

# Lower bounds for graph reconstruction with maximal independent set queries

Lukas Michel<sup>†</sup>      Alex Scott<sup>†</sup>

4 April 2024

## Abstract

We investigate the number of maximal independent set queries required to reconstruct the edges of a hidden graph. We show that randomised adaptive algorithms need at least  $\Omega(\Delta^2 \log(n/\Delta) / \log \Delta)$  queries to reconstruct  $n$ -vertex graphs of maximum degree  $\Delta$  with success probability at least  $1/2$ , and we further improve this lower bound to  $\Omega(\Delta^2 \log(n/\Delta))$  for randomised non-adaptive algorithms. We also prove that deterministic non-adaptive algorithms require at least  $\Omega(\Delta^3 \log n / \log \Delta)$  queries.

This improves bounds of Konrad, O’Sullivan, and Traistaru, and answers one of their questions. The proof of the lower bound for deterministic non-adaptive algorithms relies on a connection to cover-free families, for which we also improve known bounds.

## 1 Introduction

The graph reconstruction problem is the problem of determining the edges of a hidden graph  $G = (V, E)$  through queries that reveal some partial information about the graph. For a given type of query, the following natural question presents itself.

What is the minimum number of queries required to reconstruct any graph?

This problem has been extensively studied for independent set queries that reveal whether a queried set of vertices contains an edge. Bounds on the minimum number of queries required were studied for general graphs [AC08, AB19], specific families of graphs [AA05, ABK<sup>+</sup>04], and hypergraphs [AC06, ABM14]. In related graph problems, other query models were considered. For instance, maximal matching queries were used to approximate maximum matchings in a graph [KK20, KNS23], and vertex degree queries were used to estimate the average degree of a graph [Fei06, GR08].

Konrad, O’Sullivan, and Traistaru [KOT24] recently investigated graph reconstruction for maximal independent set queries. In this setting, an algorithm can query a set of vertices  $Q \subseteq V$  to obtain a maximal independent set of the subgraph of  $G$  induced by  $Q$ . Using this information, the algorithm has to reconstruct all edges of the graph.

---

<sup>†</sup>Mathematical Institute, University of Oxford, United Kingdom ({michel,scott}@maths.ox.ac.uk). Research of Alex Scott supported by EPSRC grant EP/X013642/1.

There are two important properties that influence the minimum number of queries required by such an algorithm. Firstly, queries can either be adaptive, which means that they depend on the outcomes of other queries, or they are non-adaptive if they do not. While adaptive algorithms require fewer queries, non-adaptive algorithms are desirable because their queries can be performed in parallel. Secondly, algorithms can randomise their queries. In contrast to deterministic algorithms which always reconstruct the graph correctly, randomised algorithms only succeed with high probability, but they might use far fewer queries.

**Randomised algorithms.** One way of choosing non-adaptive queries for a randomised algorithm is to choose each query by independently including each vertex with a fixed probability. By analysing this strategy, Konrad, O’Sullivan, and Traistaru [KOT24] showed that  $\mathcal{O}(\Delta^2 \log n)$  non-adaptive queries suffice to reconstruct graphs with  $n$  vertices and maximum degree  $\Delta$  with high probability.

**Theorem 1.1** (Konrad, O’Sullivan, and Traistaru). *There is a randomised non-adaptive algorithm that uses  $\mathcal{O}(\Delta^2 \log n)$  queries to reconstruct any graph of maximum degree  $\Delta$  with high probability.*

To complement this upper bound, Konrad, O’Sullivan, and Traistaru proved that any randomised adaptive algorithm that succeeds with probability at least  $1/2$  requires at least  $\Omega(\Delta^2 + \log n)$  queries. They asked whether an algorithm could attain this lower bound. We show that this is not possible.

**Theorem 1.2.** *The number of queries of a randomised adaptive algorithm that reconstructs any graph of maximum degree  $\Delta$  with probability at least  $1/2$  is at least*

$$\Omega\left(\frac{\Delta^2 \log\left(\frac{n}{\Delta}\right)}{\log \Delta}\right).$$

For non-adaptive algorithms, we further improve this lower bound by a factor of  $\log \Delta$ . This matches [Theorem 1.1](#) for  $\Delta \leq n^{1-\varepsilon}$  when  $\varepsilon > 0$  is fixed.

