TGINF **November 4, 2018**

Graph Complexity

Notes: Chi-Ning Chou

1 Introduction

Graph complexity is a program for proving circuit lower bound with a long history initiated by Pudlák, Rödl,and Savickỳ [PRS88]. The general idea is reducing the task of proving circuit lower bounds to showing the lower bounds for certain graph properties. It is not clear whether this approach makes the lower bound problem even harder or makes our lives better. On the bright side, the strongly exponential lower bound for depth-3 AC^0 circuits with bottom XOR gates had been proved using graph complexity by Jukna [Juk06]. On the other hand, several conjectures in graph complexity that have great consequences in circuit lower bound remain widely open.

In this notes, we focus on graph complexity of bipartite graph and emphasize more on the results on bounded depth circuit models. For interested readers, please refer to Jukna's survey [Juk13] for a more comprehensive study.

The idea of the graph complexity program for proving circuit lower bounds is associating a bipartite graph to a boolean function as follows. Let $m \in \mathbb{N}$ and $n = 2^m$. Consider a bipartite graph G where $V(G) = V_1 \cup V_1$, $V_1 = V_2 = [n]$, and $E(G) \subseteq V_1 \times V_1$. For each $u \in V_1$ (resp. $v \in V_2$), define $\mathbf{x}(u) \in \{0,1\}^m$ (resp. $\mathbf{y}(v)$) be its binary representation. Similarly, for each $\mathbf{x} \in \{0,1\}^m$ (resp. $\mathbf{y}$), define $u(\mathbf{x}) \in V_1$ (resp. $v(\mathbf{y})$) be the corresponding vertex. Now, we can define the *characteristic function* of G as follows. $f_G(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m)$ with the following property.

$$f_G(\mathbf{x}, \mathbf{y}) = 1 \;\Leftrightarrow\; (u(\mathbf{x}), v(\mathbf{y})) \in E(G).$$

Note that the number of variables m of f_G is exponentially smaller than the number of vertices n in G. The gem of graph complexity can be described by the following informal inequality.

$$\text{Circuit-Complexity}(f_G) \geq \text{Graph-Complexity}(G),$$

where Circuit-Complexity(f_G) and Graph-Complexity(G) will be defined later. The tricky point here is that there is a scaling implicitly lies in the above inequality. While the circuit complexity is measured in m and the graph complexity is measured in n, as long as one can get a $n^{\Omega(1)}$ lower bound for G, then a strongly exponential lower bound $n^{\Omega(1)} = 2^{\Omega}(m)$ is obtained for f_G.

To make things concrete, let us start with the definition of graph complexity. For simplicity, in this notes we focus on the *star complexity* of graphs.

1.1 Star complexity

The graph complexity for a bipartite graph G is defined as the difficulty of *representing* G using elementary operations. Here, we say representing a graph G in the sense that there is a way to text

whether a pair of vertices forms an edge in G. Formally, a function $f : V_1 \times V_2 \to \{0,1\}$ represents G if $f(u,v) = 1 \Leftrightarrow (u,v) \in E(G)$. The graph complexity of G is then defined as the complexity of function that represents G.

In the following, we focus on a specific way of representing a bipartite graph with *stars*. A *star* is a graph that consists of a vertex having edges to everyone else and there is no edge between the other vertices. For a bipartite graph on vertex sets V_1, V_2, a star is then a vertex $u \in V_1$ (or $v \in V_2$) having edges to every vertices in V_2 (or V_1). Denote the star centered at vertex u as S_u. See Figure 1.

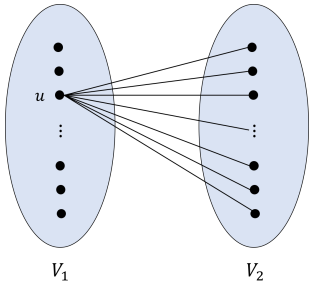

Figure 1: An example of star S_u.

