with $A \subseteq L$ and $B \subseteq R$ represents the union $(A \times R) \cup (L \times B)$ of two bicliques (bipartite complete graphs), that is, the bipartite complement of the biclique $\overline{A} \times \overline{B}$. So, an AND of ORs represents a graph G which is an intersection of complements of bicliques or, in other words, the bipartite complement $\overline{G}$ of the graph G itself is just a union of bicliques.

An XOR $F(x) = \oplus_{w \in A \cup B} x_w$ represents the union $(A \times \overline{B}) \cup (\overline{A} \times B)$ of two *vertex-disjoint* bicliques. What graphs are represented by ANDs of XORs? It is not difficult to verify that these are exactly the so-called *fat matchings*, that is, bipartite graphs consisting of vertex-disjoint bicliques (these bicliques need not to cover all vertices). This holds because the graph represented by an XOR gate is a fat matching (consisting of two bicliques), and intersection of two fat matchings is a gain a fat matching. Thus,

- single variable x_w = a star,
- OR of variables = union of stars = union of two blicliques,
- XOR of variables = union of two vertex-disjoint bicliques,
- AND of ORs = complement of a union of bicliques,
- AND of XORs = fat matching.

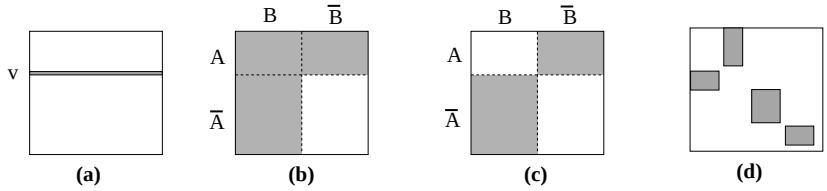


Fig. 1.2 The adjacency matrices of bipartite graphs represented by: (a) a single variable x_v, (b) an OR gate $\bigvee_{v\in A\cup B} x_v$, and (c) an XOR gate $g = \bigoplus_{v\in A\cup B} x_v$. The last matrix (d) is the adjacency matrix of a fat matching.

Star complexity of almost all graphs

It is easy to see that every bipartite $n \times n$ graph $G \subseteq L \times R$ can be represented by the monotone circuit

$$\bigvee_{(u,v)\in G} x_u \wedge x_v \quad \text{as well as by} \quad \bigvee_{u\in L} x_u \wedge \Big(\bigvee_{v\in N(u)} x_v \Big),$$

where $N(u)$ is the set of al neighbors of u in G. Since an OR of l variables can be computed using $l-1$ OR gates of fanin-2, the first circuit has $2|G| - 2$ fanin-2 gates, and the second circuit has $n - 1 + \sum_{u\in L} |N(u)| = |G| + n - 1$ fanin-2 gates. Thus, $\mathrm{Star}(G) = \mathcal{O}(n^2)$ for every $n \times n$ graph. It turns out that this trivial upper bound can be improved by a logarithmic factor.

Let $\mathrm{Star}(n)$ denote the maximum of $\mathrm{Star}(G)$ over all bipartite $n\times n$ graphs G.

Theorem 1.7 $\mathrm{Star}(n) = \Theta(n^2/\log n)$.

Proof. Lower bound. Let $\phi(n, M)$ denote the number of distinct boolean functions of n variables $x_1, \ldots, x_n$ that are computable by using at most M AND, OR and NOT gates. In particular, at most $\phi(n, M)$ distinct bipartite $n \times n$ graphs G can have $\mathrm{Star}(G) \leq M$. On the other hand, it is well known and easy to show (see, e.g., [22], Lemma 1.11) that $\phi(n, M) \leq (cM)^{M+n}$ for a constant c. Since we have 2^{n^2} bipartite $n \times n$ graphs, and each of them requires its own circuit to represent, the bound M on the star complexity of all graphs must satisfy the inequality $(cM)^{M+n} \geq 2^{n^2}$, from which the desired lower bound $M = \Omega(n^2/\log n)$ follows.

To prove the upper bound $\mathrm{Star}(n) = \mathcal{O}(n^2/\log n)$, we need the following result about biclique coverings of graphs. A *biclique covering* of a graph G is a collection of bicliques (complete bipartite subgraphs) of G such that each edge of G belongs to at least one of the bicliques. If each edge belongs to exactly one of the bicliques, then we have a *biclique decomposition* of G. The *weight* of a biclique is the number of vertices in it. The *weight* of a biclique covering (decomposition) of G is the sum of weight of all bicliques in that covering (decomposition).

Let $\mathrm{Cov}(G)$ denote the smallest weight of a biclique covering, and $\mathrm{Dec}(G)$ the smallest weight of a biclique decomposition of G. It is clear that $\mathrm{Cov}(G) \leq \mathrm{Dec}(G)$.

Lemma 1.8 (Lupanov [32]) *For every bipartite $n \times m$ graph G,*

$$\mathrm{Dec}(G) \leq \frac{2nm}{\log_2 n}\,.$$

Proof. Our goal is to prove the following claim: every $n \times m$ graph has a biclique decomposition of weight at most $n+m2^{m-1}$. Then we can decompose a given $n\times m$ graph G into m/k subgraphs of dimension $n\times k$. By our claim, each of these subgraphs has a biclique decomposition of weight at most $n + k2^{k-1}$, implying that the total weight of the biclique decomposition of G is at most $nm/k + m2^{k-1}$. The lemma then follows by taking $k = \lfloor \log_2 n - 2\log_2 \log_2 n \rfloor$.

To prove our claim, take an $n\times m$ graph G, and let $A = (a_{ij})$ be the adjacency matrix of G, that is, $a_{ij} = 1$ if and only if $(i, j) \in G$. Split the rows of A into groups, where the rows in one group all have the same values. This gives us a decomposition of G into $t \leq 2^m$ bicliques. For the i-th of these matrices, let r_i be the number of its nonzero rows, and c_i the number of its nonzero columns. Hence, $r_i + c_i$ is the weight of the i-th biclique in our decomposition. Since each nonzero row of A lies in exactly one of the these matrices, the total weight of the decomposition is

$$\sum_{i=1}^{t} r_i + \sum_{i=1}^{t} c_i \leq n + \sum_{j=0}^{n} \sum_{i:c_i=j} j \leq n + \sum_{j=0}^{m} \binom{m}{j} \cdot j = n + m2^{m-1}\,,$$

where the last equality is easy to prove: just count in two ways the number of pairs (x, S) with $x \in S \subseteq \{1, \ldots, m\}$. □

By Lemma 1.8, it remains to show that $\mathrm{Star}(G) \leq \mathrm{Cov}(G)$. For this, recall that a biclique $S\times T$ can be represented by a trivial circuit $(\vee_{u\in S} x_u)\wedge(\vee_{v\in T} x_v)$. This implies that $\mathrm{Star}(K_{s,t}) \leq s + t - 1$. Now let $G = K_{s_1,t_1} \cup \cdots \cup K_{s_r,t_r}$ be a biclique covering of G of weight $w = \sum_{i=1}^{r}(s_i + t_i)$. Then

$$\mathrm{Star}(G) \leq \sum_{i=1}^{r} \mathrm{Star}(K_{s_i,t_i}) \leq r + \sum_{i=1}^{r}(s_i + t_i - 1) = \sum_{i=1}^{r}(s_i + t_i) = w\,. \qquad \square$$

Actually, many "combinatorially interesting" graphs G have much smaller star complexity.

Example 1.9. (Kneser graphs D_n) The *Kneser graph* $KG_{l,k}$ ($l > 2k \geq 4$) has all k-element subsets v of $[l] = \{1, \ldots, l\}$ as vertices, and two vertices are adjacent iff the corresponding k-subsets are disjoint. These graphs were introduced by Lovász [31] in his famous proof of Kneser's conjecture [25] that whenever the

k-subsets of a $(2k+s)$-set are divided into $s+1$ classes, then two disjoint subsets end up in the same class.

For us of interest will be bipartite version of Kneser graphs. Let $n = 2^l$. The *bipartite Kneser graph* is a bipartite $n \times n$ graph $D_n \subseteq L \times R$ whose vertices u in each color class are subsets of $[l] = \{1, \ldots, l\}$, and two vertices u and v from different color classes are adjacent if and only if $u \cap v = \emptyset$. Since $\log_2 3 > 1.58$, the graph D_n has

$$|D_n| = \sum_{u \in L} d(u) = \sum_{u \in L} 2^{l-|u|} = \sum_{i=0}^{l} \binom{m}{i} 2^{l-i} = 3^l \geq n^{1.58}$$

edges. On the other hand, the following monotone boolean function

$$f(x) = \bigwedge_{i=1}^{l} \bigvee_{v \in S_i} x_v \tag{1.2}$$

where $S_i = \{w \subseteq [l] \colon i \notin w\}$, represents D_n. Indeed, two vertices $u \in L$ and $v \in R$ are non-adjacent in D_n iff $u \cap v \neq \emptyset$ iff there is an $i \in u \cap v$ iff $\{u, v\} \cap S_i = \emptyset$ for some i iff uv is rejected by some OR $\bigvee_{v \in S_i} x_v$. Thus, $\mathrm{Star}(D_n) \leq ln = n \log_2 n$.

In fact, one can show that $\mathrm{Star}(D_n) \leq 2n - \log_2 n$. This follows from the fact (Lemma 1.16 below) that, for every integer $1 \leq s \leq l$, every collection of l boolean sums (that is, ORs) of n variables can be simultaneously computed by a circuit consisting solely of at most $sn + s2^{l/s} - 2l - s$ fanin-2 OR gates. Since in our case $l = \log_2 n$, we can take $s = 1$, implying that $2n - 2l - 1$ fanin-2 OR gates are enough to compute all l ORs in (1.2). By adding $l - 1$ fanin-2 AND gates we obtain the desired circuit computing $f(x)$.

Example 1.10. (Sylvester graphs H_n) An *Hadamard matrix* of order n is an $n \times n$ matrix with entries ± 1 and with row vectors mutually orthogonal. A graph associated with an Hadamard matrix M (or just an Hadamard graph) of order n is a bipartite $n \times n$ graph where two vertices u and v are adjacent if and only if $M(u, v) = +1$.

A prominent example of an Hadamard graph is the *Sylvester graph* H_n. This is a bipartite $n \times n$ graph with $n = 2^l$ vertices on each part identified with subsets of $\{1, \ldots, l\}$; two vertices u and v are adjacent iff $|u \cap v|$ is odd. This graph H_n has about[2] n^2 edges, but it can be represented by the following boolean function

$$h(x) = \bigoplus_{i=1}^{l} \bigvee_{v \in S_i} x_v \tag{1.3}$$

2) We will often use terms "f is about g" instead of $f = \Theta(g)$, "f is at least about g" instead of $f = \Omega(g)$, and "f is at most about g" instead of $f = \mathcal{O}(g)$.

where $S_i = \{w \subseteq [l]\colon i \notin w\}$, and $x \oplus y$ stands for XOR $x + y$ mod 2. In [18] it is shown that the graph H_n contains a Ramsey $\sqrt{n} \times \sqrt{n}$ graph G as its induced subgraph; a graph is a *Ramsey graph* if neither the graph nor its complement contains a copy of $K_{t,t}$ for $t = \mathcal{O}(\log n)$. By setting to 0 all variables in (1.3) corresponding to vertices lying outside G, we obtain that some Ramsey graphs can be represented as an XOR of $l = \log_2 n$ complements of cliques. Thus, even such "combinatorially complicated" graphs, as Ramsey graphs, have very compact representations.

Star complexity and biclique coverings

We have shown in the proof of Theorem 1.7 that $\mathrm{Star}(G) \leq \mathrm{Cov}(G)$, where $\mathrm{Cov}(G)$ is the smallest weight of a biclique covering of G. So, a natural question is: how good $\mathrm{Cov}(G)$ approximates the star complexity? It turns out that for some $n \times n$ graphs, the fraction $\mathrm{Cov}(G)/\mathrm{Star}(G)$ may be large. This is not very surprising because biclique coverings correspond to star complexity of graphs in a very restricted circuit model where we want to represent a graph just as a union of bicliques.

For a graph G, let $\rho(G)$ denote the maximum of $ab/(a+b)$ over all pairs $a, b \geq 1$ of integers such that G contains a copy of a complete bipartite $a \times b$ subgraph.

Lemma 1.11 $\mathrm{Cov}(G) \geq |G|/\rho(G)$.

Proof. Let $G = \cup_{i=1}^{r} E_i$ with $E_i = S_i \times T_i$ be a bipartite clique covering of G of minimal weight. We know that $|E_i|/(|S_i| + |T_i|) \leq \rho(G)$. Hence, the weight of the covering is

$$\sum_{i=1}^{r}(|A_i| + |B_i|) = \sum_{i=1}^{e}\sum_{e \in E_i} \frac{|S_i| + |T_i|}{|E_i|} \geq \sum_{i=1}^{r}\sum_{e \in E_i} \frac{1}{\rho(G)} = \frac{|G|}{\rho(G)}\,. \qquad \square$$

Now consider the bipartite Kneser graph D_n defined in Example 1.9.

Theorem 1.12 $\mathrm{Cov}(D_n) \geq n^{0.08} \cdot \mathrm{Star}(D_n)$.

Proof. Let $n = 2^l$. We already know (see Example 1.9) that $|D_n| \geq n^{1.58}$. On the other hand, the graph D_n can contain a complete bipartite $a \times b$ subgraph $\emptyset \neq S \times T \subseteq D_n$ only if $a \leq 2^k$ and $b \leq 2^{l-k}$ for some $0 \leq k \leq l$, because then it must hold that $(\cup_{u \in S} u) \cap (\cup_{v \in T} v) = \emptyset$. Since

$$\min\{2^k, 2^{l-k}\colon 1 \leq k \leq l\} = 2^{l/2}\,,$$

we have that $\rho(D_n) \leq 2^l/2^{l/2} = 2^{l/2} = \sqrt{n}$. By Lemma 1.11, every biclique cover of D_n must have weight at least $|D_n|/\rho(G) \geq n^{1.08}$. Since $\mathrm{Star}(D_n) \leq 2n$ (see Example 1.9), we are done. $\square$

1.3 From graphs to boolean functions

As we already mentioned, our main motivation to consider the star complexity of graphs is the wish to prove new lower bounds for boolean functions. That is, we use graphs as "auxiliary" objects—objects of primary interest remain boolean functions.

One of the oldest fields dealing with the computational complexity—initiated more than 60 years ago by pioneering works of Shannon—is that of *boolean circuit complexity*. In this case, objects are boolean functions $f(x_1, \ldots, x_l)$, that is mappings $f : \{0,1\}^l \to \{0,1\}$. The class of elementary operations which can be used at the gates is called a *basis*. The *circuit complexity* of a given boolean function f is the smallest number of these elementary operations which is enough to compute f.

A circuit can also be viewed as a labeled directed graph without cycles (see Fig. 1.3). The sources (fanin-0 nodes) are labeled by generators. Each of the remaining nodes is called *gate* and performs some of the elementary operations on nodes that have direct wires to that gate. The *fanin* of a gate is the number of wires entering it. The boolean function computed by the circuit is defined in the obvious way. The *size* of a circuit is the total number of gates in the circuit. Another important measure is the *depth* of the circuit which is the length of the longest directed path in the graph.

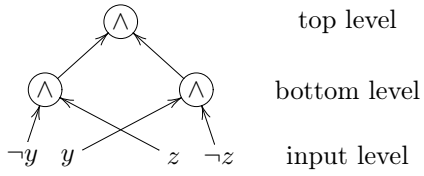


Fig. 1.3 A circuit F over the basis $\{\wedge, \vee, \neg\}$ of size 3 and depth 2 computing the XOR function: $F(y,z) = 1$ iff $y \neq z$. We will always assume that the NOT gates are only applied to the input variables, that is, inputs are literals (variables and their negations).

Easy counting shows that most boolean functions of l variables require circuits of size $2^l/l$: we have 2^{2^l} boolean function of l variables, but only about t^t circuits of size t. This was shown by Shannon more than 60 years ago. But despite of intensive research during the decades no *specific* function requiring "merely", say, $10l$ gates was found. Even in restricted circuit classes the progress is rather modest. Say, it remains open to prove a super-linear (in the number of variables) lower bound in the class of circuits of logarithmic depth.

The difficulty in proving that a given boolean function has high complexity lies in the nature of our adversary: the circuit. Small circuits may work in a

counterintuitive fashion, using deep, devious, and fiendishly clever ideas. How can one prove that there is no clever way to quickly compute the function? This is the main issue confronting complexity theorists.

One of the impediments in the lower bounds area is a shortage of problems of *intermediate* difficulty which lend insight into the harder problems. Most of known boolean functions are either "very simple" (parity, majority, etc.) or are "very complex": clique problem, satisfiability of CNFs, and all other **NP**-hard problems.

On the other hand, there are fields—like graph theory or matrix theory—with a much richer spectrum of known objects. It therefore makes sense to look more carefully at the graph structure of boolean functions: that is, to move from a "bit level" to a more global one and consider a given boolean function as a matrix or as a graph.

And indeed, it turns out that strong lower bounds on the complexity of bipartite graphs imply strong lower bound for circuit complexity boolean functions. Let, for example $\mathrm{Star}_3(G)$ be the smallest number s such that a bipartite $n \times n$ graph G can be written in the form

$$G = \bigcap_{i=1}^{s} \bigcup_{j=1}^{s} A_{ij} \times B_{ij} \,,$$

That is, we want to represent the graph as an intersection of at most s graphs, each of which is a union of at most s bipartite complete graphs. Since we have only $(2^{2n})^{s^2}$ such representations, and since every of 2^{n^2} graphs requires its own representation, we have that $(2^{2n})^{s^2} \geq 2^{n^2}$, from which $s \geq \sqrt{n/2}$ follows. In particular, almost all graphs G have $\mathrm{Star}_3(G) = \Omega(n^{1/2})$.

On the other hand, we will see (Problem 1.59 below) that any *explicit* sequence of graphs $(G_n \colon n = 1, 2, \ldots)$ with $\mathrm{Star}_3(G_n) = \Omega(n^{\epsilon})$ for an arbitrarily small constant $\epsilon > 0$ would resolve a 30 years old problem in circuit complexity: would give us a boolean function which cannot be computed by log-depth circuits of linear size.

We now show how the complexity of graphs is related to the circuit complexity of boolean functions. For simplicity of notation, we will consider bipartite graphs $G \subseteq K_{n,m} = L \times R$ where $n = |L|$ and $m = |R|$ are powers of 2:

$$n = 2^l \quad \text{and} \quad m = 2^r$$

for some integers $l, r \geq 1$. We can therefore identify vertices $u \in L$ with binary vectors $\vec{u} \in \{0,1\}^l$, and vertices $v \in R$ with binary vectors $\vec{v} \in \{0,1\}^r$.

Definition 1.13 (Adjacency function) The *adjacency function* of a graph $G \subseteq K_{n,m}$ is a boolean function f_G of $l + r$ variables such that for every $u \in L$ and $v \in R$, $f_G(\vec{u}, \vec{v}) = 1$ if and only if $(u, v) \in G$.