**Theorem 1.3.** *The number of queries of a randomised non-adaptive algorithm that reconstructs any graph of maximum degree  $\Delta$  with probability at least  $1/2$  is at least*

$$\Omega\left(\Delta^2 \log\left(\frac{n}{\Delta}\right)\right).$$

The proofs of these lower bounds rely on graphs that contain a clique of size  $\Theta(\Delta)$  while the remaining vertices form an independent set. We will show that maximal independent set queries reveal only very little information about such graphs. Counting the number of these graphs then gives the desired lower bounds.

**Deterministic algorithms.** To obtain a deterministic algorithm, Konrad, O’Sullivan, and Traistaru [KOT24] adapted the idea of their randomised algorithm. They showed that a fixed collection of sufficiently many random queries will reconstruct every graph correctly. This gives a deterministic algorithm with  $\mathcal{O}(\Delta^3 \log n)$  non-adaptive queries.<sup>1</sup>

**Theorem 1.4** (Konrad, O’Sullivan, and Traistaru). *There is a deterministic non-adaptive algorithm that uses  $\mathcal{O}(\Delta^3 \log n)$  queries to reconstruct any graph of maximum degree  $\Delta$ .*

---

<sup>1</sup>In fact, their arguments show that  $\mathcal{O}(\Delta^3 \log(n/\Delta))$  queries are sufficient.

Konrad, O’Sullivan, and Traistaru also showed that every deterministic non-adaptive algorithm needs at least  $\Omega(\Delta^3/(\log \Delta)^2 + \log n)$  queries. We provide a lower bound that matches the upper bound from [Theorem 1.4](#) up to a factor of  $\log \Delta$ .

**Theorem 1.5.** *The number of queries of a deterministic non-adaptive algorithm that reconstructs any graph of maximum degree  $\Delta$  is at least*

$$\Omega\left(\min\left\{n^2, \frac{\Delta^3 \log n}{\log \Delta}\right\}\right).$$

Our proof of [Theorem 1.5](#) relates deterministic non-adaptive algorithms to cover-free families. Here, a family of sets  $\mathcal{F}$  is  *$(w, r)$ -cover-free* if for all  $A_1, \dots, A_w \in \mathcal{F}$  and all  $B_1, \dots, B_r \in \mathcal{F} \setminus \{A_1, \dots, A_w\}$  we have

$$\bigcap_{i=1}^w A_i \not\subseteq \bigcup_{i=1}^r B_i.$$

The size of cover-free families has been widely studied because of their relevance to non-adaptive group testing [[HS87](#), [DH00](#), [DH06](#), [AJS19](#)], key distribution patterns [[MP88](#), [DFFT95](#)], superimposed codes [[KS64](#), [DR82](#)], and more [[Wei06](#), [Eng96](#)]. For a given  $n$ , let  $t(n, w, r)$  denote the minimal  $t$  such that there exists a  $(w, r)$ -cover-free family  $\mathcal{F} \subseteq \mathcal{P}(t)$  with  $n$  sets. We observe the following connection between cover-free families and deterministic non-adaptive algorithms.

**Lemma 1.6.** *The minimum number of queries of a deterministic non-adaptive algorithm that reconstructs any graph of maximum degree  $\Delta$  is at least  $t(n, 2, 2\Delta - 2)$  and at most  $t(n, 2, 2\Delta)$ .*

Upper and lower bounds on  $t(n, w, r)$  have been extensively studied. On the side of upper bounds, it is well-known [[DFFT95](#), [Eng96](#), [STW00](#), [SW04](#)] that

$$t(n, w, r) = \mathcal{O}\left(\frac{(w+r)^{w+r+1}}{w^w r^r} \log n\right).$$

If  $w$  is fixed, this yields  $t(n, w, r) = \mathcal{O}(r^{w+1} \log n)$  which gives an alternative proof of [Theorem 1.4](#).

In terms of lower bounds, when  $w = 1$ , D’yachkov and Rykov [[DR82](#)] proved that  $t(n, 1, r) = \Omega(r^2 \log n / \log r)$  if  $r$  is fixed and  $n \rightarrow \infty$ . Later, Ruszinkó [[Rus94](#)] and Füredi [[Für96](#)] gave purely combinatorial proofs of this result. More generally, for arbitrary  $r$  and  $n$ , the argument of Füredi gives the following lower bound.

**Theorem 1.7** (Füredi). *If  $1 \leq r < n$ , then*

$$t(n, 1, r) = \Omega\left(\min\left\{n, \frac{r^2 \log n}{\log r}\right\}\right).$$

This shows that  $t(n, 1, r) = \Omega(r^2 \log n / \log r)$  for  $r = \mathcal{O}(\sqrt{n})$ , matching the upper bound up to a factor of  $\log r$ , and  $t(n, 1, r) = \Theta(n)$  for  $r = \Omega(\sqrt{n})$ .