There are $2n$ possible stars in a bipartite graph of n vertices on each side. For $u \in V_1$, we associate a variable $z_u : V_1 \times V_2 \to \{0,1\}$ to the star centered at u such that

$$z_u(u', v') = 1 \;\Leftrightarrow\; (u', v') \in E(S_u) \neq \emptyset. \tag{1.1}$$

Similarly, we can define z_v for $v \in V_2$.

The key idea here is using a function of stars to check whether an edge belongs to $E(G)$. For example, an OR of two stars $z_u \vee z_v$ represents the graph $S_u \cup S_v$ where the union is taken over the edge set. Similarly, $z_u \wedge z_v$ represents $S_u \cap S_v$. Given a graph G, the goal is then finding a function f over the stars such that $f(\mathbf{v}_{(u',v')}) = 1$ if and only if $(u', v') \in E(G)$. We call such f a *star-representation* of G. Formally, the definition is as follows.

Definition 1.2 (Star-representation)**.** *Let G be a bipartite graph with vertex sets V_1 and V_2. We say a function $f : \{0,1\}^{|V_1|+|V_2|} \to \{0,1\}$ star-represents G if for any $u' \in V_1, v' \in V_2$,*

$$f\left(\{z_u(u', v')\}_{u \in V_1}, \{z_v(u', v')\}_{v \in V_2}\right) = 1 \;\Leftrightarrow (u', v') \in E(G).$$

$\Diamond$

For example, the following function star-represents bipartite graph G.

$$f = \left(\bigvee_{u \in V_1} z_u\right) \bigwedge \left(\bigvee_{v \in V_2, (u,v) \in E(G)} z_v\right). \tag{1.3}$$

The *star complexity* of a graph G is then naturally defined as the complexity of the function that star-represents G. Note that the star complexity can be defined with respect to any common circuit classes. Specifically, we are usually interested in monotone circuit classes. That is, circuits with no negation gate.

For instance, we can star-represents every bipartite graph using monotone CNF formula as in Equation 1.3. Thus, it is natural to define the monotone CNF star-complexity of G as follows.

Definition 1.4 (Monotone CNF star-complexity)**.** *Let G be a bipartite graph. The monotone CNF star-complexity of G, denoted as $\mathsf{mcnf}(G)$, is defined the minimum number of clauses of a monotone CNF that star-represents G.* ◊

As we did in Equation 1.3, every graph has monotone CNF star-complexity at most $O(n + E(G)) = O(n^2)$. With some efforts, one can show that for every bipartite graph G, $\mathsf{mcnf}(G) = O(\frac{n^2}{\log n})$ and there exists a bipartite graph G such that $\mathsf{mcnf}(G) = \Omega(\frac{n^2}{\log n})$. See the survey of Jukna [citation] for proofs.

1.2 Magnification lemma

There is a connection between the circuit complexity of f_G with the star-complexity of G through the *magnification lemma*. Specifically, the goal of this subsection is proving the following magnification lemma for CNFs.

Lemma 1.5 (Magnification for CNFs)**.** *Let G be a bipartite graph. We have $\mathsf{CNF\text{-}SIZE}(f_G) \geq \mathsf{mcnf}(G)$.*

Proof of Lemma 1.5. Suppose f_G is computed by a CNF of s clauses. The idea is replacing the each input gate with an OR of star-variables $\{z_u\}_{u \in V_1}$ and $\{z_v\}_{v \in V_2}$.

First, without loss of generality, we can assume the input gates are $x_{i,b}$ for each $i \in [m]$ and $b \in \{0,1\}$ where $x_{i,0} = 1$ (resp. $x_{i,1} = 1$) when $\mathbf{x}_i = 0$ (resp. $\mathbf{x}_i = 1$). As a result, the CNF computing is monotone in these new variables.

Second, observe that the function $f_{i,b} = x_{i,b}$ corresponds to a biclique $G_{i,b}$ where $E(G_{i,b}) = \{(u,v) : \mathbf{x}(u)_i = b,\ v \in V_2\}$. This can be written as an OR of 2^{m-1} many star variables as follows.

$$x_{i,b} = \bigvee_{u \in V_1,\ \mathbf{x}(u)_i = b} z_u.$$

Similar gadget works for $y_{j,b}$ for any $j \in [m]$ and $b \in \{0,1\}$.