Thus, every bipartite $2^l \times 2^r$ graph gives us a boolean function f_G of $l + r$ variables, and every boolean function of $l+r$ variables is the adjacency function

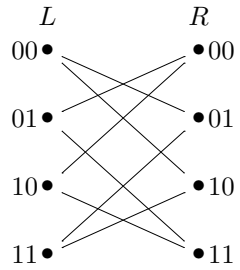


Fig. 1.4 A bipartite 4×4 graph G. Its adjacency function is the XOR function $f(y_1, y_2, z_1, z_2) = y_1 \oplus y_2 \oplus z_1 \oplus z_2$.

of some bipartite $2^l \times 2^r$ graph (see Fig. 1.4). But this trivial observation is not a big deal: we just used different terms for the same concept, the boolean function. The deal becomes more interesting when we ask the following question:

> How does the circuit complexity of f_G is related to the star complexity of the graph G?

The main relation between circuit complexity of boolean functions and the star complexity of graphs is given by the following lemma. In this lemma, under a circuit we understand any circuit whose inputs are literals (boolean variables and their negations); a circuit is *positive* if it has no negated variables as inputs.

Magnification Lemma *In any circuit computing f_G it is possible to replace each of its $2l + 2r$ input literals by an OR of new variables so that the resulting positive circuit represents G.*

Remark 1.14 Instead of replacing input literals by ORs one can also replace them by any other boolean functions that compute 0 on the all-0 vector, and compute 1 on any input vector with exactly one 1. In particular, one can take XORs instead of ORs of variables.

Proof. Let $G \subseteq L \times R$ be a bipartite $n \times m$ graph with $L = \{0,1\}^l$, $R = \{0,1\}^r$, and take a circuit $F(y, z)$ circuit computing its adjacency function $f_G : L \times R \to \{0,1\}$. That is, $F(u, v) = 1$ if and only if $(u, v) \in G$. The circuit F takes $2l + 2r$ input literals as inputs; we have $2l$ y-literals[3] y_i^a for $a = 0, 1$ and $i = 1, \ldots, m$, and $2r$ z-literals.

Let $X = \{x_u : u \in L \cup R\}$ be a set of new boolean variables, one for each vertex of G. We will show that it is possible to replace each y-literal by an

3) As usually, y_i^1 stands for the variable y_i itself, and y_i^0 stands for its negation $\neg y_i$.

OR of n new variables x_u with $u \in L$, and each z-literal by an OR of m new variables x_v with $v \in R$ such that the obtained positive circuit $F'(X)$ represents the graph G. Recall that a circuit represents G if for every $u \in L$ and $v \in R$, the circuit accepts the vector $e_u + e_v$ if and only if $(u, v) \in G$; here e_u is the vector in $\{0,1\}^{n+m}$ with exactly one 1 in the u-th position.

An input literal y_i^a with $a \in \{0,1\}$ in the circuit $F(y,z)$ accepts an input $(u,v) \in \{0,1\}^{l+r}$ if and only if $u(i) = a$ (the vector u has a in the i-th position). Hence, if we let $Y_i^a(X)$ to be the OR of all variables x_w such that $w \in L$ and $w(i) = a$, then

$$Y_i^a(e_u + e_v) = 1 \text{ iff } u(i) = a \text{ iff } y_i^a(u,v) = 1.$$

Similarly, if we let $Z_j^a(X)$ to be the OR of all variables x_w such that $w \in R$ and $w(j) = a$, then

$$Z_j^a(e_u + e_v) = 1 \text{ iff } v(j) = a \text{ iff } z_j^a(u,v) = 1.$$

Thus, the outputs of input literals y_i^a and z_j^a of the original circuit on the input $(u,v) \in \{0,1\}^{l+r}$ are the same as the outputs of the ORs Y_i^a and Z_j^a on the input $e_u + e_v \in \{0,1\}^{n+m}$. Since the rest of the new circuit F' is the same, we obtain that

$$F'(e_u + e_v) = 1 \text{ iff } F(u,v) = 1 \text{ iff } (u,v) \in G$$

implying that the new circuit F' represents the graph G, as desired. □

Remark 1.15 The Magnification Lemma is particularly appealing when dealing with circuit models allowing *unbounded* fanin OR (or unbounded fanin XOR) gates on the bottom, next to the input layer. In this case the total number of gates in the monotone circuit representing a graph G is just the same as in a non-monotone circuit computing f_G! That is, in such circuit models we have that

$$\text{circuit complexity of } f_G \;\geq\; \text{star complexity of } G. \tag{1.4}$$

Thus, if we could prove that some explicit bipartite $n \times n$ graph with $n = 2^l$ cannot be represented by such a circuit of size n^ϵ, then this would immediately imply that the corresponding boolean function $f_G(x,y)$ in $2l$ variables cannot be computed by a (non-monotone!) circuit of size $2^{\epsilon l}$, which is already exponential in the number $2l$ of variables of f. This is where the term "magnification" comes from:

> Small (linear) lower bounds on the star complexity of graphs yield large (exponential) lower bounds on the non-monotone circuit complexity of boolean functions.

Let us now consider the standard model of boolean circuits with *fanin*-2 AND and OR gates; inputs again are variables and their negations. This is the classical circuit model for which no super-linear lower bounds are known. For a boolean function f, let $\mathrm{Circuit}(f)$ denote the smallest number of gates in such a circuit computing f. Recall that $\mathrm{Star}(G)$ is the smallest number of fanin-2 AND and OR gates in a *monotone* circuit representing G; a circuit is monotone if it does not have negated variables as inputs. The question is: how $\mathrm{Circuit}(f_G)$ is related with $\mathrm{Star}(G)$?

Since now the gates have small fanin, the inequality (1.4) relating circuit complexity of boolean functions and graphs does not hold. In order to have at least some "approximate" inequality, we have to show how to compute the set of all $2(l+r) = 2\log_2 nm$ boolean sums (ORs) of variables using as few as possible fanin-2 OR gates, as given in the Magnification Lemma. If we compute all these sums separately, we will need $2l\log_2 n + 2r\log_2 m = 4n\log_2 n$ fanin-2 OR gates, if $m = n$. Using the so-called Transposition Principle, this trivial upper bound can be substantially improved to about $4n$.

Strong Magnification Lemma *For every bipartite $n \times m$ graph G,*

$$\mathrm{Circuit}(f_G) \geq \mathrm{Star}(G) - 2(n+m) - 8(\sqrt{n} + \sqrt{m})\,.$$

In particular, if $m = o(n)$ then $\mathrm{Circuit}(f_G) \geq \mathrm{Star}(G) - (2+o(1))n$. Recall that almost all graphs G have $\mathrm{Star}(G) = \Omega(n^2/\log n)$ (see Theorem 1.7).

Proof of the Strong Magnification Lemma

To prove that lemma, we have first to show how to simultaneously compute many boolean sums (ORs of variables) using relatively few fanin-2 OR gates. That is, we a given a collection

$$\bigvee_{j\in S_1} x_i, \ldots, \bigvee_{j\in S_m} x_i \tag{1.5}$$

of m boolean sums on the same set of n variables $x_1, \ldots, x_n$. We want to simultaneously compute these sums by a circuit consisting solely of fanin-2 OR gates. The smallest number of gates in such a circuit is the *disjunctive complexity* of the collection of sums.

We can specify each collection of boolean sums (1.5) by its *incidence matrix*: this is an $m \times n$ boolean matrix $A = (a_{ij})$, where $a_{ij} = 1$ if and only if $j \in S_i$. Then

$$\bigvee_{j\in S_i} x_i = \bigvee_{j:a_{ij}=1} x_j = \bigvee_{j=1}^{n} a_{ij}x_j\,.$$

Thus, computing the collection of boolean sums (1.5) means to compute a "linear transformation" $x \mapsto Ax$ over the boolean semiring. We are thus interested

in the smallest number $\mathrm{D}(A)$ of fanin-2 OR gates in a circuit computing the collection of boolean sums specified by the matrix A; in this case we that the circuit *computes* the matrix A.

We need the following useful fact relating the disjunctive complexity of a matrix A with the disjunctive complexity of the transposed matrix A^T; recall that the transpose of a matrix $A = (a_{ij})$ is the matrix $A^T = (b_{ij})$ with $b_{ij} = a_{ji}$.

The following fact was independently pointed out by Bordewijk [7] and Lupanov [32] in the context of rectifier networks.

Transposition Principle *If A is a boolean matrix with m rows and n columns, then $\mathrm{D}(A^T) = \mathrm{D}(A) + m - n$.*

Proof. Take a minimal circuit F with fanin-2 OR gates computing $y = Ax$, and let $\alpha(F)$ be the number of gates in it. We can view F as a rectifier $n \times m$ network (a directed acyclic graph) with n input and m output nodes "realizing" the matrix A in the following sense: there is a path from input node j to an output node i of F if and only if $a_{ij} = 1$. (We will investigate these networks further in Sect. 1.8.) If we reverse the direction of each wire in this network, the obtained network F^T will realize the transposed matrix A^T. Both networks F and F^T have the same number e of wires and the same number v of nodes (only the roles of input and output nodes is reversed). Moreover, since we had fanin-2 gates in the original circuit F, the number of OR gates in that circuit was $\alpha(F) = e - v + n$; this holds because $e = 2 \cdot \alpha(F)$ and $\alpha(F) = v - n =$ the number of non-input nodes. In the new $m \times n$ circuit F^T some OR gates may have fanin $d > 2$. In this case, we replace each such node by a binary tree of OR gates:

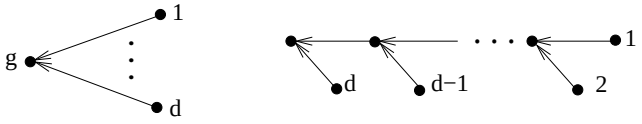


Fig. 1.5 We replace a node (an OR gate) g of fanin d by $d-1$ nodes each of fanin 2. In the former circuit we have $e - v = d - 1$, and in the latter $e' - v' = 2(d-1) - (d-1) = d - 1 = e - v$.

Thus, the difference $e' - v'$ between the numbers of wires and nodes in F^T does not exceed $e - v$, implying that the number of gates in the new circuit F^T is at most $e' - v' + m \leq e - v + m = \alpha(F) - n + m$. This shows the inequality $\mathrm{D}(A^T) \leq \mathrm{D}(A) + m - n$, and by symmetry, that $\mathrm{D}(A) \leq \mathrm{D}(A^T) + n - m$. □

Using the Transposition Principle, we can prove the following upper bound on the disjunctive complexity of any boolean matrix.

Lemma 1.16 (Lupanov [32]) *For every integer $1 \leq s \leq m$, every collection of m boolean sums of n variables can be simultaneously computed by a circuit consisting solely of at most $sn + s2^{m/s} - 2m - s$ fanin-2 OR gates.*

In particular, any collection of $m = s \log_2 n$ boolean sums in n variables can be simultaneously computed by a circuit consisting of at most $2sn$ fanin-2 OR gates.

Proof. Given a boolean $m \times n$ matrix A, we want to compute the set of m disjunctions of n variables defined by A. For this consider the transposed $n \times m$ matrix A^T. We can split A^T into into s submatrices, each of dimension $n \times k$ where $k \leq m/s$. By taking a circuit computing all possible disjunction of k variables, we can compute disjunctions in each of these submatrices using at most $2^k - k - 1$ OR gates. By adding $n(s-1)$ gates to combine the results of ORs computed on the rows of the submatrices, we obtain that

$$\mathrm{D}(A^T) \leq s2^k - m - s + n(s-1) \leq s2^{m/s} - m - s + n(s-1)$$

and, by the Transposition Principle,

$$\mathrm{D}(A) \leq \mathrm{D}(A^T) + n - m \leq sn + s2^{m/s} - 2m - s\,. \qquad \square$$

The *complement* of a boolean matrix $A = (a_{ij})$ is the matrix $\overline{A} = (\overline{a}_{ij})$ where $\overline{a}_{ij} = 1 - a_{ij}$. Let $\mathrm{D}(A, \overline{A})$ denote the minimum number of fanin-2 OR gates required to simultaneously compute the matrix A *and* its complement $\overline{A}$.

Lemma 1.17 *Let A be a boolean $p \times q$ matrix. Then $\mathrm{D}(A, \overline{A}) \leq q + 2^{p+2}$.*

Proof. The argument is similar to that in the proof of Lemma 1.8. Split the matrix A into $t \leq 2^p$ submatrices $A_1, \ldots, A_t$, each consisting of equal columns of A. Form a $p \times t$ matrix B by taking one column from each A_i. By taking $s = 1$ in Lemma 1.16, we obtain that $\mathrm{D}(B) \leq t + 2^p - 2p - 1 \leq 2^{p+1}$. Since the same argument applies also to $\overline{B}$, we obtain that both $\mathrm{D}(B)$ and $\mathrm{D}(\overline{B})$ are at most 2^{p+1}. Thus, there are circuits $F_1(z_1, \ldots, z_t)$ and $F_2(z_1, \ldots, z_t)$ computing Bz and $\overline{B}z$ such that both F_1 and F_2 have at most 2^{p+1} OR gates.

If $I_j \subseteq [n]$ is the set of indices of columns in A_j, then associate with submatrix A_j the sum $S_j = \vee_{i \in I_j} x_i$. Since the I_j are disjoint, all these t sums can be computed using at most $\sum_{j=1}^{t} (|I_j| - 1) = q - t < q$ OR gates. By taking the outputs of this circuit as inputs for F_1 and F_2, we obtain a circuit with at most $q + 2 \cdot 2^{p+1} = q + 2^{p+2}$ gates which computes both A and $\overline{A}$. $\square$

Proof of the Strong Magnification Lemma. In the Magnification Lemma we replace each of $l = \log_2 n$ y-variables by a boolean sum of $n = |L|$ new variables. Let A be the boolean $l \times n$ matrix corresponding to this set of boolean sums. The negations of y-variables are also replaced by boolean sums, and the corresponding matrix for these sums is just the complement $\overline{A}$ of A. Split the matrix A into two $(l/2) \times n$ submatrices A_1 and A_2. Applying Lemma 1.17 with $p = l/2 = (\log_2 n)/2$ and $q = n$, we obtain that

$$\mathrm{D}(A, \overline{A}) \leq \mathrm{D}(A_1, \overline{A}_1) + \mathrm{D}(A_2, \overline{A}_2) \leq 2(n + 4\sqrt{n})$$

fanin-2 OR gates are enough to compute all $2l$ boolean sums corresponding to the y-literals. Since the same argument yields a circuit with $2(m + 4\sqrt{m})$ fanin-2 OR gates computing the sums corresponding to the z-literals, the Strong Magnification Lemma is proved. □

Remark 1.18 The Strong Magnification Lemma can also be used to show that some graphs $G \subseteq K_{n,n}$ with $n = 2^l$ have *small* star complexity: for this it is enough to show that the adjacency function f_G can be computed by a small circuits over $\{\wedge, \vee, \neg\}$; recall that f_G has only $2l = 2\log_2 n$ variables. Since $\mathrm{Star}(G) \leq (4 + o(1))n + \mathrm{Circuit}(f_G)$, we have that $\mathrm{Star}(G) \leq (4 + o(1))n$ for all graphs G whose adjacency functions have circuits of *polynomial* in l size!

Towards the $(2 + c)n$ lower bound

We already known (Theorem 1.7) that bipartite $n \times m$ graphs G of star complexity $\mathrm{Star}(G) = \Omega(nm/\log n)$ exist; in fact, such are almost all graphs. On the other hand, the Strong Magnification Lemma implies that even a lower bound of $\mathrm{Star}(G) \geq (2 + c)n$ for an arbitrarily small constant $c > 0$ on the star complexity of an *explicit* $n \times m$ graph G with $m = o(n)$ would have great consequences in circuit complexity: such a graph would give an explicit boolean function f_G requiring circuit of exponential (in the number $\log_2 nm$ of variables) size! (Recall that, for boolean functions, even super-linear lower bounds are not known so far.) In particular, if the graph G is such that the adjacency of vertices in G can be determined by a nondeterministic Turing machine running in time polynomial in the binary length $\log_2 n$ of the codes of vertices, then a lower bound

$$\mathrm{Star}(G) \geq (2 + c)n$$

for an arbitrarily small constant $c > 0$ would imply that $\mathbf{P} \neq \mathbf{NP}$. Thus, star complexity of graphs captures one of the most fundamental problems of computer science.

On the other hand, the lower bound $\mathrm{Star}(G) \geq 2n - O(1)$ is achieved on relatively simple graphs. Say that a graph $G \subseteq K_{n,m} = L \times R$ has *distinct neighbors* if no vertex in L has degree 0 or m, and no two vertices in L have the same set of neighbors in R.

Theorem 1.19 (Chashkin [10]) *If a bipartite $n \times m$ graph G has distinct neighbors, then* $\mathrm{Star}(G) \geq 2n - 1$.

The proof of this theorem goes deeply in the structure of circuits representing the graphs, and is somewhat involved. We will therefore demonstrate the main ideas by giving a simpler proof for non-bipartite graphs.

Let $G_n = K_{n-1} + E_1$ be a complete graph on $n-1$ vertices plus one isolated vertex. We identify the vertices of G_n with boolean variables $x_1, \ldots, x_n$. Assume that the first $n-1$ variables form a cliques, and x_n is an isolated vertex.

Theorem 1.20 $\mathrm{Star}(G_n) \geq 2n - 6$.

Proof. The main property of the graph G_n we will use is that functions representing it are related to threshold functions. The *threshold-k* function of n variables is a monotone boolean function Th_k^n defined by:

$$\mathrm{Th}_k^n(x_1, \ldots, x_n) = 1 \text{ if and only if } x_1 + x_2 + \cdots + x_n \geq k\,.$$

Claim 1.21 *Let $f(x_1, \ldots, x_n)$ be a monotone boolean function representing G_n. Then $f(x_1, \ldots, x_{n-1}, 0) = \mathrm{Th}_2^{n-1}(x_1, \ldots, x_{n-1})$.*

Proof. Let $g(x_1, \ldots, x_{n-1}) := f(x_1, \ldots, x_{n-1}, 0)$. Let $\mathrm{e}_i \in \{0,1\}^n$ denote the i-th unit vector with exactly one 1 in the i-th position.

First, observe that $g(\mathrm{e}_i + \mathrm{e}_j) = 1$ for all $1 \leq i < j < n$ because vertices x_i and x_j are adjacent in G_n. Next, observe that $g(\mathrm{e}_i) = 0$ for all $i < n$. Indeed, if $g(\mathrm{e}_i) = 1$ for some $i < n$, then $f(\mathrm{e}_i + \mathrm{e}_n) = 1$ because f is monotone. But this is a contradiction, because vertices x_i and x_n are not adjacent in G_n, implying that $f(\mathrm{e}_i + \mathrm{e}_n) = 0$. We have thus shown that $g : \{0,1\}^{n-1} \to \{0,1\}$ accepts every input vector with at least two 1s, and rejects all vectors with fewer than two 1s, that is, $g = \mathrm{Th}_2^{n-1}$. □

Claim 1.22 *Even if all boolean functions in at most two variables are allowed as gates, the function Th_2^n requires at least $2n - 4$ gates.*

Proof. The proof is by induction on n. For $n = 2$ and $n = 3$ the bound is trivial. For the induction step, take an optimal circuit for Th_2^n, and suppose that the bottom-most gate g acts on variables x_i and x_j with $i \neq j$. This gate has the form $g = \varphi(x_i, x_j)$ for some $\varphi : \{0,1\}^2 \to \{0,1\}$. Notice that under the four possible settings of these two variables, the function Th_2^n has *three* different subfunctions Th_0^{n-2}, Th_1^{n-2} and Th_2^{n-2}. It follows that either x_i or x_j fans out to another gate h, for otherwise our circuit would have only *two* inequivalent sub-circuits under the settings of x_i and x_j. Why? Just because the gate $g = \varphi(x_i, x_j)$ can only take *two* values, 0 and 1.