For  $w \geq 2$ , previous lower bounds were more complicated. Stinson, Wei, and Zhu [[SWZ00](#)] showed that  $t(n, w, r) = \Omega(r^{w+1} \log n / \log r)$  if  $r$  is fixed and  $n \rightarrow \infty$ . It was

only later determined by Ma and Wei [MW04] that this result holds for  $r = \mathcal{O}(\sqrt{n})$ . For  $r = \Omega(\sqrt{n})$ , Abdi and Bshouty [AB16] proved the best lower bounds known. They showed that  $t(n, w, r) = \Omega(r^{w+1} \log n / \log^{k+1} r)$  for  $\Omega(n^{(k-1)/k}) \leq r \leq \mathcal{O}(n^{k/(k+1)})$  where  $2 \leq k \leq w$ , and  $t(n, w, r) = \Theta(n^w)$  for  $r = \Omega((n \log n)^{w/(w+1)})$ .

Using probabilistic arguments, we generalise [Theorem 1.7](#) to  $w \geq 2$ . This improves the previous lower bounds for  $r = \Omega(\sqrt{n})$ .

**Theorem 1.8.** *If  $w \geq 1$  is fixed and  $1 \leq r < n$ , then*

$$t(n, w, r) = \Omega\left(\min\left\{n^w, \frac{r^{w+1} \log n}{\log r}\right\}\right).$$

This result implies that  $t(n, w, r) = \Omega(r^{w+1} \log n / \log r)$  for  $r = \mathcal{O}(n^{w/(w+1)})$  and  $t(n, w, r) = \Theta(n^w)$  for  $r = \Omega(n^{w/(w+1)})$ . [Theorem 1.5](#) is an immediate consequence of this result in combination with [Lemma 1.6](#).

The rest of the paper is organised as follows. In [Section 2](#) we prove [Theorem 1.8](#), our lower bound for cover-free families. We use this result in [Section 3](#) to obtain a lower bound on the number of queries of deterministic algorithms, proving [Theorem 1.5](#) and [Lemma 1.6](#). In [Section 4](#), we then consider randomised algorithms and prove [Theorems 1.2](#) and [1.3](#). We finish with some open problems in [Section 5](#).

**Notation.** We denote the dual of a family of sets  $\mathcal{F} \subseteq \mathcal{P}(X)$  by  $\mathcal{F}^* = \{\mathcal{F}_x : x \in X\}$  where  $\mathcal{F}_x = \{F \in \mathcal{F} : x \in F\}$ . Moreover, if  $G = (V, E)$  is a graph and  $X \subseteq V$ , then  $G[X]$  is the subgraph of  $G$  induced by  $X$ , and  $N_G(X)$  is the neighbourhood of  $X$  in  $G$ .

## 2 Cover-free families

In this section we prove [Theorem 1.8](#) and show that for any fixed  $w \geq 1$ ,

$$t(n, w, r) = \Omega\left(\min\left\{n^w, \frac{r^{w+1} \log n}{\log r}\right\}\right).$$

We split the proof of this result into two regimes. If  $r = \mathcal{O}(\sqrt{n})$ , we use [Theorem 1.7](#) to show that  $t(n, w, r) = \Omega(r^{w+1} \log n / \log r)$ . We will do this with the following result.

**Lemma 2.1.** *If  $w, r, s \geq 1$  and  $w + r \leq n$ , it holds that*

$$t(n, w + 1, r + s) \geq \left(\frac{w + r}{w}\right)^w \cdot t(n - w - r, 1, s).$$

*Proof.* Let  $\mathcal{F} \subseteq \mathcal{P}(t)$  be a  $(w + 1, r + s)$ -cover-free family with  $n$  sets, and suppose that

$$t < \left(\frac{w + r}{w}\right)^w \cdot t(n - w - r, 1, s).$$

Pick  $A_1, \dots, A_w \in \mathcal{F}$  uniformly and independently at random. Let  $\mathcal{A} = \{A_1, \dots, A_w\}$ . Then, pick  $B_1, \dots, B_r \in \mathcal{F} \setminus \mathcal{A}$  uniformly and independently at random and let  $\mathcal{B} = \{B_1, \dots, B_r\}$ . For any  $x \in [t]$ , if  $\alpha_x = |\mathcal{F}_x|/n$ , we have

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{F}_x, \mathcal{B} \subseteq \mathcal{F}_x^c) \leq \left(\frac{|\mathcal{F}_x|}{n}\right)^w \left(\frac{n - |\mathcal{F}_x|}{n - w}\right)^r = \left(\frac{n}{n - w}\right)^r \alpha_x^w (1 - \alpha_x)^r.$$