Finally, we can simply replace the variables $\mathbf{x}_{i,b}$ and $\mathbf{y}_{j,b}$ in C with ORs of $\{z_u\}_{u \in V_1}$ and $\{z_v\}_{v \in V_2}$. Note that this would not increase the number of clauses in the CNF since the bottom gates are also OR and can be merged with the gadget. Now, we have a CNF of size s star-representing G. That is, $s \geq \mathsf{mcnf}(G)$. See Figure 2.

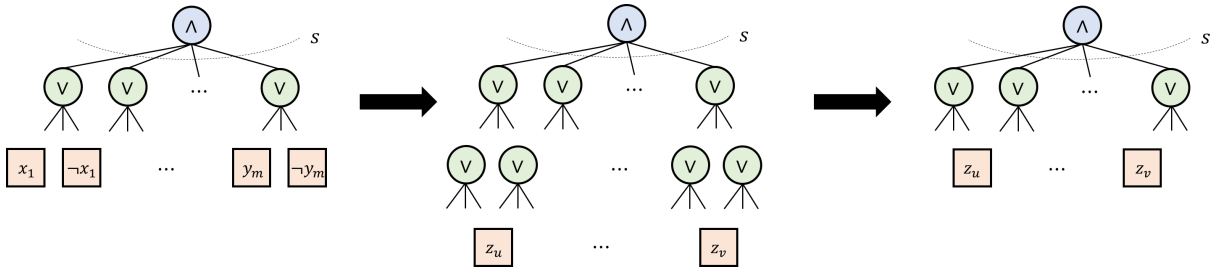

Figure 2: An example of magnification from CNF circuit for f_G to monotone CNF that star-represents G. Note that the magnification does not increase the number of clauses in the CNF.

□

Note that the connection in Lemma 1.5 says that $\mathsf{mcnf}(G) \geq n^\epsilon$ would imply $\mathsf{CNF\text{-}SIZE}(f_G) \geq n^\epsilon = 2^{\epsilon m}$, which is a strong exponential lower bound. Nevertheless, strong exponential lower bound for CNF are already know (for PARITY). In the next subsection, we are going to see how to utilize Lemma 1.5 to get a strong exponential lower bound for depth-3 AC^0.

2 Some known results

Let us start this section with a table of known connection between star complexity and circuit complexity. All the details can be found in [Juk13].

Star Complexity	Circuit Class	Ideal Lower Bound	Consequence	Status
Monotone circuit	Circuit	$(2+\Omega(1))n$	$\mathsf{P} \neq \mathsf{NP}$	?
Monotone formula	Formula	$n \log^{3+\Omega(1)} n$	$n^{3+\Omega(1)}$ formula lb	?
Monotone CNF	CNF	$\Omega(n)$	Strongly exponential CNF lb	Folklore
Monotone $\oplus \circ \vee$	$\oplus \circ \vee$	$\Omega(n)$	Strongly exponential $\oplus\vee$ lb	Folklore
Monotone $SYM \circ \vee$	$SYM \circ \vee$	$2^{\log \log^{\omega(1)} n}$	Super-polynomial $\mathsf{ACC0}$ lb	?
Monotone $\vee \circ \wedge \circ \vee$	$\vee \circ \wedge \circ \vee$	$n^{\Omega(1)}$	Strongly exponential $\vee \circ \wedge \circ \vee$ lb	?
Monotone $\wedge \circ \vee \circ \oplus$	$\wedge \circ \vee \circ \oplus$	$n^{\Omega(1)}$	Strongly exponential $\wedge \circ \vee \circ \oplus$ lb	[Juk06]

Table 1: Connection between star complexity and circuit complexity.