Now suppose that it is x_j that fans out to h. Setting x_j to 0 eliminates the need of both gates g and h. The resulting circuit computes Th_2^{n-1}, and by induction, has at least $2(n-1) - 4$ gates. Adding the two eliminated gates to this bound shows that the original circuit has at least $2n - 4$ gates, as desired. □

To finish the proof of the theorem, let $F(x_1, \ldots, x_n)$ be a circuit (even non-monotone) representing the graph G_n. If we fix the last variable x_n to 0,

then Claim 1.21 implies that the resulting circuit F' computes Th_2^{n-1}. By Claim 1.22, this circuit (and hence, also the original circuit F) must gave at least $2(n-1)-4 = 2n-6$ gates, as desired. □

1.4 Formula complexity of graphs

As before, we consider circuits with fanin-2 AND and OR gates; inputs are literals (variables and their negation). A circuit is *monotone* if it has no negated variables as inputs. Such a circuit is a *formula* if all its gates have fanout 1, that is, if the underlying graph of the circuit is a binary tree. By a *leafsize* of a formula we will mean the number of leaves in its underlying tree, that is, the number of occurrences of input literals. Since each gate has fanin 2, this number is equal two times the total number of gates.

There are some super-linear lower bounds on the leafsize computing explicitly defined boolean functions of l variables. A lower bound $l^{3/2}$ for XOR function was first proved by Subbotovskaya [44]. A quadratic lower bound l^2 for XOR was then proved by Khrapchenko [24]. Using more complicated boolean functions, this lower bound was improved to $l^{5/2}$ by Andreev [4], and further improved to $l^{3-o(1)}$ by Håstad [17]. But no explicit sequence of boolean functions is known which needs formulas larger than l^3. This is in a big contrast with *monotone* formulas: here even exponential in n lower bounds are known (see, e.g., the book [22]), and even for circuits, not only for formulas!

On the other hand, the Magnification Lemma relates the formula size of boolean functions to the star complexity of graphs as follows. For a boolean function f, let $L(f)$ denote the smallest leafsize of a formula computing f. For a graph G, let $L_+(G)$ denote the smallest leafsize of a *monotone* formula representing G. That is, $L_+(G)$ is the star complexity of G in the class of formulas. By Lemma 1.7, we know that $L_+(G) = \mathcal{O}(n^2/\log n)$ for every bipartite $n \times n$ graph, and graphs G with $L_+(G) = \Omega(n^2/\log n)$ exist.

The Magnification Lemma immediately yields that for every bipartite $n \times n$ graph G,

$$L(f_G) \geq \frac{2}{n} \cdot L_+(G)\,. \tag{1.6}$$

If $n = 2^l$, then the adjacency function f_G is a boolean function in $2l = 2\log_2 n$ variables. Thus, any explicit graph G with $L_+(G) \geq n\log^K n$ gives us an explicit boolean function $f = f_G$ of $2l$ variables such that $L(f) = \Omega(l^K)$. Recall that, so far, the strongest known lower bound has the form $L(f) = \Omega(l^3)$.

The star complexity of graphs deals with *monotone* circuits and formulas, and for such circuits even exponential lower bounds are known (see, e.g., the book [22]). So, why we cannot apply these arguments to lower-bound $\mathrm{Star}(G)$

or $L_+(G)$? By the definition of star complexity, we have that $L_+(G)$ is the minimum of $L_+(h)$ over all monotone boolean functions $h(x_1, \ldots, x_n)$ representing G:

$$L_+(G) = \min\{L_+(h) \colon h \text{ is monotone and represents } G\}\,.$$

Thus, even though we only need to consider *monotone* formulas, the difficulty is that we have to prove that *none* of boolean functions representing G has a small formula.

A standard monotone boolean function representing a graph $G = ([n], E)$ is the *quadratic function* of G defined by:

$$h_G(x_1, \ldots, x_n) = \bigvee_{\{i,j\} \in E} x_i \wedge x_j\,. \tag{1.7}$$

As before, we can view boolean functions $h(x_1, \ldots, x_n)$ as set-theoretic functions $h : 2^{[n]} \to \{0,1\}$: such a function accepts a set $S \subseteq [n] = \{1, \ldots, n\}$ if and only if it accepts the characteristic vector $\chi_S \in \{0,1\}^n$ with $\chi_S(i) = 1$ if and only if $i \in S$. Hence, the quadratic function of a graph G is the unique monotone boolean function $h_G : 2^{[n]} \to \{0,1\}$ such that, for every set of vertices $I \subseteq [n]$, we have that

$$h_G(I) = 0 \text{ if and only if } I \text{ is an independent set in } G. \tag{1.8}$$

Representation (1.7) shows that $L_+(h_G) \leq 2|E|$ holds for any graph $G = ([n], E)$, but for some graphs this trivial upper bound may be very far from the truth. Say, a complete bipartite $n \times n$ graph $K_{n,n} = L \times R$ has n^2 edges, but can be represented by a monotone formula $(\vee_{u \in L} x_u) \wedge (\vee_{v \in R} x_v)$ with $2n$ leaves.

Since, so far, we are unable to prove super-linear lower bounds for monotone formulas *representing* an explicit graph, a natural question is: what quadratic functions require monotone formulas of super-linear size to *compute* them? It turns out that such are dense graphs of girth > 4, that is, dense graphs without triangles and without 4-cycles. This can be proved using rank arguments.

The rank argument

Let $h : 2^{[n]} \to \{0,1\}$ be a boolean function. A matrix associated to h is an arbitrary $|h^{-1}(1)| \times |h^{-1}(0)|$ matrix A whose rows are labeled by subsets accepted by h, and columns by subsets rejected by h. Note that we do not put any restrictions on what the actual entries of A should be—one can define the entries in an arbitrary way. The goal is to choose A in such a way that the rank of A over some field is large, but the rank of every "legal" submatrix of A is small.

More precisely, say that a submatrix B of A is *legal* if there exists an $i \in [n]$ such that $i \in a$ and $i \notin b$ holds for all labels a of the rows of B, and all labels b of the columns of B. Note that if h is a monotone function ($a \subseteq b$ and $f(a) = 1$ implies $f(b) = 1$), then every single entry (a, b) of A is a legal submatrix, because $h(a) = 1$ and $h(b) = 0$ imply that $i \in a$ and $i \notin b$ must hold for at least one position i, because $a \not\subseteq b$. Let $\mathrm{rk}(A)$ denote the rank of A over GF(2). Then, for every matrix A associated with h,

$$L_+(h) \geq \frac{\mathrm{rk}(A)}{\max \mathrm{rk}(B)}\,, \tag{1.9}$$

where the maximum is over all legal submatrices B of A. The proof of this lower bound is based on a result of Khrapchenko [24] and Rychkov [43] that, if $L_+(h) = t$ then every matrix associated with h can be decomposed into t legal submatrices $B_1, \ldots, B_t$; this can be shown by an easy induction on t. By the subadditivity of rank, we then have

$$\mathrm{rk}(A) \leq \sum_{i=1}^{t} \mathrm{rk}(B_i) \leq t \cdot \max_i \mathrm{rk}(B_i)\,.$$

A lower bound for quadratic function

We will now use the rank argument to prove that quadratic functions of some graphs require monotone formulas of almost maximal leafsize. Recall that the quadratic function h_G (as defined in (1.7)) of every graph G with m edges can be computed by a monotone formula with at most $2m$ leaves. For graphs without 4-cycles, almost this number of leaves is also necessary.

Theorem 1.23 ([19]) *If $G = (V, E)$ is a triangle-free graph without 4-cycles, then*

$$L_+(h_G) \geq |E|\,.$$

Proof. We consider vertices as one-element and edges as two-element sets. Recall that $h_G : 2^V \to \{0, 1\}$ is a monotone boolean function accepting/rejecting subsets $I \subseteq V$ of vertices of G. Namely, $h_G(I) = 1$ if I contains a pair of two adjacent vertices (an edge), and $h_G(I) = 0$ if I is an independent set. We will concentrate on a special collection of independent sets defined by vertices and by edges as follows.

For a vertex $y \in V$, let I_y be the set of its neighbors. For an edge $y \in E$, let I_y be the set of all its *proper* neighbors; that is, $v \in I_y$ precisely when $v \notin y$ and v is adjacent with an endpoint of y. Let $\mathcal{I} = \{I_y \colon y \in V \cup E\}$. Since G has no triangles and no 4-cycles, the sets in $\mathcal{I}$ are independent sets, and must be rejected by h_G. We will concentrate on only these independent sets.

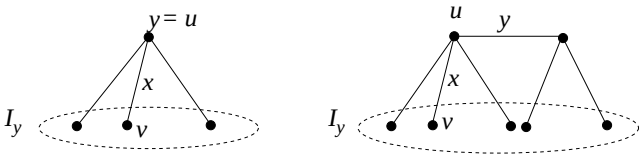


Fig. 1.6 The cases when $y \in V$ (left) and when $y \in E$ (right).

Let A be a submatrix of the matrix associated with h_G defined as follows. The rows are labeled by edges and columns by edges and vertices of G; a column labeled by y corresponds to the independent set I_y. The entries are defined by:

$$A[x,y] = \begin{cases} 1 & \text{if } x \cap y \neq \emptyset, \\ 0 & \text{if } x \cap y = \emptyset. \end{cases}$$

Claim 1.24 $\mathrm{rk}(A) = |E|$.

Proof. The matrix A has $|E|$ rows. We are going to show that A has full row-rank $|E|$ over GF(2). For this, take an arbitrary subset $\emptyset \neq F \subseteq E$ of edges. We have to show that the columns of the submatrix M' of M corresponding to the rows labeled by edges in F cannot sum up to the all-0 column over GF(2).

If F is not an even factor, that is, if the number of edges in F containing some vertex v is odd, then the column of v in M' has an odd number of 1s, and we are done.

So, we may assume that F *is* an even factor. Take an arbitrary edge $y = uv \in F$, and let $H \subseteq F$ be the set of edges in F incident to at least one endpoint of y. Since both vertices u and v have even degree (in F), the edge y has a nonempty intersection with an *odd* number of edges in F: one intersection with itself and an even number of intersections with the edges in $H \setminus \{y\}$. Thus, the y-th column of M' contains an odd number of 1s, as desired. □

By (1.9), it remains to prove the following claim.

Claim 1.25 *If B is a legal submatrix of A, then* $\mathrm{rk}(B) \leq 1$.

To prove this, let S be the set of all labels of rows, and T the set of all labels of columns of B. Since B is a legal submatrix of A, there must be a vertex $v \in V$ such that all edges $x \in S$ and all edges or vertices $y \in T$,

$$v \in x \text{ and } v \notin I_y \text{ for all } x \in S \text{ and } y \in T. \tag{1.10}$$

Thus, for each $y \in T$, we have two possible cases: either v is in y or not.

Case 1: $v \in y$. Since $v \in x$ for all $x \in S$, in this case we have that $x \cap y \supseteq \{v\} \neq \emptyset$, implying that $M_R[x,y] = 1$ for all $x \in S$. That is, in this case the y-th column of M_R is the all-1 column.

Case 2: $v \notin y$. We claim that in this case the y-th column of M_R must be the all-0 column. To show this, assume that $M_R[x, y] = 1$ for some edge $x \in S$. Then $x \cap y \neq \emptyset$, implying that x and y must share a common vertex $u \in x \cap y$ (see Fig. 1.6). Moreover, $u \neq v$ since $v \notin y$. Together with $v \in x$, this implies that $y = \{u, v\}$. But then $v \in I_y$, a contradiction with (1.10). □

Remark 1.26 Note that the lower bound $L_+(h) \geq |E|$ in Theorem 1.23 remains true for any monotone boolean function h such that (1.8) is only required to hold for every subset I of $|I| \leq 2d-1$ vertices, where d is the maximum degree of G. This is because then $|I_y| \leq 2d-1$ for every vertex or edge y.

Thus, to have a large lower bound on $L_+(h_G)$, we need that the graph G has many edges, and has no triangles and no copies of $K_{2,2}$. If the graph is bipartite, then the triangle-freeness condition is trivially fulfilled.

Construction 1.27 (Sum-product graph) Let p be a prime number and take a bipartite $n \times n$ graph with vertices in both its parts being pairs (a, b) of elements of a finite field $\mathbb{Z}_p$; hence, $n = p^2$. We define a graph G on these vertices, where (a, b) and (c, d) are joined by an edge if and only if $ac = b + d$ (all operations modulo p). For each vertex (a, b), its neighbors are all pairs $(x, ax - b)$ with $x \in \mathbb{Z}_p$. Thus, the graph is p-regular, and has $n = np = p^3 = n^{3/2}$ edges. Finally, the graph is $K_{2,2}$-free, because every system of two equations $ax = b+y$ and $cx = d + y$ has at most one solution (x, y). So, $L_+(h_G) = \Theta(n^{3/2})$.

Construction 1.28 (Point-line incidence graph) For a prime power q, a projective plane $PG(2, q)$ has $n = q^2 + q + 1$ points and n subsets of points (called lines). Every point lies in $q+1$ lines, every line has $q+1$ points, any two points lie on a unique line, and any two lines meet is the unique point. Now, if we put points on the left side and lines on the right, and joint a point x with a line L by an edge if and only if $x \in L$, then the resulting bipartite $n \times n$ graph G will have $(q + 1)n = \Theta(n^{3/2})$ edges and is $K_{2,2}$-free. So, we again have a matching lower bound $L_+(h_G) = \Theta(n^{3/2})$.

Thus, we can exhibit explicit graphs G whose quadratic functions require monotone formulas of leafsize $\Omega(n^{3/2})$. But the quadratic function h_G is just one of many possible boolean functions representing the graph G. Could we show that $L_+(h) = \Omega(n^{3/2})$ for *all* functions h representing G, then this would give us a lower bound $L_+(G) = \Omega(n^{3/2})$ on the star complexity of G, and by (1.6), a lower bound of $\Omega(\sqrt{n}) = \Omega(2^{l/2})$ on the non-monotone formula complexity of an explicit boolean function of $2l$ variables! Recall that the current "record" is a cubic lower bound $\Omega(l^3)$.

As noted above, besides the quadratic function h_G, there may be many other monotone boolean functions representing G—these functions may "wrongly" accept some independent sets of G of cardinality larger than two. On the other hand, there is a large class of graphs G for which h_G is the *only* monotone boolean function representing G.

Namely, call graph G *saturated* if it has no independent sets with more than two vertices, that is, if the complement of G is a triangle-free graph.

Proposition 1.29 *If $G = (V, E)$ is a saturated star-free graph, then h_G is the only monotone boolean function representing G.*

Proof. Let $h : 2^V \to \{0, 1\}$ be an arbitrary monotone boolean function representing G. We have to show that $h(S) = h_G(S)$ for all subsets $S \subseteq V$. If $h_G(S) = 1$ then S contains both endpoints of some edge. This edge must be accepted by h and, since h is monotone, $h(S) = 1$. If $h_G(S) = 0$ then S is an independent set of G, and $|S| \leq 2$ since G is saturated. Hence, S is either a single vertex or a non-edge. In the latter case we have that $h(S) = 0$ because h must reject all non-edges of G. If $S = \{v\}$, then we also have that $h(S) = 0$, because otherwise h would accept all edges of the star around the vertex v, contradicting the star-freeness of G. Thus, h must coincide with h_G, as desired. □

Unfortunately, so as it is, the argument in the proof of Theorem 1.23 does not work for saturated graphs.

1.5 Lower bounds via graph entropy

We now present a general argument allowing us to prove super-linear lower bounds on the leafsize of formulas representing graphs. Recall that a boolean function (or formula) $f(x)$ represents a graph $G \subseteq K_n$ if it behaves correctly of all inputs $\mathrm{e}_i + \mathrm{e}_j$ with exactly two 1s: $f(\mathrm{e}_i + \mathrm{e}_j) = 1$ if and only if i and j are adjacent in G. In particular, on inputs e_i with exactly one 1, the function can output arbitrary values. We say that f *strongly represents* G if we additionally have that $f(e_i) = 0$ for all $i = 1, \ldots, n$. Let $\ell_+(G)$ denote the smallest leafsize of a monotone formula strongly representing G.

To see the difference between this measure and the star complexity $L_+(G)$ of graphs in the class of formulas, let us consider the complete graph K_n. Since K_n is the union of n stars, this graph can be represented by the OR $x_1 \vee x_2 \vee \cdots \vee x_n$, implying that $L_+(K_n) \leq n$. In the case of *strong* representation, we have $\ell_+(K_n) \leq n\lceil \log_2 n \rceil$. For this, it is enough to write K_n as a union of $t \leq \lceil \log_2 n \rceil$ bipartite complete graphs $A_i \times B_i$ with $A_i \cap B_i = \emptyset$ and $|A_i| = |B_i| = n/2$. So, K_n can be strongly represented by a monotone formula

$$\bigvee_{i=1}^{t} \Big(\bigvee_{j \in A_i} x_j \Big) \wedge \Big(\bigvee_{k \in B_i} x_k \Big)$$

of leafsize at most tn. Below we will show that K_n has no better strong representation: $\ell_+(K_n) \geq n \log_2 n$. Although this lower bound is useless in

the framework of star complexity—after all we are looking for $n \cdot \text{poly}(\log n)$ lower bounds on $L_+(G)$—we still present the argument because it uses yet another interesting measure of graphs—their entropy—which can apparently be adopted to handle also star complexity.

Let μ be a measure which assigns to each graph $G \subseteq K_n$ a non-negative real number $\mu(G)$. Say that such a measure μ is a *good* graph-measure if

- $\mu(\emptyset) = 0$;
- μ is subadditive: $\mu(G \cup H) \leq \mu(G) + \mu(H)$;
- μ is monotone: $G \subseteq H$ implies $\mu(G) \leq \mu(H)$;
- μ respects bicliques: if G forms a complete bipartite graph on m (out of n) vertices, then $\mu(G) \leq m/n$.

Theorem 1.30 (Newman and Wigderson [36]) *For every graph G and for every good graph-measure μ,*

$$\ell_+(G) \geq n \cdot \mu(G)\,.$$

In fact, it is shown in [36] that a result of Krichevskii [28] implies the same lower bound for *non-monotone* formulas.

Proof. Let $f(x_1, \ldots, x_n)$ be a monotone boolean function. Then f can be written as an OR of monomials, where each monomial is an AND of variables. We concentrate on monomials of length 1 and 2. Monomials of length 2 define the graph $E_f \subseteq K_n$, where two vertices i and j are adjacent if and only if $x_i \wedge x_j$ is a monomial of f. We also let $V_f \subseteq [n]$ denote the set of vertices such that x_i is a monomial of f. Our goal is to prove that for every monotone boolean function f,

$$L_+(f) \geq n \cdot \mu(E_f) + |V_f|\,. \tag{1.11}$$

To see that this already implies the theorem, observe that f strongly represents a graph $G \subseteq K_n$ if and only if $E_f = G$ and $V_f = \emptyset$. Thus, every monotone formula strongly representing G must have $\geq n \cdot \mu(E_f) = n \cdot \mu(G)$ leaves, as claimed.