Note that the term  $\alpha_x^w(1 - \alpha_x)^r$  is maximised by  $\alpha_x = w/(w + r)$ , and so we get

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{F}_x, \mathcal{B} \subseteq \mathcal{F}_x^c) \leq \left(\frac{n}{n-w}\right)^r \left(\frac{w}{w+r}\right)^w \left(\frac{r}{w+r}\right)^r \leq \left(\frac{w}{w+r}\right)^w,$$

where the last inequality used the fact that  $w + r \leq n$ . Thus, if

$$X = \{x \in [t] : \mathcal{A} \subseteq \mathcal{F}_x, \mathcal{B} \subseteq \mathcal{F}_x^c\},$$

then  $\mathbb{E}(|X|) \leq (w/(w+r))^w \cdot t < t(n-w-r, 1, s)$ , and so with positive probability we have  $|X| < t(n-w-r, 1, s)$ . This implies that the sets  $F \cap X$  with  $F \in \mathcal{F} \setminus (\mathcal{A} \cup \mathcal{B})$  cannot form a  $(1, s)$ -cover-free family<sup>2</sup>, and so there exist  $A_{w+1} \in \mathcal{F} \setminus (\mathcal{A} \cup \mathcal{B})$  and  $C_1, \dots, C_s \in \mathcal{F} \setminus (\mathcal{A} \cup \mathcal{B} \cup \{A_{w+1}\})$  such that

$$A_{w+1} \cap X \subseteq \bigcup_{i=1}^s (C_i \cap X) \subseteq \bigcup_{i=1}^s C_i.$$

Now, suppose that  $x \in \bigcap_{i=1}^{w+1} A_i$ . If  $x \notin X$ , we must have  $x \in \bigcup_{i=1}^r B_i$ . Otherwise  $x \in X$ , so  $x \in A_{w+1} \cap X$  and therefore  $x \in \bigcup_{i=1}^s C_i$ . So,

$$\bigcap_{i=1}^{w+1} A_i \subseteq \bigcup_{i=1}^r B_i \cup \bigcup_{i=1}^s C_i.$$

This contradicts the fact that  $\mathcal{F}$  is  $(w+1, r+s)$ -cover-free.  $\square$

If  $r = \Omega(\sqrt{n})$ , then  $\log r = \Theta(\log n)$ , and so the lower bound we want to show simplifies to  $t(n, w, r) = \Omega(\min\{n^w, r^{w+1}\})$ . Using arguments very similar to the proof of the previous lemma, we show that this is true.

**Lemma 2.2.** *If  $w, r, s \geq 1$  and  $w + r \leq n$ , it holds that*

$$t(n, w, r + s) \geq \min\left\{\frac{1}{2}\left(\frac{n}{w}\right)^w, \frac{s}{2}\left(\frac{w+r}{w}\right)^w\right\}.$$

*Proof.* Let  $\mathcal{F} \subseteq \mathcal{P}(t)$  be a  $(w, r+s)$ -cover-free family with  $n$  sets, and suppose that

$$t < \min\left\{\frac{1}{2}\left(\frac{n}{w}\right)^w, \frac{s}{2}\left(\frac{w+r}{w}\right)^w\right\}.$$

We proceed as in the proof of [Lemma 2.1](#). Pick  $A_1, \dots, A_w \in \mathcal{F}$  uniformly at random and let  $\mathcal{A} = \{A_1, \dots, A_w\}$ . Then, pick  $B_1, \dots, B_r \in \mathcal{F} \setminus \mathcal{A}$  uniformly at random and let  $\mathcal{B} = \{B_1, \dots, B_r\}$ . For any  $x \in [t]$ , we have

$$\mathbb{P}(\mathcal{A} \subseteq \mathcal{F}_x, \mathcal{B} \subseteq \mathcal{F}_x^c) \leq \left(\frac{w}{w+r}\right)^w.$$

Thus, if  $X = \{x \in [t] : \mathcal{A} \subseteq \mathcal{F}_x, \mathcal{B} \subseteq \mathcal{F}_x^c\}$ , then  $\mathbb{E}(|X|) \leq (w/(w+r))^w \cdot t \leq s/2$ , and so  $\mathbb{P}(|X| > s) \leq 1/2$ . Moreover, the event  $\{\mathcal{A} = \mathcal{F}_x\}$  can only happen if  $|\mathcal{F}_x| \leq w$ , and in that case we have  $\mathbb{P}(\mathcal{A} = \mathcal{F}_x) \leq (w/n)^w$ . So,

$$\mathbb{P}(\mathcal{A} = \mathcal{F}_x \text{ for some } x \in [t]) \leq t \cdot \left(\frac{w}{n}\right)^w < \frac{1}{2}.$$

<sup>2</sup>Note that it could happen that  $F \cap X = F' \cap X$  for some distinct  $F, F' \in \mathcal{F} \setminus (\mathcal{A} \cup \mathcal{B})$ . In that case, we could simply pick  $A_{w+1} = F$  and  $C_1 = \dots = C_s = F'$  in the argument that follows.