2.1 Depth-3 AC^0 with bottom XOR gates

In this subsection, we consider the depth-3 AC^0 with bottom XOR gates. Concretely, we study function in the following form that star-represents a bipartite graph G.

$$\bigvee_i \bigwedge_j \bigoplus_u z_{i,j,u}. \tag{2.1}$$

We denote the smallest size of the above function star-representing G as $\mathrm{Star}_3^*(G)$. Two immediate facts after the definitions. First, Pudlák and Rödl [PR94] showed that for any bipartite graph G, $\mathrm{Star}_3^*(G) = O(\frac{n}{\log n})$. Second, by the magnification lemma, we have $\oplus_3(f_G) \geq \mathrm{Star}_3^*(G)$ where $\oplus_3(f_G)$ is the smallest size of unbounded fan-in $\vee \circ \wedge \circ \oplus$ circuit for f_G. This circuit class is also known as DNF of parity. Thus, the goal is explicitly finding a bipartite graph G such that $\mathrm{Star}_3^*(G) = n^{\Omega(1)}$. It turns out that there exists such explicit graph found by Jukna [citation].

Theorem 2.2 (informal)**.** *There exists an explicit bipartite graph G such that $\mathrm{Star}_3^*(G) = n^{\Omega(1)}$.*

An immediate corollary of the above theorem is the first strongly exponential lower bound for DNF of parity. To prove Theorem 2.2, we need to first give a combinatorial characterization for $\mathrm{Star}_3^*(G)$.

Definition 2.3 (Fat matching)**.** *A fat matching in a bipartite graph is a family of vertex-disjoint bi-clique. Let $\mathrm{fat}(G)$ be the smallest number of fat matching that covers*[1] *G.* ◊

[1] We say $H_1, \ldots, H_t$ covers G if H_i is a subgraph of G for all $i \in [t]$ and every edge in G is contained in at least one H_i.

Lemma 2.4 (Combinatorial characterization for Star_3^*)**.** *$Star_3^*(G) = fat(G)$.*

Proof of Lemma 2.4. Let us start with the bottom gate of Equation 2.1. Observe that an XOR of stars in a bipartite graph is union of two vertex-disjoint biclique, *a fat matching*. Concretely, $\oplus_i S_{u_i} \oplus_j S_{v_j}$ star-represents $\{(u,v) : \exists i, u = u_i, \forall j, v \neq v_j\} \cup \{(u,v) : \forall i, u \neq u_i, \exists j, v = v_j\}$. As for the middle layer, as AND of fat matching is still fat matching, we know that the bottom two gates star-represents a fat-matching of G. Finally, the top OR gate is simply an union and thus Equation 2.1 gives a fat matching covering for G. □

Lemma 2.5. *For any $a, b \geq 1$ and $K_{a,b}$-free bipartite graph G, $fat(G) \geq \frac{|E(G)|}{(a+b)n}$.*

Proof of Lemma 2.5. The high-level intuition is that $K_{a,b}$-free graph does not have a large fat matching. Concretely, let $H = \cup_{i=1}^{t} A_i \times B_i$ be a fat matching in G, we have

$$|E(H)| = \sum_{i \in [t]} |A_i| \cdot |B_i| \leq \sum_{i:|A_i|<a} a \cdot |B_i| + \sum_{i:|A_i| \geq a} |A_i| \cdot b \leq (a+b)n,$$

where the last inequality is due to the vertex-disjoint property of a fat matching. As each fat matching in G has size at most $(a+b)n$, a fat matching covering for G must be of size at least $\frac{|E(G)|}{(a+b)n}$. □

Theorem 2.2 is then a corollary of the following lemma. We leave the proof to Appendix A.

Lemma 2.6 (Explicit construction of $K_{2,2}$-free graphs with large average degree)**.** *For any prime power q and $n = q^2 + q + 1$, there exists a bipartite graph G of n and n vertices such that G is $K_{2,2}$-free and $(q+1)$-regular. Furthermore, there exists an algorithm that given two distinct vertices u, v of G in bit representation[2], outputs whether $(u,v) \in E(G)$ in $\mathsf{poly}(\log n)$ time.*

3 An approach for $\vee \circ \wedge \circ \vee$

As we saw in Table 1, there are still many open questions in graph complexity. Some immediately seems inapproachable as it implies super strong consequences while some looks innocent. In particular, for the $\vee \circ \wedge \circ \vee$ connections in Table 1, there are explicit conjectures pointing out a path towards proving them. Let us first state the conjecture without context.