To prove (1.11), associate with every monotone boolean function f of n variables its *cost*

$$c(f) := \mu(E_f) + \frac{|V_f|}{n}\,.$$

If $f = x_i$ is a variable (a leaf of a formula), then $E_f = \emptyset$, $V_f = \{i\}$, and we get $c(x_i) = 1/n$. Moreover, the monotonicity of μ implies that the cost function is monotone with respect to inclusion: if $V_g \subseteq V_h$ and $E_g \subseteq E_h$, then $c(g) \leq c(f)$.

Claim 1.31 *$c(g \vee h) \le c(g) + c(h)$ and $c(g \wedge h) \le c(g) + c(h)$.*

Note that this claim already implies (1.11) since the cost of every leaf in a formula is $1/n$ and, by Claim 1.31, the cost of the output function does not exceed the sum of the costs of all the leaves. Thus $c(f) \le \frac{1}{n} \cdot L_+(f)$, implying that

$$L_+(f) \ge n \cdot c(f) \ge n \cdot \mu(E_f) + |V_f| \,.$$

So, it remains to prove the claim.

Case 1: $f = g \vee h$. Then $V_f = V_g \cup V_h$ and $E_f = E_g \cup E_h$. The subadditivity of μ yields

$$\begin{aligned} c(f) &= \mu(E_g \cup E_h) + \frac{|V_g \cup V_h|}{n} \\ &\le \mu(E_g) + \mu(E_h) + \frac{|V_g|}{n} + \frac{|V_h|}{n} = c(g) + c(h) \,. \end{aligned}$$

Case 2: $f = g \wedge h$. Denote $A = V_g$ and $B = V_h$. Since $V_f = A \cap B$ and

$$E_f = (E_g \cap E_h) \cup K_{A,B} \subseteq E_g \cup E_h \cup K_{A,B} \,,$$

where $K_{A,B} := (A \setminus B) \times (B \setminus A)$, we get:

$$\begin{aligned} c(f) &\le \mu(E_g \cup E_h \cup K_{A,B}) + \frac{|A \cap B|}{n} && \text{(monotonicity of } \mu) \\ &\le \mu(E_g) + \mu(E_h) + \mu(K_{A,B}) + \frac{|A \cap B|}{n} && \text{(subadditivity of } \mu) \\ &\le \mu(E_g) + \mu(E_h) + \frac{|A \setminus B| + |B \setminus A|}{n} + \frac{|A \cap B|}{n} && (\mu \text{ respects bicliques}) \\ &= \mu(E_g) + \mu(E_h) + \frac{|A|}{n} + \frac{|B|}{n} = c(g) + c(h) \,. \end{aligned}$$

This completes the proof of the claim, and thus the proof of the lemma. □

In order to use Theorem 1.30 we have to define some good measure of graphs. For this purpose, Newman and Wigderson (1995) used the measure of graph entropy introduced by Körner (1973).

Let G be a graph on $|V| = n$ vertices. The *graph entropy*, $\mathsf{E}(G)$, of G is the minimum

$$\mathsf{E}(G) = \frac{1}{n} \cdot \min_Y \sum_{v \in V} \log_2 \frac{1}{\mathrm{Prob}[v \in Y]} = -\frac{1}{n} \cdot \min_Y \sum_{v \in V} \log_2 \mathrm{Prob}[v \in Y]$$

taken over all (arbitrarily distributed) random variables Y ranging over independent sets in G. If $G = \emptyset$, then we set $\mathsf{E}(G) = 0$.

Lemma 1.32 *Graph entropy is a good measure.*

We have to show that the graph entropy is monotone, subadditive and respects bicliques.

Claim 1.33 (Monotonicity) *If $G \subseteq H$ are graphs on the same set of vertices, then $\mathsf{E}(G) \leq \mathsf{E}(H)$.*

Proof. Let Y be the random variable taking values in independent sets of H, which attains the minimum in the definition of the entropy $\mathsf{E}(H)$. Since an independent set in H is also an independent set in G, we have

$$\mathsf{E}(G) \leq -\frac{1}{n} \sum_{v \in V} \log_2 \mathrm{Prob}[v \in Y] = \mathsf{E}(H)\,. \quad \square$$

Claim 1.34 (Subadditivity) *If G and H are graphs on the same set of vertices, then $\mathsf{E}(G \cup H) \leq \mathsf{E}(G) + \mathsf{E}(H)$.*

Proof. Let Y_1, Y_2 be random variables taking values in independent sets of G and H, respectively, which attain the minimum in the definition of entropy. We can assume that Y_1, Y_2 are independent. Also note that $Y_1 \cap Y_2$ is a random variable taking values in independent sets of $G \cup H$. We therefore have

$$\begin{aligned}
\mathsf{E}(G) + \mathsf{E}(H) &= -\frac{1}{n} \sum_{v \in V} \log_2 \mathrm{Prob}[v \in Y_1] - \frac{1}{n} \sum_{v \in V} \log_2 \mathrm{Prob}[v \in Y_2] \\
&= -\frac{1}{n} \sum_{v \in V} \log_2 (\mathrm{Prob}[v \in Y_1] \cdot \mathrm{Prob}[v \in Y_2]) \\
&= -\frac{1}{n} \sum_{v \in V} \log_2 \mathrm{Prob}[v \in Y_1 \cap Y_2] \\
&\geq \mathsf{E}(G \cup H)\,. \quad \square
\end{aligned}$$

Claim 1.35 (Respecting bicliques) *If G is a bipartite graph with m (out of n) vertices, then $\mathsf{E}(G) \leq m/n$.*

Proof. Let $A, B \subseteq V$ be the parts of G; hence, $|A \cup B| = m$ and $|V| = n$. By the monotonicity, we can assume that G is a complete bipartite graph, $G = A \times B$. Define a random independent set Y by letting $\mathrm{Prob}[Y = A] = \mathrm{Prob}[Y = B] = 1/2$ and $\mathrm{Prob}[Y = C] = 0$ for all remaining independent sets.

Then

$$\begin{aligned}\mathsf{E}(G) &\le -\frac{1}{n}\sum_{v\in V}\log_2 \mathrm{Prob}[v\in Y]\\ &= -\frac{1}{n}\sum_{v\in A\cup B}\log_2 \mathrm{Prob}[v\in Y]\\ &= -\frac{1}{n}\sum_{v\in A\cup B} -1\\ &= \frac{|A\cup B|}{n} = \frac{m}{n}\,.\end{aligned}$$

This completes the proof of Claim 1.35, and thus of Lemma 1.32. □

Together with Theorem 1.30 we obtain the following general lower bound on the size of formulas strongly representing graphs.

Corollary 1.36 *For every graph G on n vertices, $\ell_+(G) \ge n \cdot \log_2 \mathsf{E}(G)$.*

In general, graph entropy of explicit graphs is not easy to compute. On the other hand, it can be lower-bounded in terms of the *independence number* $\alpha(G)$ of a graph G, that is, the maximum number of vertices in G no two of which are adjacent.

Theorem 1.37 *For every graph G on n vertices,*

$$\ell_+(G) \ge n \cdot \log_2 \frac{n}{\alpha(G)}\,.$$

Proof. By Corollary 1.36, it is enough to show that

$$\mathsf{E}(G) \ge \log_2 \frac{n}{\alpha(G)}\,.$$

Let Y be a random independent set in G which attains the minimum in the definition of the entropy $\mathsf{E}(G)$. For a vertex v, let $p_v := \mathrm{Prob}[v \in Y]$. Then $\sum_{v=1}^n p_v$ is the expected value of $|Y|$, and hence, cannot exceed $\alpha(G)$. On the other hand, since $\log_2 x$ is a concave function, we can apply Jensen's inequality and obtain

$$\begin{aligned}\mathsf{E}(G) &= -\sum_{v=1}^{n}\frac{1}{n}\log_2 p_v \ge -\log_2\Big(\sum_{v=1}^{n}\frac{1}{n}p_v\Big)\\ &\ge -\log_2\frac{\alpha(G)}{n} = \log_2\frac{n}{\alpha(G)}\,.\end{aligned}$$ □

In particular, for the complete graph K_n, we have that $\ell_+(K_n) \ge n\log_2 n$. As we have shown at the beginning of this section, this bound is almost tight. This is actually the bad news: using good graph-measures μ one cannot expect

to prove lower bounds larger than $\Omega(n \log n)$. The reason for this is the *monotonicity* condition of good graph-measures: one of the "simplest" graphs—the complete graph K_n—has the largest measure. It would be interesting to remove this condition.

Star complexity and affine dimension of graphs

Let W be a vector space of dimension d over some field $\mathbb{F}$. An *affine representation* of a graph G associates an affine space $S_v \subseteq W$ with every vertex v in such a way that two vertices u and v are adjacent in G iff $S_u \cap S_v \neq \emptyset$. The *affine dimension*, $\mathrm{adim}_{\mathbb{F}}(G)$, of G is the minimum d such that G has a d-dimensional affine representation.

A *partial matrix* over $\mathbb{F}$ is a usual matrix with the exception that some entries can be left empty (marked by $*$) without placing into them any elements of the underlying field $\mathbb{F}$. An *extension* of such a matrix is a fully defined matrix obtained by filling the unspecified entries by some elements of $\mathbb{F}$. The *rank* of a partial matrix is the minimum rank of its extension.

Given a bipartite graph $G \subseteq L \times R$, we can associate with it the following partial *edge-nonedge matrix* A_G whose rows correspond to edges x and columns to nonedges y of G. Fix any two elements $l \neq r$ of the field $\mathbb{F}$, and define the entries of A_G by:

$$A_G[x,y] = \begin{cases} l & \text{if } x \text{ and } y \text{ share a vertex in } L; \\ r & \text{if } x \text{ and } y \text{ share a vertex in } R; \\ * & \text{if } x \cap y = \emptyset. \end{cases}$$

Recall that if G is an $n \times n$ graph with $n = 2^l$, then its adjacency function f_G is a boolean function of $2l = 2\log_2 n$ variables.

Theorem 1.38 (Razborov [42]) *For every bipartite graph G,*

$$L(f_G) \geq \mathrm{rk}(A_G) \geq \mathrm{adim}_{\mathbb{F}}(G)\,.$$

Proof. The proof of the first inequality uses similar ideas as the proof of (1.9), and we omit it. We only prove the last inequality $\mathrm{rk}(A_G) \geq \mathrm{adim}_{\mathbb{F}}(G)$. Let A be an extension of the partial edge-nonedge matrix A_G such that $\mathrm{rk}(A) = \mathrm{rk}(A_G)$. Let a_x be the row of A corresponding to edge x of G. Assign to each vertex v of G an affine space S_v spanned by all rows a_x with $v \in x$, that is, S_v is the set of all affine combinations of these rows. If two vertices u and v are adjacent, then the spaces S_u and S_v contain the vector a_{uv}, and hence $S_u \cap S_v \neq \emptyset$.

Now suppose that u and v are not adjacent, and consider the y-th column of A, where $y = uv$. Since $v \in R$, all rows a_x with $v \in x$ must have r in the y-th position (in the partial matrix A_G, and hence also in its extension A), implying that their affine combination (with coefficients summing up to 1) must

also have r in that position. Thus, all vectors in S_v have r in the y-th position. But $u \in L$ implies that all vectors in S_u must have l in the y-th position. Thus, $S_u \cap S_v = \emptyset$. We have therefore constructed an affine representation of G of dimension $\mathrm{rk}(A)$. □

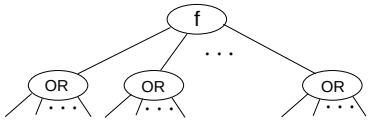
If the underlying field $\mathbb{F}$ has a finite number q of elements, then there are at most $\sum_{i=0}^{d} \binom{q^d}{i} \leq q^{d^2}$ possibilities to assign an affine space $S_v \subseteq \mathbb{F}^d$ of dimension $\leq d$ to each of the $2n$ vertices. Thus, there are at most $q^{2d^2 n}$ different affine representation. On the other hand, we have 2^{n^2} graphs in total. By comparing these bounds, we obtain that graphs G with $\mathrm{adim}_{\mathbb{F}}(G) = \Omega(\sqrt{n})$ exist. For every such graph we have that every non-monotone formula computing f_G must have $\Omega(\sqrt{n}) = \Omega(2^{l/2})$ leaves, which is exponential in the total number $2l$ of variables of f_G. Unfortunately, so far, no *explicit* graph of affine dimension larger than $\log_2 n$ is known.

1.6 Depth-2 complexity

The *lower bounds problem* for graphs (just as that for boolean functions) is to exhibit *specific* graphs of high star complexity. Results we mentioned above show that this is a very difficult problem: to prove $\mathbf{P} \neq \mathbf{NP}$, it is enough to exhibit an explicit bipartite $n \times n$ graph G such that, say, $\mathrm{Star}(G) \geq 4.0001n$ and the adjacency between any two vertices can be determined by a nondeterministic algorithm in time polynomial in $\log_2 n$.

Being unable to solve the lower bounds problem in its full generality, it is natural to try to understand the star complexity of graphs in restricted circuit models.

One of the "simplest" models is that of depth-2 formulas. Each such formula takes ORs of variables and applies some boolean function f to them:



More precisely, given a set $\mathcal{F}$ of boolean functions, a *depth-2 formula over the basis* $\mathcal{F}$ is a formula of the form

$$F(x) = f\Big(\bigvee_{w \in I_1} x_u, \ldots, \bigvee_{w \in I_r} x_u \Big), \tag{1.12}$$

where $f(y_1, \ldots, y_r)$ is some boolean function in $\mathcal{F}$. The *size* of such a circuit is the fanin r of the output gate f, and its *leafsize* if the total number $|I_1| + \cdots + |I_r|$ of occurrences of variables in it.

As before, we can view a boolean function f of r variables as a function $f : 2^{[r]} \to \{0,1\}$ accepting/rejecting *subsets* $S \subseteq [r] = \{1,\ldots,r\}$. This set-theoretic view at boolean functions gives us a bridge between depth-2 complexity of graphs and the well-studied subject of *intersection representations* of graphs.

Definition 1.39 (Intersection representation of graphs) Let $\mathcal{F}$ be some class of boolean functions. An *$\mathcal{F}$-intersection representation* of a bipartite graph $G \subseteq L \times R$ of *dimension* r is an assignment of (not necessarily distinct) subsets $S_w \subseteq [r]$ of positive integers (labels) to the vertices w for which there exists a boolean function $f \in \mathcal{F}$ of r variables such that for all vertices $u \in L$ and $v \in R$,

$(u,v) \in G$ if and only if $f(S_u \cap S_v) = 1$.

The *weight* of such a representation is the sum $\sum_w |S_w|$.

For a boolean function $f : 2^{[r]} \to \{0,1\}$, define its *complement*[4] to be the function$f^* : 2^{[r]} \to \{0,1\}$ defined by $f^*(S) = f(\overline{S})$, where $\overline{S} = [r] \setminus S$ is the complement of S. That is, the function f^* is obtained from f by negating all its variables. For example, the complement of AND function $x \wedge y$ is the negation of $x \vee y$: $(x \wedge y)^* = \overline{x} \wedge \overline{y} = \neg(x \vee y)$. For a class of boolean functions $\mathcal{F}$, let $\mathcal{F}^* = \{f^* : f \in \mathcal{F}\}$.

Proposition 1.40 (The bridge) *Let $\mathcal{F}$ be some class of boolean functions. A graph can be represented by a depth-2 circuit over $\mathcal{F}$ of size r and weight W if and only if the graph has an $\mathcal{F}^*$-intersection representation of dimension r and weight W.*

Proof. We only prove the "only if" direction (the "if" direction is similar). Suppose that some circuit $F(x)$ of the form (1.12) represents a graph G. Assigning to each vertex w the set $S_w = \{i : w \notin I_i\}$. Since $F(x)$ represents the graph G, we have that $(u,v) \in G$ if and only if $F(\mathrm{e}_u + \mathrm{e}_v) = 1$, which happens if and only if the top gate f accepts the set $\overline{S}_u \cup \overline{S}_v$ of indices i of those ORs $\bigvee_{e \in I_i} x_w$ that are "on" on input $\mathrm{e}_u + \mathrm{e}_v$. Thus,

$(u,v) \in G$ iff $f(\overline{S}_u \cup \overline{S}_v) = 1$ iff $f(\overline{S_u \cap S_v}) = 1$ iff $f^*(S_u \cap S_w) = 1$. □

Depth-2 with AND on the top

We first consider representation of graphs by depth-2 formulas of the form (1.12), where the top gate f is an AND gate. Such a formula has the form

$$F(x) = \bigwedge_{i=1}^{r} \Big(\bigvee_{v \in I_i} x_v \Big). \tag{1.13}$$

4) This should not be mixed with the *negation* $\neg f$ which is defined by $\neg f(S) = 1 - f(S)$.

Formulas of this form are usually called (monotone) CNFs (conjunctive normal forms). The *size* of such a formula is the number r of ORs in it. Let $\mathrm{cnf}(G)$ denote the smallest size a monotone CNF representing G. Note that for a bipartite graph $G \subseteq L \times R$, $\mathrm{cnf}(G)$ is just the smallest number r such that G can be written as an intersection

$$G = \bigcap_{i=1}^{r} \overline{A_i \times B_i} \tag{1.14}$$

of bipartite complements $\overline{A_i \times B_i} = (L \times R) \setminus (A_i \times B_i)$ of bicliques (bipartite complete graphs) $A_i \times B_i$, where $A_i = L \setminus I_i$ and $B_i = R \setminus I_i$. Equivalently, $\mathrm{cnf}(G)$ is just the smallest number r such that the bipartite complement of G can be written as a union

$$\overline{G} = \bigcup_{i=1}^{r} A_i \times B_i$$

of r bicliques. This implies that $\mathrm{cnf}(G) = \mathrm{bc}(\overline{G})$, where $\mathrm{bc}(H)$ is the *biclique covering number* of a graph H defined as the smallest number complete bipartite subgraphs of H such that each edge of H belongs to at least one of these subgraphs.

The measure $\mathrm{cnf}(G)$ is also tightly related to another combinatorial parameter of G–it's *disjointness dimension* $\theta(G)$. This is defined as the smallest number r for which it is possible to assign (not necessarily distinct) subsets of $[r]$ to vertices such that two vertices from different parts are adjacent in G if and only if their sets are disjoint. Since the complement of an AND function is the negation of an OR function, Proposition 1.40 implies that $\mathrm{cnf}(G) = \theta(G)$. Thus, we have the following equivalent definitions of $\mathrm{cnf}(G)$:

$$\mathrm{cnf}(G) = \theta(G) = \mathrm{bc}(\overline{G})\,. \tag{1.15}$$

These equivalences gives us a handy tool to prove bounds on the depth-2 complexity of graphs, then the top gate is an AND gate.

Proposition 1.41 *Every $n \times m$ graph has a CNF of size $\min\{n, m\}$, and graphs requiring CNFs of size at least $nm/(n+m)$ exits.*

Proof. Upper bound. Let $G \subseteq L \times R$ be a bipartite graph where $|L| = n$ and $|R| = m$. Associate with each vertex $u \in L$ the set $S_u = R \setminus N(u)$, where $N(u) \subseteq R$ is the set of all neighbors of u in G. Associate with each $v \in R$ the singleton set $S_v = \{v\}$. Then $S_u \cap S_v = \emptyset$ if and only if $v \in N(u)$, which happens if and only if $(u, v) \in G$. Thus, $\mathrm{cnf}(G) = \theta(G) \leq n$. The inequality $\mathrm{cnf}(G) \leq m$ is proved in the same way by interchanging the roles of L and R.