This implies that with positive probability we have  $|X| \leq s$  and  $\mathcal{A} \neq \mathcal{F}_x$  for all  $x \in [t]$ . Note that for each  $x \in X$  we have  $\mathcal{A} \subseteq \mathcal{F}_x$ , and so there must exist some  $C_x \in \mathcal{F}_x \setminus \mathcal{A}$ .

Now, suppose that  $x \in \bigcap_{i=1}^w A_i$ . If  $x \notin X$ , we must have  $x \in \bigcup_{i=1}^r B_i$ . Otherwise  $x \in X$ , and so  $x \in C_x$ . Therefore,

$$\bigcap_{i=1}^w A_i \subseteq \bigcup_{i=1}^r B_i \cup \bigcup_{x \in X} C_x.$$

This contradicts the fact that  $\mathcal{F}$  is  $(w, r + s)$ -cover-free.  $\square$

Putting these two lemmas together, we obtain the desired result.

*Proof of Theorem 1.8.* If  $r = \mathcal{O}(\sqrt{n})$ , Lemma 2.1 together with Theorem 1.7 implies that

$$t(n, w, r) \geq \left( \frac{w - 1 + \lceil \frac{r}{2} \rceil}{w - 1} \right)^{w-1} \cdot t\left(n + 1 - w - \lceil \frac{r}{2} \rceil, 1, \lfloor \frac{r}{2} \rfloor\right) = \Omega\left(\frac{r^{w+1} \log n}{\log r}\right).$$

Otherwise, if  $r = \Omega(\sqrt{n})$ , Lemma 2.2 shows that

$$t(n, w, r) \geq \min\left\{ \frac{1}{2} \left(\frac{n}{w}\right)^w, \frac{\lfloor \frac{r}{2} \rfloor}{2} \left(\frac{w + \lceil \frac{r}{2} \rceil}{w}\right)^w \right\} = \Omega\left(\min\left\{ n^w, \frac{r^{w+1} \log n}{\log r} \right\}\right). \quad \square$$

### 3 Deterministic non-adaptive algorithms

In this section, we use the lower bound for cover-free families to prove a lower bound on the number of queries of deterministic non-adaptive algorithms. Formally, such an algorithm corresponds to a family of queries  $\mathcal{Q} \subseteq \mathcal{P}(V)$  with the following property: for all distinct graphs  $G$  and  $H$  with maximum degree  $\Delta$  there exists a query  $Q \in \mathcal{Q}$  such that no set of vertices is a maximal independent set of both  $G[Q]$  and  $H[Q]$ . We call such a family  $\mathcal{Q}$  a *query scheme*.

To bound the number of queries in such schemes, we relate them to cover-free families. Our main observation is that a family of queries  $\mathcal{Q} \subseteq \mathcal{P}(V)$  is a query scheme if and only if its dual  $\mathcal{Q}^*$  is  $(2, r)$ -cover-free for  $r \approx 2\Delta$ . This was essentially already proved by Konrad, O'Sullivan, and Traistaru [KOT24], but with different terminology. We provide a proof of the direction necessary for the lower bound in Lemma 1.6.

**Lemma 3.1.** *If  $\mathcal{Q} \subseteq \mathcal{P}(V)$  is a query scheme, then  $\mathcal{Q}^*$  is  $(2, 2\Delta - 2)$ -cover-free.*

*Proof.* If  $\mathcal{Q}^*$  is not  $(2, 2\Delta - 2)$ -cover-free, there exist two distinct vertices  $u, v \in V$  and a set of vertices  $W \subseteq V \setminus \{u, v\}$  with  $|W| \leq 2\Delta - 2$  such that  $\mathcal{Q}_u \cap \mathcal{Q}_v \subseteq \bigcup_{w \in W} \mathcal{Q}_w$ . Let  $G$  be a graph with the least number of edges that has maximum degree  $\Delta - 1$  and satisfies  $N_G(\{u, v\}) = W$ . Let  $H = G \cup \{uv\}$ .