Conjecture 3.1. *There exists $\epsilon > 0$ such that for any n large enough and G be the explicit $K_{2,2}$-free $\sqrt{n}$-regular graph from Lemma 2.6. Let $G = H_1 \cup \cdots \cup H_{n^\epsilon}$ be a decomposition of G, there exists $i \in [n^\epsilon]$ such that the star CNF complexity of H_i is $n^{2\epsilon}$.*

Let us formally define the circuit complexity for depth-2 and depth-3 unbounded fan-in circuits.

Definition 3.2. *Let $f : \{0,1\}^m \to \{0,1\}$ be a boolean function. Define*

- $\mathsf{CNF\text{-}SIZE}(f)$ *as the size of the smallest unbounded fan-in circuit with top gate being AND and bottom gates being OR that computes f,*

[2]Since there are $2n$ vertices, they can be represented in $\lceil \log 2n \rceil$ bits.

- Σ_3-SIZE(f) *as the size of the smallest unbounded fan-in circuit with top gate being OR, middle gates being AND, and bottom gates being OR that computes* f*, and*

$\Diamond$

Theorem 3.3 (Conjecture 3.1 $\Rightarrow$ Strong exponential lower bound for depth-3 AC^0)**.** *If Conjecture 3.1 is true. Then, there exists a constant* $\epsilon > 0$ *such that for any* $m \in \mathbb{N}$ *large enough, there is an explicit bipartite graph* G *where* $\Sigma_3\text{-SIZE}(f_G) = 2^{\epsilon m}$*.*

3.1 Monotone CNF star complexity and covering number

A key step towards proving depth-3 lower bound is a connection between $\mathsf{mcnf}(G)$ and the *covering number* of the complement of A_G explained as follows.

Let C be a monotone CNF that star-represents G. C would be in the following form.

$$C = \bigwedge_{\ell \in [t]} \bigvee_{u \in T_\ell} z_u, \tag{3.4}$$

where $T_\ell \subseteq V_1 \cup V_2$ is a subset of vertices. Let us take a closer look at a single clause $\vee_{u \in T_\ell} z_u$. Recall from Equation 1.1 that $z_u(u', v') = 1$ if and only if $(u', v') \in E(S_u)$, where S_u is the star centered at vertex u. Thus, this clause will be evaluated to 1 if and only if $(u', v') \in \cup_{u \in T_\ell} E(N_u)$.

Since C star-represents G, $(u', v') \in E(G)$ if and only if every clause evaluated to 1 on (u', v'). In other words, $(u', v') \notin E(G)$ if and only if $(u', v') \notin \cup_{u \in T_\ell} E(S_u)$ for some $\ell \in [t]$. That is, we have the following two observations.

Lemma 3.5. *Let* G *be a bipartite graph and* $C = \wedge_{\ell \in [t]} \vee_{u \in T_\ell} z_u$ *a monotone CNF that star-represents* G*. Then*

- *each* $\overline{T_\ell}$ *should be an independent set in* G *and*
- *these independent sets* $\overline{T_1}, \overline{T_2}, \ldots, \overline{T_t}$ *cover all the non-edge of* G*.*

Proof of Lemma 3.5. For the first item, let us assume there is an $\ell \in [t]$ such that $\overline{T_\ell}$ is not an independent set in G. That is, there is an edge $(u', v') \in E(G)$ where $u', v' \in \overline{T_\ell}$. However, this means that $u', v' \notin T_\ell$ and thus the clause $\vee_{u \in T_\ell} z_u$ will evaluate to 0 on (u', v'), which contradicts to the fact that C star-represents G.