The lower bound follows by easy counting. We have at most $(2^r)^{n+m} = 2^{r(n+m)}$ possible encodings of $n + m$ vertices by subsets of $\{1, \ldots, r\}$. Hence,

at most $2^{r(n+m)}$ of all 2^{nm} bipartite $n \times m$ graphs can have depth-2 complexity $\leq r$. Thus, graphs requiring $r \geq nm/(n+m)$ exist. □

One can also easily exhibit explicit graphs of maximal CNF-complexity. Moreover, the complements of some graphs have exponentially larger complexity than the graphs themselves. To demonstrate this, let us consider the bipartite *n-matching* M_n. This is a bipartite $n \times n$ graph consisting of n vertex-disjoint edges.

Proposition 1.42 $\mathrm{cnf}(M_n) \leq \lceil \log_2 n \rceil$ *but* $\mathrm{cnf}(\overline{M}_n) = n$.

Proof. It is clear that $\mathrm{bc}(M_n) = n$: no two edges of M_n lie in one biclique. Thus, (1.15) immediately yields $\mathrm{cnf}(\overline{M}_n) = \mathrm{bc}(M_n) = n$. On the other hand, one can take the set of the first $\lceil \log_2 n \rceil$ natural numbers as labels, and assign to each vertex u on the left side its *own* subset S_u of labels, and assign the complement $S_v = \overline{S}_u$ of S_u to the unique vertex v on the right side matched by M_n. Then the sets S_u and S_v are disjoint if and only if $(u,v) \in M_n$, implying that $\mathrm{cnf}(M_n) = \theta(M_n) \leq \log_2 n$. □

We have just seen that some graphs of small degree (like matchings) have small CNFs. By slightly modifying the argument of Alon [1], it was shown in [20] that *all* graphs of small degree have small CNFs.

Lemma 1.43 ([1, 20]) *Every bipartite $n \times n$ graph of maximum degree $d \geq 1$ can be represented by a CNF of size at most $6d \ln n$.*

Proof. Let $H = \overline{G}$ be the bipartite complement of G. By (1.15), it is enough to show that the edges of H can be covered by about $d \ln |G|$ bicliques (bipartite complete subgraphs) of H.

To do this, we construct a biclique $S \times T \subseteq H$ via the following probabilistic procedure: pick every vertex $u \in U$ independently, with probability $p = 1/d$ to get a random subset $S \subseteq U$, and let be the set of all those vertices $v \in V$ that are adjacent in H to *all* vertices in S. It is clear that each so constructed complete bipartite graph $S \times T$ is a subgraph of H. Note that $(u,v) \in S \times T$ if (i) u was chosen in S, and (ii) none of (at most d) neighbors of u in $G = \overline{H}$ was chosen in S. Hence, this happens with probability at least $p(1-p)^d \geq p\mathrm{e}^{-pd} = p/\mathrm{e}$.

If we apply this procedure t times to get t complete bipartite subgraphs, then the probability that an edge (u,v) of H is covered by *none* of these subgraphs does not exceed $(1-p/\mathrm{e})^t \leq \mathrm{e}^{-tp/\mathrm{e}}$. Hence, the probability that some edge of H remains uncovered is smaller than $n^2\mathrm{e}^{-tp/\mathrm{e}t} = \exp(2\ln n - t/(\mathrm{e}d))$, which is smaller than 1 for $t = 2\mathrm{e}d \ln n$. □

By Proposition 1.42, already such simple graphs as the complement of an n-matching have maximal CNF complexity. By the Magnification Lemma, this implies that the boolean function $f(x,y)$ of $2l$ variables (with $l = \log_2 n$), defined by $f(x,y) = 1$ iff $x \neq y$, requires CNFs with at least 2^l clauses. Of

course, such a lower bound for CNFs is far from being interesting: it is easy to show that, say, the XOR of $2l$ variables needs even 2^{2l-1} clauses. Still, strong lower bounds on the CNF complexity of graphs could imply impressive lower bounds for boolean functions, if we could prove such bounds for graph *properties*.

Of particular interest is the following question: what monotone properties of graphs force their large CNFs? A property of graphs is *monotone* if it is preserved under deletion of edges. For example, the property of avoiding some fixed graph as a subgraph is a monotone property. It is conjectured that already $K_{2,2}$-freeness of graphs should force large depth-2 complexity. Namely, Pudlák, Rödl and Savický [41] conjectured that every bipartite $K_{2,2}$-free graph of average degree D requires CNFs of size $D^{\Omega(1)}$.

This conjecture was recently disproved by Katz [23] using probabilistic arguments: there exist $K_{2,2}$-free $n \times n$ graphs H of average degree D such that $\mathrm{cnf}(H) = \mathcal{O}(\log D)$. However, the average degree in these this graph is only about $n^{0.1}$. On the other hand, we already know $K_{2,2}$-free graphs whose minimum degree is about $n^{1/2}$; see Constrictions 1.27 and 1.28. So, let G stand for any of these graphs.

Open Problem 1.44 *Does there exist constants $\epsilon, \delta > 0$ such that $\mathrm{cnf}(H) \geq D^{\epsilon}$ holds for every subgraph H of G of average degree $D \geq n^{1/2-\delta}$?*

We will show in Section 1.7 that a positive solution to this problem would have several impressing consequences in circuit complexity theory.

Depth-2 with XOR on the top

We now consider the representation of graphs by depth-2 formula of the form (1.12), where the top gate f is an XOR gate (a sum modulo 2 of its inputs). Such a formula has the form

$$F(x) = \bigoplus_{i=1}^{r} \Big(\bigvee_{v \in I_i} x_v \Big). \tag{1.16}$$

Let $\mathrm{xor}(G)$ denote the smallest size (smallest fanin r of the top XOR gate) in such a formula representing G. Note that $\mathrm{xor}(G)$ is the smallest number r such that G can be written as a symmetric difference of r unions of stars.

For a bipartite G, let $\mathrm{rk}(G)$ denote the rank of the adjacency matrix of G over GF(2).

Proposition 1.45 *For every bipartite graph G, $|\mathrm{xor}(G) - \mathrm{rk}(G)| \leq 1$.*

Proof. The complement $\overline{z}_1 \oplus \overline{z}_2 \oplus \cdots \oplus \overline{z}_r$ of an XOR function $z_1 \oplus z_2 \oplus \cdots \oplus z_r$ is either the XOR itself (if r is even), or the negation of that XOR. Thus, by Proposition 1.40, $\mathrm{xor}(G) \leq r$ if and only if the adjacency matrix of G or of its complement $\overline{G}$ can be written as a matrix of scalar products over GF(2) of

vectors in $\{0,1\}^r$. Since the ranks of a boolean matrix and of its complement differ by at most 1, we are done. □

Thus, already such simple graphs as n-matching M_n (a bipartite $n \times n$ graph consisting of n vertex-disjoint edges) require large top fanin: $\mathrm{xor}(M_n) \geq n-1$. Since stars are really simplest graphs, one could expect that "combinatorially complicated" graphs should require large fanin as well. It turns, however, out that being "combinatorially complicated" does not necessarily imply large computationally complexity. To illustrate this, we now show that $\mathrm{xor}(G) = \mathcal{O}(\log n)$ for some Ramsey graphs. A bipartite graph is *t-Ramsey graph* if neither the graph nor its complement contains a complete bipartite $t \times t$ graph $K_{t,t}$.

Theorem 1.46 *Let n be a power of 2. There exist bipartite $n \times n$ graphs H such that H is t-Ramsey for $t = 2\log_2 n$ but $\mathrm{xor}(G) \leq t$.*

Proof. Let $n = 2^l$, and take the Sylvester $n^2 \times n^2$ graph H_{n^2}. Recall that vertices of this graph are vectors x in $\mathrm{GF}(2)^{2l}$, and two vertices x and y are adjacent in if and only if their scalar product over $\mathrm{GF}(2)$ is equal to 1. Thus,

$$\mathrm{xor}(H_{n^2}) \leq 2l = 2\log_2 n\,.$$

On the other hand, using probabilistic arguments, it can be shown that the graph H_{n^2} contains a bipartite $n \times n$ t-Ramsey graph H for $t = 2l = 2\log_2 n$ as an induced subgraph (see, e.g., Sect. 11.7 of [22]). Since H is an *induced* subgraph, we can obtain a circuit representing H from any circuit representing H_{n^2} by just setting to 0 all variables corresponding to vertices outside the graph H. Thus, $\mathrm{xor}(H) \leq \mathrm{xor}(H_{n^2}) \leq 2\log_2 n = t$. □

By Theorem 1.46, some strongly Ramsey $n \times n$ graphs can be represented as an XOR of only $2\log_2 n$ ORs. That is, some of such graphs are just a symmetric differences of only $2\log_2 n$ complements of bicliques. This is quite interesting because Ramsey graphs are very difficult to construct: best known constructions can only give t-Ramsey graphs for $t = n^\epsilon$, where $\epsilon > 0$ is arbitrary small, but constant.

Depth-2 with symmetric top gates

A *symmetric formula* of depth 2 is a formula of the form (1.12), where the output gate f is a symmetric boolean function, that is, a function whose output only depends on the number of 1s in the input vector. In set-theoretic terms, a boolean function $f : 2^{[r]} \to \{0,1\}$ is symmetric if for every set $S \subseteq [r]$, the value $f(S)$ only depends on the number $|S|$ of elements in S. Let $\mathrm{sym}(G)$ denote the smallest size, and $\mathrm{Sym}(G)$ the smallest weight of a symmetric depth-2 circuit representing G.

By Proposition 1.40, $\mathrm{sym}(G)$ is the smallest number of labels for which it is possible to assign each vertex w a subset S_w of labels so that

$$|S_u \cap S_v| \neq |S_x \cap S_y| \text{ for all } (u,v) \in G \text{ and } (x,y) \notin G.$$

Open Problem 1.47 *Exhibit an explicit bipartite $n \times n$ graph G such that $\mathrm{sym}(G) \geq 2^{(\ln \ln n)^{\alpha}}$ for some $\alpha(n) \to \infty$.*

By impressing results of Yao [48] and Beigel and Tarui [5], this would imply that the adjacency function f_G of G cannot be computed by an ACC circuit of polynomial size; see [22] for how does this happen. These are constant-depth circuits where, besides AND, OR and NOT gates, the *counting* gates can be used; a counting gate outputs 1 if and only if the number of 1s in input is divisible by some fixed number p. Exponential lower bounds for ACC circuits are only known when counting modulo a prime power p are allowed as gates. The case of composite moduli p remains open.

Actually, by the results of Green et al. (1995), it would be enough to prove such a lower bound on $\mathrm{sym}(G)$ as in Problem 1.47 for special depth-2 circuits where the top (output) gate f is the so-called "middle-bit" function: $f(S) = 1$ if and only if the middle bit of the binary representation of $|S|$ is 1.

By Proposition 1.41, $\mathrm{sym}(G) \leq \mathrm{cnf}(G) \leq n$ holds for all bipartite $n \times n$ graphs G. Moreover, easy counting shows that graphs with $\mathrm{sym}(G) \geq n/2$ exist. To see this, argue as in the proof of Proposition 1.41: there are at most $2^{r+1} \cdot (2^r)^{2n} = 2^{r+1+2rn}$ distinct symmetric intersection representations of dimension r. Thus, to represent all 2^{n^2} graphs, we need that $r \geq n/2$.

If all vertices in one color class have different sets of neighbors, then the sets of labels assigned to these vertices must be distinct in any intersection representation. Thus, $\mathrm{sym}(G) \geq \log_2 n$ for any such graph. Unfortunately, no stronger lower bounds for explicit graphs are known. Stronger lower bounds are only known under some restrictions of the form of sets S_w of labels associated with vertices.

Let us say that an intersection representation $w \mapsto S_w$ of a graph $G \subseteq L \times R$ is *balanced*, if exist two vertices $x, y \in L$ such that

$$|S_x \cap S_v \cap S_w| = |S_y \cap S_v \cap S_w| \text{ for all vertices } v \neq w \in R.$$

It is easy to see that every bipartite $n \times n$ graph G has a balanced intersection representation using n labels: assign to each vertex $x \in L$ the set $S_x := N(x) \subseteq R$ of its neighbors in G, and assign to each vertex $v \in R$ the single-element set $S_v = \{v\}$. This is clearly an intersection representation of G because $(x,v) \in G$ iff $v \in N(x) = S_x$ iff $|S_x \cap S_v| = 1$. Moreover, the representation is balanced because $S_v \cap S_w = \{v\} \cap \{w\} = \emptyset$ for all $v \neq w \in R$.

We now show that many graphs, including explicit ones, have large intersection dimension under any balanced representation.

Definition 1.48 (Isolated graphs) A bipartite graph $G \subseteq L \times R$ is *k-isolated* if for any two distinct vertices $x \neq y \in L$ there exists a subset $S \subseteq R$ of $|S| = k$ vertices such that every vertex $v \in S$ is adjacent to exactly one of the vertices x and y.

Recall that the *Sylvester graph* is a bipartite $n \times n$ graph $H = H_n$ with $n = 2^l$ whose vertices are vectors x in $\mathrm{GF}(2)^l$, and $(x, y) \in H$ if and only if $\langle x, y\rangle = 1$, where $\langle x, y\rangle = x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ly_l$ is the scalar product over $\mathrm{GF}(2)$.

Proposition 1.49 *Every non-zero vertex of the Sylvester $n \times n$ graph has exactly $n/2$ neighbors, and the graph is k-isolated for $k \geq n/2$.*

Proof. Let $H \subseteq L \times R$ be the Sylvester $n \times n$ graph with $L = R = \mathrm{GF}(2)^l$. We will use the following well-known property of the scalar product over $\mathrm{GF}(2)$.

Claim 1.50 *Every non-zero vector in $\mathrm{GF}(2)^l$ is orthogonal to exactly half of the vectors in $\mathrm{GF}(2)^l$.*

Proof. Take a vector $z \in \mathrm{GF}(2)^l$, $z \neq \vec{0}$. Then $z_i = 1$ for at least one position i. Hence we can partition the set $\mathrm{GF}(2)^l$ into $2^{l-1} = n/2$ pairs x, x' that differ only in their i-th position. For each of these pairs, we have that $\langle z, x\rangle \neq \langle z, x'\rangle$. Thus, $\langle z, x\rangle = 0$ for exactly half of vectors in $\mathrm{GF}(2)^l$. □

Claim 1.50 immediately implies the first claim of the lemma. To prove the second claim, fix an arbitrary pair of vectors $x \neq y \in L$. Since the vector $z = x \oplus y$ is a non-zero vector, Claim 1.50 gives us a set $S \subseteq V$ of $|S| = n/2$ vectors such that $\langle z, v\rangle = 1$ for all $v \in S$. Thus, every vector $v \in S$ is adjacent in H to exactly one of the vectors x and y. □

Theorem 1.51 ([20]) *If a bipartite graph G is k-isolated, then any balanced intersection representation of G must use at least k labels.*

In particular, bipartite $n \times n$ Sylvester graph H requires at least $n/2$ labels. On the other hand, by its definition, the graph H has an intersection representation of dimension $l = \log_2 n$, even relative to the XOR function. This shows that being balanced is a severe restriction on intersection representations.

Proof. Let $w \mapsto S_w$ be a balanced intersection representation of a graph $G \subseteq L \times R$ using r labels. Our goal is to show that $r \geq k$. Since the representation is balanced, there must exist two vertices $x \neq y \in L$ such that their sets of labels $X = S_x$ and $Y = S_y$ satisfy

$$|X \cap S_v \cap S_w| = |Y \cap S_v \cap S_w| \qquad \text{for all } v \neq w \in R. \tag{1.17}$$

On the other hand, since the graph is k-isolated, there must be a subset $V \subseteq R$ of $|V| = k$ vertices such that every vertex $v \in V$ is adjacent to exactly one of the vertices x and y. Hence, we must have that

$$|X \cap S_v| \neq |Y \cap S_v| \qquad \text{for all } v \in V. \tag{1.18}$$

Consider now the intersection matrix M of the set-system $\{S_v : v \in V\}$. That is, M is a $k \times k$ matrix with entries $M[v,w] = |S_v \cap S_w|$. For a linear multivariate polynomial $f : \mathbb{R}^r \to \mathbb{R}$, define the f-intersection version M_f of M by $M_f[v,w] = f(S_v \cap S_w)$.

Claim 1.52 *If f has N monomials, then* $\mathrm{rk}(M_f) \leq N$.

Proof. Let $f(z_1,\ldots,z_r) = \sum_{I\subset[n]} a_I \prod_{i\in I} z_i$ be a linear multivariate polynomial with $N = |\{I : a_I \neq 0\}|$ monomials. Each monomial of f accepts a set $A \cap B$ if and only if it accepts both A and B. Thus, the value $f(A \cap B)$ is just the scalar product of two vectors of length N, implying that $\mathrm{rk}(M_f) \leq N$. □

Consider now the following multilinear polynomial over the reals:

$$f(z_1,\ldots,z_r) = \sum_{i\in X} z_i - \sum_{i\in Y} z_i\,.$$

Note that for every subset $T \subseteq [r]$, the value $f(T)$ is just the difference between $|X \cap T|$ and $|Y \cap T|$. Hence, by taking $T = S_v \cap S_w$, (1.17) implies that $f(S_v \cap S_w) = 0$ for all $v \neq w \in V$, and (1.18) implies that $f(S_v \cap S_v) \neq 0$ for all $v \in V$. That is, the f-intersection matrix M_f of M is a real diagonal matrix with nonzero diagonal entries, implying that $\mathrm{rk}(M_f) = |V| = k$. On the other hand, polynomial f has $|X \cup Y| \leq r$ monomials. Claim 1.52 implies that $\mathrm{rk}(M_f) \leq r$, and the desired lower bound $r \geq k$ follows. □

Weight of symmetric depth-2 representations

We now consider the *weight* of symmetric depth-2 formulas representing graphs, that is, the total number of occurrences of variables in them. Recall that the weight of such a circuit

$$F(x) = f\Big(\bigvee_{u\in I_1} x_u, \ldots, \bigvee_{u\in I_r} x_u\Big)$$

for G is the sum $|I_1| + \cdots + |I_t|$. A circuit is symmetric, if f is a symmetric boolean function. Let $\mathrm{Sym}(G)$ denote the smallest weight of a symmetric depth-2 formula representing G.

Since $\mathrm{sym}(G) \leq n$ for all bipartite $n \times n$ graphs G (Proposition 1.41), we immediately obtain a trivial upper bound $\mathrm{Sym}(G) \leq 2n^2$. Using Lemma 1.8, we can get a somewhat better upper bound.

Proposition 1.53 *For every $n \times n$ graph G,* $\mathrm{Sym}(G) \leq 2n^2/\log_2 n$.