For every  $Q \in \mathcal{Q}$ , let  $I_Q$  be a maximal independent set of  $G[Q]$  with  $Q \setminus \{u, v\} \subseteq I_Q$ . We claim that  $I_Q$  is also a maximal independent set of  $H[Q]$ . Indeed, otherwise we must have  $u, v \in I_Q \subseteq Q$ . Then,  $Q \in \mathcal{Q}_u \cap \mathcal{Q}_v \subseteq \bigcup_{w \in W} \mathcal{Q}_w$ , and so  $W \cap Q \neq \emptyset$ . This implies that

$$N_G(\{u, v\}) \cap I_Q = W \cap I_Q \supseteq W \cap (Q \setminus \{u, v\}) = W \cap Q \neq \emptyset.$$

Since  $u, v \in I_Q$ , this contradicts the fact that  $I_Q$  is an independent set of  $G$ . So, for all  $Q \in \mathcal{Q}$ ,  $I_Q$  is a maximal independent set of both  $G[Q]$  and  $H[Q]$ . This contradicts the fact that  $\mathcal{Q}$  is a query scheme.  $\square$

Using arguments from Konrad, O'Sullivan, and Traistaru [KOT24], we can obtain the following converse.

**Lemma 3.2.** *Let  $\mathcal{Q} \subseteq \mathcal{P}(V)$ . If  $\mathcal{Q}^*$  is  $(2, 2\Delta)$ -cover-free, then  $\mathcal{Q}$  is a query scheme.*

Using these two results, we can now prove Lemma 1.6. Together with Theorem 1.8, this proves Theorem 1.5.

*Proof of Lemma 1.6.* Let  $\mathcal{Q} \subseteq \mathcal{P}(V)$  be a query scheme. Then, by Lemma 3.1, we know that  $\mathcal{Q}^* \subseteq \mathcal{P}(\mathcal{Q})$  is  $(2, 2\Delta - 2)$ -cover-free, and so

$$|\mathcal{Q}| \geq t(|\mathcal{Q}^*|, 2, 2\Delta - 2) = t(n, 2, 2\Delta - 2).$$

On the other hand, if  $t = t(n, 2, 2\Delta)$ , there exist a family  $\mathcal{R} \subseteq \mathcal{P}(t)$  with  $|\mathcal{R}| = n$  that is  $(2, 2\Delta)$ -cover-free. By identifying  $\mathcal{R}$  with  $V$ , we obtain a family  $\mathcal{Q} = \mathcal{R}^* \subseteq \mathcal{P}(V)$  with  $|\mathcal{Q}| = t$  such that  $\mathcal{Q}^* \cong \mathcal{R}$  is  $(2, 2\Delta)$ -cover-free. Then, by Lemma 3.2,  $\mathcal{Q}$  is a query scheme with  $t = t(n, 2, 2\Delta)$  queries.  $\square$

## 4 Randomised algorithms

Finally, we want to lower bound the number of queries of randomised algorithms, both in the adaptive and non-adaptive setting. We begin with some general observations.

First, note that any randomised algorithm  $\mathcal{A}$  can randomly decide in advance for every possible execution path which set of vertices it wants to query next. This gives a deterministic strategy  $S$  that the algorithm will then perform to reconstruct the graph. From now on, we will assume that every algorithm is of this form. That is,  $\mathcal{A}$  randomly selects in advance a deterministic strategy  $S \in \mathcal{S}$  from some family of strategies  $\mathcal{S}$ , and it will then execute this strategy to reconstruct the graph.

The following lemma tells us that by analysing the strategies  $S \in \mathcal{S}$  separately, we can show that there is a graph that will likely be incorrectly reconstructed by  $\mathcal{A}$ .

**Lemma 4.1.** *Let  $\mathcal{G}$  be a family of graphs and  $\mathcal{S}$  be a family of strategies of a randomised algorithm  $\mathcal{A}$ . If  $G \in \mathcal{G}$  is a random graph, then there exists a graph  $H \in \mathcal{G}$  such that*

$$\mathbb{P}(\mathcal{A} \text{ reconstructs } H) \leq \sup_{S \in \mathcal{S}} \mathbb{P}(S \text{ reconstructs } G).$$