For the second item, let us also prove by contradiction and assume there is a non-edge $(u', v') \notin E(G)$ that is not covered by $\overline{T_1}, \overline{T_2}, \ldots, \overline{T_t}$. This means that every clause will evaluate to 1 on (u', v') and thus contradicts to the fact that C star-represents G. □

With Lemma 3.5, we are now ready to prove the following main lemma of this subsection connecting the monotone CNF star complexity of G with the covering number of $\overline{A_G}$.

Lemma 3.6. *Let* G *be a bipartite graph. We have*

$$\mathsf{mcnf}(G) = \mathsf{cov}(\overline{A_G}),$$

where A_G *is the adjacency matrix of* G *and* $\overline{A_G}$ *is the flipped of* A_G*. Recall that* $\mathsf{cov}(M)$ *is the smallest number of all one matrices that covers* M*.*

Proof of Lemma 3.6.

- ($\mathsf{mcnf}(G) \geq \mathsf{cov}(\overline{A_G})$) Let C be a monotone CNF of size t that star-represents G. From Lemma 3.5, there are t independent sets $\overline{T_1}, \overline{T_2}, \ldots, \overline{T_t}$ that covers all the non-edge of G. For an independent set $\overline{T_\ell}$, it corresponds to an all one matrix in $\overline{A_G}$. As a non-edge in G corresponds to an 1 in $\overline{A_G}$, this gives us an size t covering for $\overline{A_G}$.
- ($\mathsf{mcnf}(G) \leq \mathsf{cov}(\overline{A_G})$) Let $M_1, M_2, \ldots, M_t$ be a covering for $\overline{A_G}$. As M_ℓ is all one, it corresponds to an independent set $\overline{T_\ell}$ in G. Also, as $M_1, M_2, \ldots, M_t$ form a covering for $\overline{A_G}$, all the non-edges of G are covered. Thus, we can use $\overline{T_1}, \overline{T_2}, \ldots, \overline{T_t}$ to construct a monotone CNF that star-represents G as we did in Equation 3.4 and Lemma 3.5.

□

3.2 Depth-3 AC^0

In this subsection, we are going to use Lemma 1.5 to get a strong exponential lower bound for depth-3 AC^0 from Conjecture 3.1. The candidate graphs are *$K_{2,2}$-free graphs* with $n^{\Omega(1)}$ average degree.

Recall that $K_{2,2}$ is the complete bipartite graph of two vertices on each side. We say a bipartite graph G is $K_{2,2}$-free if G does not contain $K_{2,2}$ as its subgraph. In Lemma 2.6, we saw that there exists explicit construction of bipartite n-vertex D-regular $K_{2,2}$-free graph for $D = \Theta(\sqrt{n})$.

A key property of $K_{2,2}$-free graph is that its subgraph is also $K_{2,2}$-free.

Lemma 3.7. *Let G be a $K_{2,2}$-free graph, then every subgraph of G is also $K_{2,2}$-free.*

Now, we are ready to prove a strong exponential lower bound for depth-3 AC^0 modulo Conjecture 3.1.

Proof of Theorem 3.3. For any $m \in \mathbb{N}$ large enough and $n = 2^m$. Let G be the $K_{2,2}$-free bipartite graph of average degree $D = \Theta(\sqrt{n})$ from Lemma 2.6. Suppose there is a size s depth-3 Σ_3 circuit C computes f_G. Then C is of the following form.

$$C(\mathbf{x}, \mathbf{y}) = \bigvee_{i \in [s]} \phi_i(\mathbf{x}, \mathbf{y}),$$

where ϕ_i is a CNF with at most s clauses for each $i \in [s]$.