Proof. Lemma 1.8 gives us a decomposition $G = H_1 \cup \cdots \cup H_r$ of G into bicliques such that $\sum_{i=1}^r |V_i| \leq t := 2n^2/\log_2 n$, where V_i is the set of vertices of H_i. By assigning each vertex w the set $S_w = \{i \in [r] : w \in V_i\} \subseteq [r]$, we have that $(u,v) \in G$ if and only if $|S_u \cap S_v| \geq 1$. We thus obtained an intersection representation of G of weight $\sum_w |S_w| = \sum_{i=1}^r |V_i| \leq t$, as desired. □

Since $\mathrm{Sym}(G) \leq n \cdot \mathrm{sym}(G)$ trivially holds for every bipartite explicit $n \times n$ graph G, any explicit graph with $\mathrm{Sym}(G) \geq n2^{(\ln\ln n)^{\alpha}}$, for some $\alpha(n) \to \infty$, would resolve Problem 1.47. However, the best we can do so far is a lower bound of about $n2^{\ln\ln n}$.

Recall that a bipartite $n \times m$ graph $G \subseteq L \times R$ is *k-isolated* if for any two distinct vertices $x \neq y \in L$ there exists a subset $S \subseteq R$ of $|S| = k$ vertices such that every vertex $v \in S$ is adjacent to exactly one of the vertices x and y. A graph is *strongly isolated* if it is k-isolated for $k = \Omega(n)$, where $n = |L|$ is the number of vertices on the left side. In particular, the Sylvester $n \times n$ graph is k-isolated for $k \geq n/2$, and hence, is strongly isolated.

We have proved (Theorem 1.51) that every strongly isolated graph has almost maximal intersection dimension $\Omega(n)$, if only balanced intersection representations are allowed. Now we show that such graphs have large intersection weight *regardless* of what intersection representations are used. The main combinatorial tool we will use is the well-known *Sunflower Lemma* discovered by Erdős and Rado [14].

A *sunflower* is a family $F_1, \ldots, F_s$ of sets of the form $F_i = P_i \cup C$, where the P_i are pairwise disjoint; the set C is the *core* of the sunflower, and the P_i's are called the *petals*. In other words, each element belongs either to none, or to exactly one, or to *all* of the F_i. Note that a family of pairwise disjoint sets is a sunflower (with an empty core).

Sunflower Lemma *Every family of more that $l!(p-1)^l$ sets, each of which has cardinality at most l, contains a sunflower with p petals.*

Proof. Take a family $\mathcal{F}$ of $|\mathcal{F}| > l!(p-1)^l$ sets, each of cardinality at most l. We proceed by induction on l. For $l = 1$, we have more than $p-1$ points (disjoint 1-element sets), so any p of them form a sunflower with p petals (and an empty core). Now let $l \geq 2$, and take a maximal family $\mathcal{S} = \{S_1, \ldots, S_t\}$ of pairwise disjoint members of $\mathcal{F}$. If $t \geq p$, these sets form a sunflower with $t \geq p$ petals (and empty core), and we are done.

Now assume that $t \leq p-1$, and let $S = S_1 \cup \cdots \cup S_t$. Then $|S| \leq l(p-1)$. By the maximality of $\mathcal{S}$, the set S intersects every member of $\mathcal{F}$. By the pigeonhole principle, some point $x \in S$ must be contained in at least

$$\frac{|\mathcal{F}|}{|S|} > \frac{l!(p-1)^l}{l(p-1)} = (l-1)!(p-1)^{l-1}$$

members of $\mathcal{F}$. Let us delete x from these sets and consider the family

$$\mathcal{F}_x = \{F \setminus \{x\} \ : \ F \in \mathcal{F},\, x \in F\}\,.$$

Each member of $\mathcal{F}_x$ has at most $l-1$ elements and, by the choice of x, there are $|\mathcal{F}_x| > (l-1)!(p-1)^{l-1}$ sets in the family. By the induction hypothesis, $\mathcal{F}_x$ contains a sunflower with p petals. Adding x to the members of this sunflower, we get the desired sunflower in the original family $\mathcal{F}$. □

Theorem 1.54 ([20]) *Every k-isolated bipartite $n \times m$ graph G requires symmetric depth-2 formula of weight at least about $k \ln n / \ln \ln n$.*

Proof. Let $G \subseteq L \times R$ with $L = [n]$ and $R = [m]$ be a bipartite k-isolated $n \times m$ graph. Fix an arbitrary intersection representation $A_1, \ldots, A_n, B_1, \ldots, B_m$ of G. We may assume that $k > 0$ (since for $k = 0$ there is nothing to prove). Hence, all sets $A_1, \ldots, A_n$ must be distinct. Let

$$\ell := c\frac{\ln n}{\ln \ln n}$$

for a sufficiently small absolute constant $c > 0$. If $\sum_{i=1}^{n} |A_i| > n\ell$, then we are done. So, assume that $\sum_{i=1}^{n} |A_i| \leq n\ell$. Our goal is to show that then $\sum_{j=1}^{m} |B_j| \geq k\ell$.

Since $\sum_{i=1}^{n} |A_i| \leq n\ell$, at least $n/2$ of the sets A_i must be of size at most $r = 2\ell$. By the Sunflower Lemma, these sets must contain a sunflower with $s = 2\ell$ petals. Having such a sunflower with a core C, we can pair its members arbitrarily, $(A_{u_1}, A_{v_1}), \ldots, (A_{u_\ell}, A_{v_\ell})$. Important for us is that all ℓ symmetric differences $D_i = A_{u_i} \oplus A_{v_i} = (A_{u_i} \cup A_{v_i}) \setminus C$ are mutually disjoint.

Since the graph is k-isolated, each two vertices $u_i \neq v_i$ have a set $S_i \subseteq R$ of $|S_i| = k$ vertices, all of which are adjacent to u_i and none of which is adjacent to v_i. Hence, $|A_{u_i} \cap B_j| \neq |A_{v_i} \cap B_j|$ must hold for all $j \in S_i$. This implies that each set B_j with $j \in S_i$ must have at least one element in the symmetric difference $D_i = A_{u_i} \oplus A_{v_i}$. Hence,

$$\sum_{j=1}^{m} |D_i \cap B_j| \geq \sum_{j \in S_i} |D_i \cap B_j| \geq |S_i| = k \qquad \text{for each } i = 1, \ldots, \ell.$$

Since the sets $D_1, \ldots, D_\ell$ are disjoint, this implies

$$\sum_{j=1}^{m} |B_j| \geq \sum_{j=1}^{m} \sum_{i=1}^{\ell} |D_i \cap B_j| = \sum_{i=1}^{\ell} \sum_{j=1}^{m} |D_i \cap B_j| \geq \sum_{i=1}^{\ell} k = k\ell. \qquad \square$$

Drucker [13] showed that the lower bound in Theorem 1.54 is essentially optimal: there are strongly separated graphs for which this lower bound cannot be improved. In fact, the graph in [13] is *explicitly* constructed, and the upper bound holds already when one takes XOR function as the output gate. We now describe this construction.

The lower bound in Theorem 1.54 works by finding a large sunflower within the family of sets associated with the vertices on the left side L of the bipartition. Thus it is natural to try to use a set family without large sunflowers to show the tightness of Theorem 1.54.

Construction 1.55 (Drucker graphs) Let $n = p^s$ where p is a prime power and $1 \leq s \leq p$ and integer. Set $m := n/p$, and fix a boolean $p \times m$ matrix M

whose rows are labeled by elements $a \in \mathrm{GF}(p)$, columns by elements $x \in [m]$, and every two rows in M differ in at least $1/4$ of their positions. (For example, one can take the Sylvester $n' \times n'$ matrix H, where n' is the smallest power of 2 satisfying $n' \geq p^s$, and form M by taking the first p rows of H. By Proposition 1.49, we know that every two rows of H differ in exactly $n'/2 \geq n/4$ positions.) For $x \in [m]$, we identify the x-th column of our "ambient" matrix M with the set $S_x \subseteq \mathrm{GF}(p)$ of its 1-positions.

The *Drucker graph* $D_{n,s} \subseteq L \times R$ is an $n \times n$ graph with $n = p^s$ for a prime power p. Vertices in L are polynomials of degree at most $s-1$ over $\mathrm{GF}(p)$; hence $|L| = p^s = n$. Vertices in R are pairs (a, x) where $a \in \mathrm{GF}(p)$, $x \in [m] = \{1, \ldots, m\}$; hence, $|R| = pm = n$. Vertices $f \in L$ and $(a, x) \in R$ are adjacent in $D_{n,s}$ if and only if $f(a) \in S_x$.

Lemma 1.56 (Drucker [13]) *The graph $D_{n,s}$ has an intersection representation of weight $2pn$, and is k-isolated for $k \geq n(p-s)/4p$.*

Proof. To define the desired intersection representation of $D_{n,s}$, associate with each vertex $f \in L$ and each vertex $(a, x) \in R$ the following subsets of $\mathrm{GF}(p)^2$:

$$A_f := \{(a, b) \in \mathrm{GF}(p)^2 \colon f(a) = b\} \quad \text{and} \quad B_{(a,x)} := a \times S_x \,.$$

Since f is a *function* (cannot take more than one value), the intersection

$$A_f \cap B_{a,x} = A_f \cap (a \times S_x)$$

can have at most one element: the element $(a, f(a))$ if $f(a) \in S_x$, and no elements otherwise. Thus, $|A_f \cap B_{a,x}| = 1$ if vertices f and (a, x) are adjacent in $D_{n,s}$, and $|A_f \cap B_{a,x}| = 0$ otherwise. This gives us an intersection representation of $D_{n,s}$ relative to *any* boolean function which rejects the all-0 vectors, and accepts all vectors with exactly one 1.

Let us show that this representation has weight at most $2pn$. Since $|A_f| = p$ for every polynomial f, and $|S_x| \leq p$ for every $x \in [m]$, the weight of the representation is

$$\sum_{f \in L} |A_f| + \sum_{(a,x) \in R} |S_x| \leq p^s \cdot p + p \cdot m \cdot p = 2pn \,,$$

as desired. It remains to show that the symmetric difference $N(f) \oplus N(g)$ of sets of neighbors $N(f) \subseteq R$ and $N(g) \subseteq R$ of any two distinct vertices $f \in L$ and $g \in L$ is at least $n(p-s)/4p$. Recall that $N(f) = \{(a, x) \colon f(a) \in S_x\}$.

For $a \in \mathrm{GF}(p)$, let $\Delta_a := \{(a, x) : f(a) \in S_x \text{ iff } g(a) \notin S_x\}$ denote the set of columns S_x of our "ambient" matrix M, whose entries in the $f(a)$-th and $g(a)$-th rows of M are distinct. Since, by the choice of M, every two distinct rows of M differ in at least $1/4$ of their $m = n/p$ positions, we have that

$|\Delta_a| \geq m/4 = n/4p$ for every $a \in D := \{a \in \mathrm{GF}(p)\colon f(a) \neq g(a)\}$. On the other hand, since any polynomial of degree s can have at most s roots, the set D has $|D| \geq p - s$ elements. Thus,

$$|N(f) \oplus N(g)| = \sum_{a \in \mathrm{GF}(p)} |\Delta_a| \geq \sum_{a \in D} |\Delta_a| \geq (p-s)\frac{n}{4p}\,. \qquad \square$$

By taking $n = p^s$ with $s = \lfloor p/2 \rfloor$ in Lemma 1.56, we obtain an explicit strongly isolated graph $G = D_{n,s}$ which *can* be represented by a symmetric depth-2 formula of weight at most about $n \ln n / \ln\ln n$. Thus, the lower bound in Theorem 1.54 is actually tight.

1.7 Depth-3 complexity

In Section 1.6 we considered representation of graphs by the simplest kind of depth-2 formulas—CNFs, that is by ANDs of ORs. Now we increase the depth by 1, and consider formulas that are ORs of CNFs. We call such formulas *depth-3 OR-formulas*. The *middle fanin* in such a formula is the maximum number of clauses in its CNFs, and the *top fanin* is the total number of CNFs used. By the *size* of such a formula we will mean the maximum of its top and middle fanins.

Explicit boolean functions of l variables requiring depth-3 formulas of size $2^{\Omega(\sqrt{l})}$ are known. In particular, such is the XOR function $x_1 \oplus x_2 \oplus \cdots \oplus x_l$, as well as the majority function which outputs 1 if and only if the input vector has more ones than zeros. Using counting arguments it is not difficult to show that most boolean functions require depth-3 formulas of size about $2^{l/2}$. But all attempts to improve the $2^{\Omega(\sqrt{l})}$ lower bound for an explicit function failed so far. To break this "square-root" barrier is one of the challenges in circuit complexity.

An even bigger challenge is to prove a lower bound of the form $2^{\alpha l/\ln\ln l}$ for a growing $\alpha \to \infty$. By Valiant's result [46], this would resolve at least two widely open problems in circuit complexity (see, e.g., Chapter 11 in [22] on how does this happen). On the other hand, these problems can be solved by exhibiting bipartite graphs requiring large OR-circuits of depth 3.

For a graph G, let $\mathrm{Star}_3(G)$ denote the minimum size of a monotone depth-3 OR-formula representing G, that is, the smallest number s such that G can be represented by a formula of the form

$$F(x) = \bigvee_{i=1}^{s} \bigwedge_{j=1}^{s} \bigvee_{u \in S_{ij}} x_u\,.$$

For a boolean function f, let $\mathrm{Circuit}_3(f)$ denote the smallest size of a (not necessarily monotone) depth-3 formula computing f. Since we have *unbounded*

fanin OR gates at the bottom, the Magnification lemma immediately yields the inequality

$$\mathrm{Circuit}_3(f_G) \geq \mathrm{Star}_3(G)\,.$$

Remark 1.57 This latter inequality has no converse. To see this, consider the bipartite $n \times n$ graph $G \subseteq L \times R$ with $n = 2^l$ which is a union of two bicliques (complete bipartite graphs) $L_0 \times R_1$ and $L_1 \times R_0$, where L_0 (L_1) is the set of all $2^{l-1} = n/2$ vertices $u \in L$ whose binary code has an even (resp., odd) number of 1s; sets R_0 and R_1 are defined similarly. Since every biclique can be represented by the AND of two ORs (see Example 1.6), we have that $\mathrm{Star}_3(G) \leq 2$. But the adjacency function $f_G(y,z)$ of this graph is the parity function of $2l$ variables, and it is well known (see, e.g., [22]) that $\mathrm{Circuit}_3(f_G) = 2^{\Theta(\sqrt{l})} = 2^{\Theta(\sqrt{\log n})}$.

Eq. 1.15, together with an obvious observation that every bipartite clique $A \times B$ can be represented by a CNF consisting of two clauses $\bigvee_{u\in A} x_u$ and $\bigvee_{v\in B} x_v$, gives an upper bound:

$$\mathrm{Star}_3(G) \leq \min\left\{\mathrm{bc}(G), \mathrm{bc}(\overline{G})\right\}\,, \tag{1.19}$$

where $\mathrm{bc}(G)$ is the smallest number of bipartite complete subgraphs of G covering all edges of G.

Let $\mathrm{Star}_3(n)$ denote the maximum of $\mathrm{Star}_3(G)$ over all bipartite $n \times n$ graphs.

Proposition 1.58 $\sqrt{n/2} \leq \mathrm{Star}_3(n) \leq \sqrt{n}$.

Proof. Upper bound. Let G be a bipartite $n \times n$ graph. Split G into $s = \sqrt{n}$ bipartite $(n/s) \times n$ graphs, $G = H_1 \cup \cdots H_s$. By Proposition 1.41, $\mathrm{cnf}(H_i) \leq \min\{s, n/s\} = \sqrt{n}$ for all $i = 1, \ldots, s$. Thus, the original graph G can be written as a union of s graphs, each of which can be represented by a depth-2 circuit (a CNF) of size s. This shows that $\mathrm{Star}_3(G) \leq \sqrt{n}$.

Lower bound. Since every CNF (depth-2 circuit) represents an intersection of bipartite complements of bicliques (see (1.14)), we have that $\mathrm{Star}_3(G) \leq s$ if and only if the graph G can be written in the form

$$G = \bigcup_{i=1}^{s} \bigcap_{j=1}^{s} \overline{A_{ij} \times B_{ij}}\,.$$

Since we have only 2^{2n} possibilities to choose a biclique $A_{ij} \times B_{ij}$, the number of graphs representable in such a form does not exceed $(2^{2n})^{s^2} = 2^{2ns^2}$. Since we have 2^{n^2} graphs, at least one of them will require $s \geq \sqrt{n/2}$. □

Open Problem 1.59 *Exhibit an explicit sequence G_n of bipartite $n \times n$ graphs with $\mathrm{Star}_3(G_n) \geq n^\epsilon$ for a constant $\epsilon > 0$.*

By the result of Valiant mentioned above, even a lower bound of n^α for $\alpha = \omega(1/\ln\ln\ln n)$ would resolve some old problems in circuit complexity. Unfortunately, the currently best lower bound remains that proved by Lokam [30].

Theorem 1.60 (Lokam [30]) *Let H be an $n \times n$ Hadamard graph. Then every monotone depth-3 formula representing H must have $\Omega(\log^3 n)$ AND gates on the bottom level.*

In view of the difficulties to prove strong lower bounds for depth-3 complexity of graphs, even understanding the depth-2 complexity is a challenge. As mentioned in Problem 1.44, dense $K_{2,2}$-free graphs could be good candidates in this latter model. We already know how to construct $K_{2,2}$-free bipartite $n \times n$ graphs G with $|G| \geq n^{3/2}$ edges (see Constructions 1.27 and 1.28). Since each depth-3 circuit is an OR of CNFs, a positive solution of Problem 1.44 would resolve Problem 1.59.

Two parameters of depth-3 circuits determining their size is the top fanin s and the middle fain r; the size of a circuit is then $\max\{s, r\}$. As we mentioned above, no explicit lower bounds on $\max\{s, r\}$ larger than $\mathrm{Star}_3(G) = \Omega(\log^{3/2} n)$ are known. On the other hand, we have the following trade-off between these two parameters.

Lemma 1.61 ([19]) *If a bipartite graph G can be represented by a monotone depth-3 formula of middle fanin r and top fanin s, then $s2^r \geq \mathrm{bc}(G)$ and $r^s \geq \mathrm{bc}(\overline{G})$.*

Proof. Take a monotone depth-3 formula of middle fanin at most r and top fanin s, and let $G \subseteq L \times R$ be the bipartite graph represented by this formula. Each gate $g = \bigvee_{i \in A \cup B} x_i$ on the bottom (next to the inputs) level, with $A \subseteq L$ and $B \subseteq R$, represents the union $H = (A \times R) \cup (L \times B)$ of two bipartite cliques (see Fig. 1.2). Since each AND on the middle level has fanin at most r, and since the intersection of any number of bipartite cliques is a (possibly empty) bipartite clique, each AND gate on the middle level represents a union of at most 2^r bipartite cliques. Since G is a union of s such graphs, we have $\mathrm{bc}(G) \leq s2^r$.

To prove $\mathrm{bc}(\overline{G}) \leq r^s$, observe that $\overline{G}$ is an intersection of s graphs, each of which is a union of r bipartite cliques. Since the intersection of any number of bipartite cliques is a bipartite clique, we have $\mathrm{bc}(\overline{G}) \leq r^s$. □

Recall that a bipartite n-matching is an $n \times n$ graph M_n consisting of n vertex-disjoint edges. Let $n = 2^l$. We already know that $\mathrm{cnf}(M_n) \leq l = \log_2 n$ (see Proposition 1.42). Thus, M_n can be represented by a depth-3 circuit of middle fanin $r = \log_2 n$ and top fanin $s = 1$. On the other hand, M_n (as well as every other graph) can be represented by a circuit with $r = 2$: let the middle fanin-2 AND gates to represent bicliques. But Lemma 1.61 implies that every depth-3 circuit for M_n with middle fanin $r \leq \epsilon \log_2 n$ must have large top fanin: $s \geq n/2^r = n^{1-\epsilon}$.