*Proof.* Let  $p = \sup_{S \in \mathcal{S}} \mathbb{P}(S \text{ reconstructs } G)$ . Denote the indicator variable of an event  $E$  by  $\mathbb{1}_E$ . Then, it holds that

$$\begin{aligned} \mathbb{E}\left(\mathbb{1}_{\{\mathcal{A} \text{ reconstructs } G\}}\right) &= \sum_{S \in \mathcal{S}} \mathbb{P}(\mathcal{A} \text{ selects } S) \cdot \mathbb{E}\left(\mathbb{1}_{\{S \text{ reconstructs } G\}}\right) \\ &= \sum_{S \in \mathcal{S}} \mathbb{P}(\mathcal{A} \text{ selects } S) \cdot \mathbb{P}(S \text{ reconstructs } G) \\ &\leq \sum_{S \in \mathcal{S}} \mathbb{P}(\mathcal{A} \text{ selects } S) \cdot p = p. \end{aligned}$$

In particular, there exists a graph  $H \in \mathcal{G}$  such that

$$\mathbb{P}(\mathcal{A} \text{ reconstructs } H) = \mathbb{E}\left(\mathbb{1}_{\{\mathcal{A} \text{ reconstructs } H\}}\right) \leq p. \quad \square$$

## 4.1 Randomised adaptive algorithms

We start by providing a lower bound on the number of queries of a randomised adaptive algorithm. For this, we will consider the family of graphs that contain a clique  $U$  of size  $\Theta(\Delta)$  while  $V \setminus U$  is an independent set.

Then, for every query  $Q \subseteq V$  of the algorithm, either  $Q \cap (V \setminus U)$  is a maximal independent set, or there is a vertex  $u \in Q \cap U$  such that  $(Q \cap (V \setminus U)) \cup \{u\}$  is a maximal independent set. So, the answer to a query will always be one out of at most  $\Theta(\Delta)$  many alternatives. This implies that with  $t$  queries, the algorithm can correctly reconstruct at most  $\mathcal{O}(\Delta^t)$  many graphs. By counting the number of graphs in our family, this yields the desired lower bound on  $t$ .

*Proof of Theorem 1.2.* Let  $U \subseteq V$  be a set with  $|U| = \lceil \Delta/2 \rceil$  and let  $\mathcal{G}$  be the family of graphs  $G$  with maximum degree  $\Delta$  such that  $U$  is a clique of  $G$  and  $V \setminus U$  is an independent set of  $G$ . Since every vertex of  $U$  can have up to  $\Delta - (|U| - 1)$  neighbours in  $V \setminus U$ , we get

$$|\mathcal{G}| \geq \binom{n - |U|}{\Delta - (|U| - 1)}^{|U|} \geq \binom{n - \Delta}{\lceil \frac{\Delta}{2} \rceil}^{\lceil \frac{\Delta}{2} \rceil} \geq \left(\frac{n - \Delta}{\lceil \frac{\Delta}{2} \rceil}\right)^{\lceil \frac{\Delta}{2} \rceil^2} \geq \left(\frac{n - \Delta}{\Delta}\right)^{\frac{\Delta^2}{4}}.$$

Now, let  $\mathcal{S}$  be the family of strategies of a randomised adaptive algorithm  $\mathcal{A}$  with success probability at least  $1/2$ , and suppose that  $\mathcal{A}$  makes at most  $t$  queries. Pick a graph  $G \in \mathcal{G}$  uniformly at random, and fix a strategy  $S \in \mathcal{S}$ .

For every possible query  $Q \subseteq V$ , note that either  $Q \cap (V \setminus U)$  is a maximal independent set of  $G[Q]$ , or there is a vertex  $u \in Q \cap U$  such that  $(Q \cap (V \setminus U)) \cup \{u\}$  is a maximal independent set of  $G[Q]$ . So, when executing  $S$  on graphs from  $\mathcal{G}$ , the answer to each query of  $S$  is always one out of  $|Q \cap U| + 1 \leq \Delta + 1$  different sets. Since  $S$  is deterministic and makes at most  $t$  queries, it follows that  $S$  will correctly reconstruct at most  $(\Delta + 1)^t$  graphs from  $\mathcal{G}$ . Therefore,

$$\mathbb{P}(S \text{ reconstructs } G) \leq \frac{(\Delta + 1)^t}{|\mathcal{G}|}.$$

Given that the success probability of  $\mathcal{A}$  is at least  $1/2$ , it follows by Lemma 4.1 that  $(\Delta + 1)^t / |\mathcal{G}| \geq 1/2$ . If  $\Delta \leq n/3$ , this implies that

$$t \geq \frac{\log\left(\frac{|\mathcal{G}|}{2}\right)}{\log(\Delta + 1)} \geq \frac{\Delta^2 \log\left(\frac{n - \Delta}{\Delta}\right) - 4 \log 2}{4 \log(\Delta + 1)} = \Omega\left(\frac{\Delta^2 \log\left(\frac{n}{\Delta}\right)}{\log \Delta}\right).$$