A key observation here is that each ϕ_i actually star-represents a subgraph of G. This is because the top get of C is an OR and thus every edge accepted by ϕ_i must also be accepted by C. Next, by averaging argument, we then know that there exists a subgraph H of G with at least $\frac{nD}{2s}$ many edges such that H is represented by ϕ_i for some $i \in [s]$. Finally, by Lemma 3.7, we know that H is still $K_{2,2}$-free and its average degree is at least $\frac{D}{s}$. Thus, we are in the situation where a CNF ϕ_i of at most s clauses that star-represents f_H. Now, apply Lemma 1.5, Lemma 3.6, and Conjecture 3.1 on H, we have

$$\mathsf{CNF\text{-}SIZE}(f_H) \overset{\text{Lemma 1.5}}{\geq} \mathsf{mcnf}(H) \overset{\text{Lemma 3.6}}{=} \mathsf{cov}(\overline{A_H}) \overset{\text{Conjecture 3.1}}{\geq} \frac{n^{2\epsilon}}{s}.$$

As ϕ_i has at most s clauses, $\mathsf{CNF\text{-}SIZE}(f_H) \leq s$ and thus we have the following inequality.

$$s \geq \frac{n^{2\epsilon}}{s}.$$

This gives us $s \geq n^{\epsilon}$. We conclude that $\mathsf{\Sigma_3\text{-}SIZE}(f_G) \geq n^{\Omega(1)} = 2^{\epsilon m}$ for some constant $\epsilon > 0$. □

References

[Juk06] Stasys Jukna. On graph complexity. *Combinatorics, Probability and Computing*, 15(6):855–876, 2006.

[Juk13] Stasys Jukna. Computational complexity of graphs. *Advances in Network Complexity*, pages 99–153, 2013.

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discrete Mathematics*, 1(136):253–279, 1994.

[PRS88] Pavel Pudlák, Vojtěch Rödl, and Petr Savickỳ. Graph complexity. *Acta Informatica*, 25(5):515–535, 1988.

A Explicit construction of $K_{2,2}$-free graphs

In this section, we give a proof for Lemma 2.6. Here, we use the *point-line incidence graph* of projective plane from the external graph theory. Note that there are different ways to construct graphs with the same desiring properties.

Let q be a prime power. The projective plane $PG(2, q)$ is defined as the set of all 1-dimensional subspace in $\mathbb{F}_q^3$. Concretely,

$$PG(2, q) = \{\langle a, b, c\rangle : \ (a, b, c) \in \mathbb{F}_q^3 \backslash \{(0, 0, 0)\}\},$$

where $\langle a, b, c\rangle = \{(\lambda a, \lambda b, \lambda c) : \ \lambda \in \mathbb{F}_q \backslash \{0\}\}$ is the linear span of (a, b, c). As $PG(2, q)$ is a partition for $\mathbb{F}_q^3 \backslash \{(0, 0, 0)\}$ and each linear subspace contains $q - 1$ points, we have

$$|PG(2, q)| = \frac{q^3 - 1}{q - 1} = q^2 + q + 1.$$

Now, define the point-line incidence graph G of $PG(2, q)$ as follows. Let $V_1 = V_2 = PG(2, q)$ and for any $\langle a, b, c\rangle \in V_1$ and $\langle a, b, c\rangle \in V_2$, there is an edge between them if and only if the two linear subspaces are orthogonal to each other. That is,

$$E(G) = \{(\langle a, b, c\rangle, \langle a', b', c'\rangle) \in V_1 \times V_2 : \ aa' + bb' + cc' = 0\}.$$

To see the degree of each vertex in G, fix $\langle a, b, c\rangle \in V_1$. The equation $aa' + bb' + cc' = 0$ has q^2 many solutions (a', b', c'). Note that these solutions include the all zeros vector and a neighbor of $\langle a, b, c\rangle$ would contain $q - 1$ many solutions. Thus, $\langle a, b, c\rangle$ has $\frac{q^2-1}{q-1} = q + 1$ many neighbors. Since $\langle a, b, c\rangle$ is arbitrary, we conclude that G is $(q + 1)$-regular.

Finally, let $n = |PG(2, q)| = q^2 + q + 1$, it is not difficult to see G can be encoded with $\lceil \log 2n \rceil$ bits and checking whether two vertices are neighbors only requires elementary arithmetic operations, which can be done in $\mathsf{poly}(\log n)$ time.