Depth-3 complexity with XOR bottom gates

Being unable to prove strong lower bounds for depth-3 formulas, where bottom (next to the input literals) gates are OR gates, we now consider the same problem for depth-3 formulas where bottom gates are XOR gates, that is, sums modulo 2 of their inputs.

By an *XOR-formula* of depth-3 we will mean a formula with unbounded fanin XOR gates on the bottom (next to the inputs) level, followed by unbounded fanin AND gates on the middle level feeding into the bottom OR gate. By the *size* of such a circuit we will mean the fanin of the top (output) gate; that is, we ignore the number of XOR gates used—it may be arbitrarily large. Such a formula is *positive* if no negated variables are used as inputs.

For a graph G, let $\mathrm{Star}_3^*(G)$ denote the size of a positive depth-3 XOR-formula circuit representing G, that is, the smallest number s such that G can be represented by a formula of the form

$$F(x) = \bigvee_{i=1}^{s} \bigwedge_{j=1}^{r_i} \bigoplus_{u \in S_{ij}} x_u \,.$$

The Magnification Lemma implies that for every bipartite graph G,

$$\text{the top fanin of any XOR-circuit for } f_G \text{ is at least } \mathrm{Star}_3^*(G)\,. \tag{1.20}$$

On the other hand, we have the following general lower bound on $\mathrm{Star}_3^*(G)$. A graph is $K_{a,b}$*-free* if it does not contain a complete $a \times b$ subgraph.

Theorem 1.62 ([19]) *If an $n \times n$ graph G is $K_{a,b}$-free, then*

$$\mathrm{Star}_3^*(G) \geq \frac{|G|}{(a+b)n}\,.$$

Proof. To prove the theorem, we first give a combinatorial characterization of $\mathrm{Star}_3^*(G)$ of the top fanin of $\Sigma_3^{\oplus}$ circuits representing bipartite graphs (Claim 1.63), and then a general lower bound on this characteristics (Claim 1.64).

Recall that a *fat matching* is a bipartite graph consisting of vertex-disjoint bipartite cliques (these cliques need not to cover all vertices). Note that a matching (a set of vertex-disjoint edges) is also a fat matching. A *fat covering* of a graph G is a family of fat matchings such that each of these fat matchings is a subgraph of G and every edge of G is an edge of at least one member of the family. Let $\mathrm{fat}(G)$ denote the minimum number of fat matchings in a fat covering of G.

Pudlák and Rödl [40] proved that $\mathrm{fat}(G) = \mathcal{O}(n/\log n)$ for every $n \times n$ graph G. We now show that $\mathrm{fat}(G)$ is exactly the depth-3 XOR-formula complexity of G.

Claim 1.63 *For every bipartite graph G,* $\mathrm{Star}_3^*(G) = \mathrm{fat}(G)$.

Proof. The claim follows fairly easily from the observation that each XOR gate $\bigoplus_{w\in S} x_w$ accepts an edge (u,v) if and only if the set S contains *exactly one* of the endpoints u and v. Thus, each such gate represents the union of two vertex disjoint bicliques, that is, a fat matching (see Fig. 1.2(c)). Since the intersection of any number of fat matching is again a fat matching, the claim follows. We leave the details to the reader. □

Claim 1.64 *For every $K_{a,b}$-free bipartite $n\times n$ graph,* $\mathrm{fat}(G) \geq |G|/(a+b)n$.

Proof. Let $H = \bigcup_{i=1}^t A_i \times B_i$ be a fat matching, and suppose that $H \subseteq G$. By the definition of a fat matching, the sets $A_1,\ldots,A_t$, as well as the sets $B_1,\ldots,B_t$ are mutually disjoint. Moreover, since G contains no copy of $K_{a,b}$, we have that $|A_i| < a$ or $|B_i| < b$ for all i. Hence, if we set $I = \{i : |A_i| < a\}$, then

$$|H| = \sum_{i=1}^t |A_i \times B_i| = \sum_{i=1}^t |A_i|\cdot|B_i| \leq \sum_{i\in I} a\cdot|B_i| + \sum_{i\notin I} |A_i|\cdot b \leq (a+b)n.$$

Thus, no fat matching $H\subseteq G$ can cover more than $(a+b)n$ edges of G, implying that we need at least $|G|/(a+b)n$ fat matchings to cover all edges of G. □

Theorem 1.62 is now a direct consequence of these two claims. □

There are many explicit bipartite graphs which are dense enough and do not have large complete bipartite subgraphs. By Theorem 1.62, each of these graphs G gives us an explicit boolean function f_G requiring large depth-3 formulas with bottom XOR gates.

To give an example, consider the bipartite Kneser graph D_n. Recall that this is a bipartite $n\times n$ graph with $n = 2^l$ whose vertices u in each color class are subsets of $[l] = \{1,\ldots,l\}$, and two vertices u and v are adjacent if and only if $u\cap v = \emptyset$. Thus, the graph D_n has disjointness dimension $\theta(G) \leq l = \log_2 n$ and, by (1.15), also $\mathrm{Star}_3(D_n) \leq \mathrm{cnf}(D_n) \leq \log_2 n$. We now show that the depth-3 complexity of this graph is much larger, if we require bottom gates be XOR gates.

Theorem 1.65 $\mathrm{Star}_3^*(D_n) \geq n^{0.08}/2$.

Proof. As we argued in the proof of Theorem 1.12, the graph D_n is $K_{a,a}$-free for $a = \sqrt{n}$. Since D_n has $|D_n| \geq n^{1.58}$ edges (see Example 1.9), Theorem 1.62 implies

$$\mathrm{Star}_3^*(D_n) \geq \frac{|D_n|}{2an} \geq \frac{n^{1.58}}{2n^{1.5}} = n^{0.08}/2\,.$$ □

The adjacency function of the graph D_n is the well-known *disjointness function* of $2l = 2\log_2 n$ variables:

$$DISJ_{2l}(y_1,\ldots,y_l,z_1,\ldots,z_l) = 1 \text{ if and only if } \bigvee_{i=1}^{l} y_i \wedge z_i = 0\,.$$

This function can be computed by a depth-2 AND-OR formula $\wedge_{i=1}^{l}(\overline{x}_i \vee \overline{y}_i)$ with $l+1$ gates. If, however, we replace bottom OR gates by XOR gates, then exponential number of gates is necessary, even in depth-3. This immediately follows from Theorem 1.65 and the lower bound (1.20).

Corollary 1.66 *Any depth-3 formula for $DISJ_{2l}$ with XOR gates on the bottom must have top fanin at least about $2^{0.08l}$.*

We now consider a generalization of depth-3 XOR-circuits, where we allow to use an arbitrary *threshold* gate (instead of an OR gate) on the top. Each threshold function of n variables is defined by specifying its *threshold value* $0 \le t \le n$; the gate accepts a boolean vector if and only if it has at least t ones. Thus, XOR-formulas we considered above (with an OR gate on the top) correspond to the case $t = 1$.

We are going to show that Hadamard graphs (see Example 1.10) require large XOR-circuits of depth 3 even if an arbitrary threshold function is allowed to be used as the top (output) gates. For this, we will use the well-known fact that Hadamard matrices are "balanced'.

Lindsey's Lemma *The absolute value of the sum of all entries in any $a \times b$ submatrix of an $n \times n$ Hadamard matrix M does not exceed $\sqrt{abn}$.*

In particular, if $ab > n$ then no $a \times b$ submatrix of M is monochromatic.

Proof. Let M be an $n \times n$ Hadamard matrix, and A one of its $a \times b$ submatrices. Assume for simplicity that A consists of its first a rows and b columns. Let α be the sum of all entries of A. We want to prove that $\alpha \le \sqrt{abn}$.

Let $v_1,\ldots,v_a$ be the first a rows of H, and $y = \sum_{i=1}^{a} v_i$. If we take the vector $x = (1^b 0^{n-b})$, then $\alpha^2 = \langle x,y\rangle^2 \le \|x\|^2\|y\|^2 = b \cdot \|y\|^2$. On the other hand, the conditions $\langle v_i,v_i\rangle = n$ and $\langle v_i,v_j\rangle = 0$ for all $i \ne j$ imply that $\|y\|^2 = \sum_{i,j=1}^{a}\langle v_i,v_j\rangle = \sum_{i=1}^{a}\langle v_i,v_i\rangle = an$. Thus $\alpha^2 \le b \cdot \|y\|^2 = abn$, as desired. □

We now will use Lindsey's Lemma to show that Hadamard graphs require large top fanin in depth-3 XOR-formulas even if arbitrary threshold function is allowed to be used as the top (output) gates.

Theorem 1.67 ([19]) *Any XOR-formula of depth 3, which has an arbitrary threshold gate on the top and represents an $n \times n$ Hadamard graph, must have top fanin $\Omega(\sqrt{n})$.*

Proof. Let $H \subseteq L \times R$ be an $n \times n$ Hadamard graph. Fix an XOR-formula of depth 3 with an arbitrary threshold gate on the top, and assume that the circuit represents H. Let s be the top fain of that circuit (the number of inputs into the output (threshold) gate, and let t be the threshold of that gate. By Claim 1.63, we know that graphs $F_1, \ldots, F_s$ represented by the depth-2 XOR-subcircuits feeding into the output threshold gate are fat matchings. Thus, a pair $(u, v) \in L \times R$ of vertices is an edge of H if and only if (u, v) belongs to at least t of the F_i. Define the *discrepancy*, $p(F_i)$, of F_i relative to H by:

$$p(F_i) := \left| \frac{|H \cap F_i|}{|H|} - \frac{|\overline{H} \cap F_i|}{|\overline{H}|} \right|$$

Claim 1.68 *For at least one $i = 1, \ldots, s$, we have $p(F_i) \geq 1/s$.*

Proof. Since every edge of H belongs to at least t of the sets $H \cap F_i$, the average size of these sets must be at least t. Since no edge of $\overline{H}$ belongs to more than $t-1$ of the sets $\overline{H} \cap F_i$, the average size of these sets must be at most $t-1$. Hence,

$$1 \leq \sum_{i=1}^{s} \frac{|H \cap F_i|}{|H|} - \sum_{i=1}^{s} \frac{|\overline{H} \cap F_i|}{|\overline{H}|} \leq s \cdot \max_{1 \leq i \leq s} p(F_i)\,. \qquad \square$$

Claim 1.69 *For every fat matching F, $p(F) \leq 4/\sqrt{n}$.*

Proof. Take an arbitrary fat matching $F = \bigcup_{i=1}^{\ell} S_i \times R_i$. Let Δ be the absolute value of the difference between $|H \cap F|$ and $|\overline{H} \cap F|$. Since both the graph H and its bipartite complement $\overline{H}$ have at least $n^2/4$ edges, it is enough to show that $\Delta \leq n^{3/2}$. By Lindsey's Lemma, the absolute value of the difference between $|H \cap (S_i \times R_i)|$ and $|\overline{H} \cap (S_i \times R_i)|$ does not exceed $\sqrt{s_i r_i n}$, where $s_i = |S_i|$ and $r_i = |R_i|$. Since both sums $\sum_{i=1}^{\ell} s_i$ and $\sum_{i=1}^{\ell} r_i$ are at most n, we obtain

$$\Delta \leq \sum_{i=1}^{\ell} \sqrt{s_i r_i n} \leq \sqrt{n} \sum_{i=1}^{\ell} \frac{s_i + r_i}{2} \leq n^{3/2}\,. \qquad \square$$

The desired lower bound $s = \Omega(\sqrt{n})$ on the top fanin of our circuit representing H follows directly by comparing bounds in Claims 1.68 and 1.69. $\square$

Theorem 1.67 has the following consequence for boolean functions. The *inner product function* is a boolean function of $2l$ variables defined by

$$IP_{2l}(y_1, \ldots, y_l, z_1, \ldots, z_l) = \sum_{i=1}^{l} y_i z_i \bmod 2\,.$$

This function has a trivial depth-2 XOR-AND circuit with $l+1$ gates. If, however, we replace the roles of gates and consider AND-XOR circuits, then

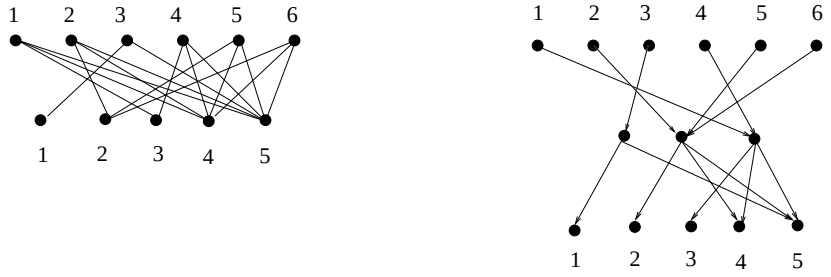


Fig. 1.7 A bipartite 6×5 graph and a depth-2 rectifier network realizing it.

even using an arbitrary threshold function of such circuits will not help: and exponential number of AND gates is then necessary. This directly follows from Theorem 1.67 and the lower bound (1.20), because IP_{2l} is the adjacency function of the Sylvester $2^l \times 2^l$ graph.

Corollary 1.70 *Any XOR-formula of depth* 3 *for* IP_{2l}*, which has an arbitrary threshold gate on the top and represents an* $n \times n$ *Hadamard graph, must have top fanin at least about* $2^{l/2}$*.*

1.8 Network complexity of graphs

Let $G \subseteq L \times R$ be a bipartite $n \times m$ graph. Suppose we want to keep all connections between vertices in L and R, but would like to use as few edges as possible. That is, the goal is to replace edges of G by *paths* so that the total number of edges in a new graph is smaller than $|G|$. Such representations of graphs are called "rectifier networks" or "diode networks".

A *rectifier network* realizing a graph $G \subseteq L \times R$ is a directed acyclic graph F whose input (fanin-0) nodes are vertices in L, output (fanout-0) nodes are vertices in R, and $(u, v) \in G$ if and only if there exists a path from u to v in F. The *size* of a network is the number of wires in it. The *depth* of a network is the maximum number of edges on a path from an input node to an output node.

Note that the network size of a graph G can be much smaller than the number $|G|$ of edges in the graph itself. For example, a complete bipartite graph $K_{n,m} = L \times R$ has nm edges, but can be realized by a depth-2 rectifier network with $n + m$ wires: just take one node $w \notin L \cup R$, and connect it with all nodes in L and in R.

Theorem 1.71 (Lupanov [32]) *Every bipartite* $n \times n$ *graph can be realized by a depth-2 rectifier network using* $2n^2/\log_2 n$ *wires, and graphs requiring about* $n^2/\log_2 n$ *wires in any rectifier network exist.*

Proof. To prove the upper bound, take an arbitrary bipartite $n \times n$ graph G. Lemma 1.8 states that G can be decomposed into bicliques so that the total weight (sum of the numbers of their vertices) of these bicliques does not exceed $2n^2/\log n$. Since (as we have seen) each biclique $S \times T$ can be realized by a network of depth-2 using only $|S| + |T|$ wires, we are done.

To prove the lower bound, we first estimate the number of rectifier networks of a given size, and then compare this number with the total number 2^{n^2} of graphs that must be realized.

Claim 1.72 *There exist at most* $(9t)^t$ *graphs with* t *edges.*

Proof. Every set of t edges is incident with at most $2t$ nodes. Using these nodes, at most $r = (2t)^2$ their pairs (potential edges) can be built. Since $x_1 + \ldots + x_r = t$ has $\binom{r+t-1}{t}$ integer solutions $x_i \geq 0$, and since $t! \geq (t/3)^t$ (by Stirling's formula), the number of graphs with t edges is at most

$$\binom{r+t-1}{t} \leq \frac{(r+t-1)^t}{t!} \leq \frac{3^t(r+t-1)^t}{t^t} \leq \frac{3^{2t}t^{2t}}{t^t} = 3^{2t}t^t\,. \qquad \square$$

By Claim 1.72, we cannot realize all graphs by networks of size t unless $(9t)^t \geq 2^{n^2}$, from which $t = \Omega(n^2/\log n)$ follows. $\square$

Several authors obtained even *asymptotically tight* bounds. Let $\mathrm{Wires}(n)$ denote the maximum, over all $n \times n$ graphs G, of the smallest number of wires in a rectifier network realizing G. Let also $\mathrm{Wires}_d(n)$ denote this measure when restricted to rectifier networks of depth d.

Lupanov [32] proved that $\mathrm{Wires}_2(n) \sim n^2/\log_2 n$. Nechiporuk [34] proved that the asymptotic for unbounded-depth networks is achieved at depth 3, namely $\mathrm{Wires}(n) \sim \mathrm{Wires}_3(n) \sim n^2/2\log_2 n$. In the same paper, Nechiporuk also obtained asymptotic bounds for graphs of a given density α. Let $\mathrm{Wires}_d(n,\alpha)$ denote the minimal number of wires which is enough to represent any bipartite $n \times n$ matrix with $|G| = \alpha n^2$ edges. Then $\mathrm{Wires}_2(n,\alpha) \sim H(\alpha) \cdot n^2/\log_2 n$, and $\mathrm{Wires}_3(n,\alpha) \sim H(\alpha) \cdot n^2/2\log_2 n$ as long as $\log_2 n \ll H(\alpha)n$ and $-\log_2 \min(\alpha, 1-\alpha) \ll \log_2 n$, where $H(x) := -x\log_2 x - (1-x)\log_2(1-x)$ be the binary entropy function.

Orlov [38] considered the realization of bipartite $m \times n$ graphs with $m \ll n$ rows and proved that $\mathrm{Wires}_2(k\log_2 n, n) \sim (k+1)n$ holds for every positive integer k, and $\mathrm{Wires}(m,n) \sim \mathrm{Wires}_2(m,n) \sim 2^{m+1} + n$ holds as long as $n \geq 2(2^m - m - 1)$.

In all these estimates, the upper bounds were obtained by constructing networks with a special property that every input is connected with every output by *at most one* path.

The bounds above only show that "hard-to-realize" graphs *exist*, and give no clue on *which* (specific) graphs are such. We now will exhibit several such "hard" graphs.

Say that a bipartite graph G is *k-free* if it does not contain any copy of a complete bipartite $k \times k$ graph. The following lower bound for k-free graphs was proved by several authors [33, 39, 47].

Theorem 1.73 *If a bipartite graph G is $(k+1)$-free, then every rectifier network realizing G must have at least $|G|/k^2$ wires.*

Proof. (Due to Pippenger [39]) Take a rectifier network F realizing G. For a node x in F, let s_x be the number of input nodes from which x is reachable, and t_x the number of output nodes reachable from x. Let us call a wire $e = (x, y)$ of F *eligible* if $s_x \leq k$ *and* $t_y \leq k$. Say that an edge $(u, v) \in G$ of the given graph G is *covered* by a wire $e = (x, y)$ of F, if there is a path in F from the input node u to x, and there is a path from y to the output node v.

Since each eligible wire $e = (x, y)$ can cover at most $s_x \cdot t_y \leq k^2$ edges of G, it remains to prove the following claim.

Claim 1.74 *Every edge of G is covered by at least one eligible wire of F.*

To prove the claim, take an edge $(u, v) \in G$. Then there must be a path $x_0, x_1, \ldots, x_r$ in the network F beginning in $x_0 = u$ and ending in $x_r = v$. Letting $s_l := s_{x_l}$ to be the number of input nodes in L from which x_l is reachable, and $t_l := t_{x_l}$ be the number of output nodes in R reachable from x_l, we have that $s_1 \leq s_2 \leq \ldots \leq s_r$ and $t_1 \geq t_2 \geq \ldots \geq t_r$.

Let p be the largest number for which $s_p \leq k$, and q the smallest number for which $t_q \leq k$. If $q \leq p + 1$, then the wire $e = (x_p, x_{p+1})$ of F covering the edge (u, v) of G is eligible, and we are done. So assume for the sake of contradiction that $q \geq p + 2$. By the definition of positions p and q, we have that $s_{p+1} > k$ and $t_{p+1} > k$. But then at least $k+1$ inputs of F are connected to at least $k+1$ outputs going through the node x_{p+1}, contradicting the $(k + 1)$-freeness of G. This completes the proof of the claim, and thus the proof of the theorem. □

There are several constructions of dense bipartite $n \times n$ graphs that are k-free. In Constructions 1.27 and 1.28 above give explicit graphs requiring $\Theta(n^{3/2})$ wires in any rectifier network realizing them. These graphs have $\Omega(n^{3/2})$ edges and are k-free for $k = 1$. Allowing larger values of k, one can construct k-free graphs with more edges.

Construction 1.75 (2-free graphs) The following construction of dense 2-free graphs is due to Brown [8]. Let p be an odd prime and let d be a non-zero element of $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ (the field of integers modulo p) such that d is a quadratic non-residue modulo p if $p \equiv 1$ modulo 4, and a quadratic residue modulo p if $p \equiv 3$ modulo 4. Let $n = p^3$, and consider the bipartite $n \times n$ graph G whose vertices correspond to all triples of elements in $\mathbb{Z}_p$. The vertices G corresponding to triples (a_1, a_2, a_3) and (b_1, b_2, b_3) are adjacent in G if and only if the sum $(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2$ modulo p is equal to d. Brown

showed that this graph has $|G| = p^4(p-1) = \Omega(n^{5/3})$ edges, and is 2-free. Thus, every rectifier network realizing G must have $\Omega(n^{5/3})$ wires.

Subsequent constructions of dense square-free matrices have lead to even higher lower bounds.

Construction 1.76 (Norm graphs) Let q be a prime-power, $t \geq 2$ an integer, and consider the field $\mathrm{GF}(q^t)$ with q^t elements. The norm of an element a of this field is defined as the element $N(a) := a \cdot a^q \cdots a^{q^{t-1}} = a^{(q^t-1)/(q-1)}$ of this field. Now let $n = q^t$, and construct a bipartite $n \times n$ graph with vertices in each part being elements of $\mathrm{GF}(q^t)$. Two vertices a and b are adjacent iff $N(a+b) = 1$. It is known that the number of solutions in $\mathrm{GF}(q^t)$ of the equation $N(x) = 1$ is $(q^t-1)/(q-1)$; this and other basic facts about finite fields can be found in the book by Lidl and Niederreiter [29]. Hence, each vertex of this graph has degree $d = (q^t-1)/(q-1)$, implying that the total number of edges is $dq^t \geq q^{2t-1} = n^{2-1/t}$. Kollár, Rónyai and Szabó [26] proved that, for any t distinct elements $a_1, \ldots, a_t$ of $\mathrm{GF}(q^t)$, the system of equations $N(a_1+x) = 1, N(a_2+x) = 1, \ldots, N(a_t+x) = 1$ has at most $t!$ solutions $x \in \mathrm{GF}(q^t)$. This immediately implies that the constructed graph G has no copy of a complete bipartite $t \times (t!+1)$ graph, and hence, is k-free for $k = t!$. Thus, every rectifier network realizing G must have $\Omega(n^{2-1/t}/t!)$ wires. Explicit graphs with slightly worse parameters were constructed earlier by Andreev [3].

Realizing graphs by circuits

Recall that a rectifier network F realized a graph $G \subseteq L \times R$ if for every $u \in L$ and $v \in R$, $(u,v) \in G$ if and only if *there exists* a path in F from the input node u to the output node v. Attach now to each non-input node of F an OR gate of its inputs. Then the resulting circuit over $\{\vee\}$ computes an operator $F : \{0,1\}^L \to \{0,1\}^R$ which *realizes* the graph G in the following sense: for every $u \in L$ and $v \in R$,

$F_v(\mathrm{e}_u) = 1$ if and only if $(u,v) \in G$;

here F_v is the v-th component of the operator $F = (F_v \colon v \in L)$, and $\mathrm{e}_u \in \{0,1\}^L$ is the binary vector with exactly one 1 in the u-th position. That is, for every input e_u, the circuit must compute the characteristic vector of the set of neighbors of u in G.

Motivated by this observation, one can consider realizations of graphs by circuits over bases where not only OR gates can be used. In particular, can the number of wires can be substantially decreased if one allows also AND gates? As shown by Nechiporuk [35], Pippenger [39], and Mehlhorn [33], at least for k-free graphs this is not the case: the number of wires can only be decreased by a factor at most $1/k$. By Construction 1.76, for every constant $t \geq 2$, there are explicit $n \times n$ graphs requiring $\Omega(n^{2-1/t})$ AND and OR gates to realize them.

If we consider *linear* circuits, that is, circuits consisting of unbounded fanin XOR gates, then such a circuit represents a graph G if and only if it computes the linear transformation $y = Ax$ over GF(2), where A is the adjacency matrix of G. Using a similar argument as in the proof of Theorem 1.71, one can show that $n \times n$ graphs requiring $\Omega(n^2/\log n)$ wires exist, and that $\mathcal{O}(n^2/\log n)$ wires are always enough, even using depth-2 circuits. But so far, no *explicit* graph requiring more than $n \log^2 n$ wires is known; the problem remains open even for depth-2 circuits.

An extreme case is to allow *arbitrary* boolean functions be used as gates. How many wires do the graph need to be realized by such general circuit?

By Theorem 1.71, we know that some explicit $n \times n$ graphs (like norm graphs for an arbitrary large integer $t \geq 1$) require about $n^{2-1/t}$ wires to be realized by circuits using only AND and OR gates, regardless of the depth of the circuit used. We now show that the situation changes drastically, if we allow more general gates: then every graph can be realized even by depth-2 circuits using about $n \log n$ wires. This can already achieved by allowing multilinear polynomials of degree $\log n$ as gates.

Theorem 1.77 ([21]) *Every bipartite $n \times n$ graph can be realized by a general depth-2 circuit using at most $1.5n \log_2 n$ wires.*

Proof. Let $G \subseteq L \times R$ be a bipartite $n \times n$ graph. We construct the desired depth-2 circuit F realizing G as follows. Take $r = \log_2 n$ middle nodes $W = \{w_1, \ldots, w_r\}$. Since $\binom{r}{r/2} \geq n$, we can assign to each input node $u \in L$ its *own* subset $S_u \subseteq W$ of $|S_u| = r/2$ middle nodes; hence, $S_{u_1} \subseteq S_{u_2}$ if and only if $u_1 = u_2$. Join u with all nodes in S_u. Finally, connect each $w \in W$ with all output nodes in V. The total number of wires is then $n(r/2)+nr = 1.5n \log_2 n$.

Now we assign gates to the nodes. At each node w on the middle layer of F we compute an OR g_w of its inputs. (Instead of ORs one can take any boolean functions that reject the all-0 vector, and accept every vector with exactly one 1.) To each output node $v \in V$ assign the gate

$$\phi_v = \bigvee \{h_x \colon x \in L, (x, v) \in G\} \quad \text{where} \quad h_x = \bigwedge_{w \in S_x} g_w \,.$$

Then

$$\begin{aligned} h_x(\mathrm{e}_u) = 1 \text{ iff } & g_w(\mathrm{e}_u) = 1 \text{ for all } w \in S_x \\ & \text{iff } u \text{ is connected in } F \text{ to all nodes in } S_x \\ & \text{iff } S_x \subseteq S_u \text{ iff } x = u. \end{aligned}$$

Hence, for every $u \in L$, we have that $h_u(\mathrm{e}_u) = 1$ and $h_x(\mathrm{e}_u) = 0$ for all $x \neq u$. But this means that the function $F_v(x)$ computed at the v-th will output 1 on vector e_u if and only if (u, v) is an edge of G, as desired. □

Remark 1.78 Drucker [13] used probabilistic arguments to show that the upper bound given in Theorem 1.77 is almost optimal: there *exist* bipartite $n \times n$ graphs G that need $\Omega(n \log n)$ wires to realize them by general circuits of arbitrary depth.

As always, the most intriguing question is to exhibit *explicit* graphs requiring many wires to realize them. It turns out that every graph, that is isolated "well enough" (see Definition 1.48) requires almost the maximal number $n \log n$ wires.

Recall that a bipartite graph $G \subseteq L \times R$ is *k-isolated* if for any two distinct vertices $x \neq y \in L$ there exists a subset $S \subseteq R$ of $|S| = k$ vertices such that every vertex $v \in S$ is adjacent to exactly one of the vertices x and y. Say that G is *strongly isolated* if it is k-isolated for $k = \Omega(n)$, where $n = |L|$ is the number of vertices on the left side. In particular, the Sylvester $n \times n$ graph is k-isolated for $k \geq n/2$, and hence, is strongly isolated.

Building on work of Alon, Karchmer and Wigderson [2], the following lower bound for general circuit complexity of graphs was proved in [21].

Theorem 1.79 *If a bipartite $n \times n$ graph G is k-isolated, then every general depth-2 circuit realizing G must have $\Omega(k \cdot \ln n / \ln \ln n)$ wires.*

The proof is similar to that of Theorem 1.54 above. On the other hand, the construction of Drucker [13] (see Lemma 1.56 above) shows that one needs other properties of graphs to force more wires: some explicit strongly isolated graphs can be realized by depth-2 circuits with $\mathcal{O}(n \ln n / \ln \ln n)$ wires, even if only OR functions or only XOR functions are used as gates. In particular, this upper bound holds also in the class of rectifier networks.

1.9 Conclusion and open problems

The star complexity of a graph is the smallest number of union and intersection operations required to generate the graph when starting from stars. An intriguing aspect of this measure is its connection to circuit complexity of Boolean functions and, in particular, to the **P** versus **NP** problem. In this chapter we described this connection as well as known bounds on the star complexity of explicit graphs. We have also shown that an improvement of any of these bounds for explicit graphs would lead to a breakthrough in circuit complexity of boolean functions.

Of particular interest is to prove strong lower bounds on the depth-3 complexity of graphs. Recall that the depth-3 complexity of a graph is the smallest number s such that the graph can be written as an intersection of $\leq s$ graphs, each of which is a union of $\leq s$ bicliques (bipartite complete graphs). Any explicit bipartite $n \times n$ graph requiring $s \geq n^c$ for a constant $c > 0$ would give us the first super-linear lower bound for non-monotone log-depth circuits,

and resolve a 30 years old open problem in circuit complexity. Even a lower bound $s \geq 2^{\alpha\sqrt{\ln n}}$ would break the about 20 years old "square-root barrier" for depth-3 circuits.

A next frontier is to understand the depth-2 complexity of graphs with symmetric output gate. Recall that the symmetric depth-2 complexity of a graph G is the smallest number r for which there exist r bicliques such that no edge and nonedge of G are edges and nonedges of the same number of these bicliques. Any explicit bipartite $n \times n$ graph requiring $r \geq 2^{(\ln \ln n)^{\alpha}}$ bicliques for $\alpha \to \infty$ would resolve yet another old problem in circuit complexity: it would give the the first super-polynomial lower bound for constant-depth circuits with modular gates.

An ultimate goal is to exhibit an explicit $n \times o(n)$ graph requiring $(2+c)n$ union and intersection operations (of fanin 2) to generate it starting from stars, where $c > 0$ is an arbitrary small constant: this would yield an even exponential lower bound for unrestricted circuits. Having proved the existence of such a graph in **NP** we would have proven the inequality $\mathbf{P} \neq \mathbf{NP}$. (Recall that a graph belongs to **NP** if the adjacency in it can be decided by a nondeterministic Turing machine in time polynomial in $\log n$.) The strongest currently known lower bounds for explicit graphs are only of the form $2n-1$, even though almost all graphs require about $n^2/\log n$ operations.

Bibliography

1 N. Alon (1986): Covering graphs by the minimum number of equivalence relations. *Combinatorica* 6, 201–206.

2 N. Alon, M. Karchmer, and A. Wigderson (1990) Linear circuits over GF(2). *SIAM J. Comput.* 19(6), 1064–1067.

3 A. E. Andreev (1986) On a family of boolean matrices. *Moscow Univ. Math. Bull.* 41, 79–82.

4 A. E. Andreev (1987a) On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow Univ. Math. Bull.* 42(1), 63–66.

5 R. Beigel and J. Tarui (1994) On ACC. *Comput. Complexity* 4, 350–366.

6 N. Biggs (1974) *Algebraic graph theory*, Cambridge University Press, London, Cambridge Tracts in Mathematics, No. 67.

7 J. L. Bordewijk (1956) Inter-reciprocity applied to electrical networks. *Appl. Sci. Res. B: Electrophysics, Acoustics, Optics, Mathematical Methods*, 1–74.

8 W. G. Brown (1966) On graphs that do not contain a Thompson graph. *Can. Math. Bull.* 9, 281–285.

9 S. Bublitz (1986) Decomposition of graphs and monotone size of homogeneous functions. *Acta Inform.* 23, 689–696.

10 A. V. Chashkin (1994) On the complexity of Boolean matrices, graphs and their corresponding Boolean functions. *Discrete Math. and Appl.* 4(3), 229 -257.

11 G. Constantine (1990) Graph complexity and the Laplacian matrix in blocked experiments, *Linear and Multilinear Algebra* 28(1-2), 49–56.

12 M. Dehmer and A. Mowshowitz (2011) A history of graph entropy measures, *Information Sciences* 181, 57–78.

13 A. Drucker (2011) Limitations of lower-bound methods for the wire complexity of boolean operators. *El. Colloquium on Comput. Complexity* (ECCC), Report Nr. 125.

14 P. Erdős and R. Rado (1960) Intersection theorems for systems of sets. *J. London Math. Soc.* 35, 85–90.

15 F. Green, J. Köbler, K. W. Regan, T. Schwentick, and J. Toran (1995) The power of the middle bit of a # P function. *J. Comput. Syst. Sci.* 50(3) 456–467.

16 R. Grone and R. Merris (1988), A bound for the complexity of a simple graph. *Discrete Math.* 69(1), 97–99.

17 J. Håstad (1998) The shrinkage exponent is 2. *SIAM J. Comput.* 27, 48–64.

18 S. Jukna (2006) Disproving the single level conjecture. *SIAM J. Comput.* 36(1), 83–98.

19 S. Jukna (2006) On graph complexity. *Combinatorics, Probability and Computing* 15, 855–876.

20 S. Jukna (2009) On set intersection representations of graphs. *J. Graph Theory* 61(1), 55–75.

21 S. Jukna (2010) Representing (0,1)-matrices by depth-2 circuits with arbitrary gates. *Discrete Math.* 310, 184–187.

22 S. Jukna (2012) *Boolean Function Complexity: Advances and Frontiers*, Algorithms and Combinatorics, Vol. 27, Springer-Verlag.

23 N. H. Katz (2012) On the CNF-complexity of bipartite graphs containing no squares. *Lithuanian Math. J.* 52(4), 385–389.

24 V. M. Khrapchenko (1971) A method of obtaining lower bounds for the complexity of π-schemes. *Math. Notes Acad. of Sci. USSR* 10 (1972) 474–479.

25 M. Kneser (1955) Aufgabe 300. *Jahresber. Deutsch. Math.-Verein* 8.

26 J. Kollár, L. Rónyai, and T. Szabó (1996) Norm-graphs and bipartite Turán numbers. *Combinatorica* 16(3), 399–406.

27 J. Körner (1973) Coding of an information source having ambiguous alphabet and the entropy of graphs. *Trans. 6-th Prague Conf. on Information Theory*, Academia, 441-425.

28 R. E. Krichevski (1964) Complexity of contact circuits realizing a function of logical algebra. *Soviet Physiscs Doklady* 8, 770–772.

29 R. Lidl and H. Niederreiter (1986) *Introduction to Finite Fields and their Applications*, Cambridge University Press.

30 S. V. Lokam (2003) Graph complexity and slice functions. *Theory of Comput. Syst.* 36(1), 71–88.

31 L. Lovász (1978) Kneser's conjecture, chromatic numbers and homotopy. *J. Combin. Theory Ser. A* 25, 319–324.

32 O. B. Lupanov (1956) On rectifier and switching-and-rectifier schemes. *Dokl. Akad. Nauk SSSR* 111, 1171–1174 (in Russian).

33 K. Mehlhorn (1979) Some remarks on Boolean sums. *Acta Inform.* 12, 371–375.

34 E. I. Nechiporuk (1969) On topological principles of self-correction. *Problemy Kibernetiki* 21, 5–102 (in Russian).

35 E. I. Nechiporuk (1969) On a Boolean matrix, *Problemy Kibernetiki* 21, 237–240 (in Russian). English transl. in *Systems Theory Res.* 21 (1970), 236-239.

36 I. Newman and A. Wigderson (1995) Lower bounds on formula size of boolean functions using hypergraph-entropy. *SIAM J. Discrete Math.* 8(4), 536–542.

37 D. L. Neel and M. E. Orrison (2006) The linear complexity of a graph, *El. J. of Combinatorics* 13, #R9.

38 V. A. Orlov (1970) Realization of "narrow" matrices by rectifier networks. *Problemy Kibernetiki* 22, 45–52 (in Russian).

39 N. Pippenger (1980) On another boolean matrix. *Theor. Comput. Sci.* 11, 49–56.

40 P. Pudlák and V. Rödl (1994) Some combinatorial-algebraic problems from complexity theory. *Discrete Math.* 136(1-3), 253–279.

41 P. Pudlák, V. Rödl, and P. Savický (1988) Graph Complexity. *Acta Inf.* 25(5), 515-535.

42 A. A. Razborov (1990) Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica* 10(1), 81–93.

43 K. L. Rychkov (1985) A modification of Khrapchenko's method and its application to lower bounds for π-schemes of code functions. *Metody Diskretnogo Analiza*, 42 (Novosibirsk), 91–98 (in Russian).

44 B. A. Subbotovskaya (1961), Realizations of linear functions by formulas using $+, ., -$. *Soviet Math. Dokl.* 2, 110–112.

45 Z. Tuza (1984) Covering of graphs by complete bipartite subgraphs, complexity of 0-1 matrices. *Combinatorica* 4, 111–116.

46 L. G. Valiant (1977) Graph-theoretic methods in low-level complexity. Springer LNCS, vol. 53, 162–176.

47 I. Wegener (1980) A new lower bound on the monotone network complexity of Boolean sums. *Acta Inform.* 15 147–152.

48 A. C. Yao (1990) On ACC and threshold circuits. Proc. of 31th Ann. IEEE Symp. on Foundations of Comput. Sci., 619–627.

Index