For  $\Delta \geq n/3$ , we can simply apply this argument with  $\Delta = n/3$  to obtain the desired lower bound.  $\square$



## 4.2 Randomised non-adaptive algorithms

To obtain a lower bound on the number of queries of a randomised non-adaptive algorithm, we adapt the arguments of the proof from the last section. In this proof, recall that the answer to a query  $Q$  of the algorithm is always one out of  $|Q \cap U| + 1$  many alternatives. We used that this is at most  $\Delta + 1$ , but if  $Q \cap U$  is much smaller than this for most queries, then we should get a better lower bound on the number of queries.

For adaptive algorithms, we cannot hope for this to be true because such algorithms can easily identify  $U$  with the first query and then always ensure that  $U \subseteq Q$ . For non-adaptive algorithms, however, a randomly chosen  $U$  will usually have a very small intersection with a query  $Q$  of the algorithm, unless  $Q$  is large. But if  $Q$  is large, then most vertices of  $U$  will have a neighbour in  $Q$  which again restricts the number of possible answers to the query. Using this idea, we can improve the lower bound on the number of queries for non-adaptive algorithms by a factor of  $\log \Delta$ .

*Proof of Theorem 1.3.* For disjoint sets  $U, W \subseteq V$  with  $|U| = \lceil \Delta/3 \rceil$  and  $|W| = \lfloor \Delta/3 \rfloor$ , let  $\mathcal{G}_{U,W}$  be the family of graphs  $G$  with maximum degree  $\Delta$  such that  $U$  is a clique of  $G$ ,  $V \setminus U$  is an independent set of  $G$ , and all vertices in  $U$  are connected to all vertices in  $W$ . Let  $\mathcal{G}$  be the union of all the families  $\mathcal{G}_{U,W}$ . Note that

$$N = |\mathcal{G}_{U,W}| \geq \binom{n - |U| - |W|}{\Delta - (|U| - 1) - |W|}^{|U|} \geq \binom{n - \Delta}{\lceil \frac{\Delta}{3} \rceil}^{\lceil \frac{\Delta}{3} \rceil} \geq \left( \frac{n - \Delta}{\Delta} \right)^{\frac{\Delta^2}{9}}.$$

Let  $\mathcal{S}$  be the family of strategies of a randomised non-adaptive algorithm  $\mathcal{A}$  with success probability at least  $1/2$ , and suppose that  $\mathcal{A}$  makes at most  $t$  queries. Pick a graph  $G \in \mathcal{G}$  at random by first picking the sets  $U, W \subseteq V$  uniformly at random and then picking  $G \in \mathcal{G}_{U,W}$  uniformly at random. Fix a strategy  $S \in \mathcal{S}$ . Since  $\mathcal{A}$  is non-adaptive, the strategy  $S$  corresponds of a family of queries  $\mathcal{Q} \subseteq \mathcal{P}(V)$  with  $|\mathcal{Q}| \leq t$ .

For every query  $Q \in \mathcal{Q}$ , note that  $Q \cap (V \setminus U)$  is a maximal independent set of  $G[Q]$  whenever  $Q \cap W \neq \emptyset$ . Otherwise, either  $Q \cap (V \setminus U)$  or  $(Q \cap (V \setminus U)) \cup \{u\}$  for some  $u \in Q \cap U$  is a maximal independent set of  $G[Q]$ . So, when executing  $S$  on graphs from  $\mathcal{G}_{U,W}$ , the answer to each query  $Q \in \mathcal{Q}$  is always one out of  $D_Q$  different sets, where  $D_Q = 1$  if  $Q \cap W \neq \emptyset$  and  $D_Q = |Q \cap U| + 1$  otherwise. It follows that,  $S$  will correctly reconstruct at most  $D = \prod_{Q \in \mathcal{Q}} D_Q$  graphs from  $\mathcal{G}_{U,W}$ . Therefore, for any  $M \geq 0$ ,

$$\mathbb{P}(S \text{ reconstructs } G \mid D \leq M) \leq \frac{M}{N}.$$

Note that

$$D_Q = 1 + \sum_{u \in U} \mathbb{1}_{\{u \in Q, W \subseteq Q^c\}}.$$

Therefore,

$$\mathbb{E}(\log D_Q) \leq \mathbb{E}(D_Q) \leq 1 + \sum_{u \in U} \mathbb{E}(\mathbb{1}_{\{u \in Q, W \subseteq Q^c\}}) = 1 + \sum_{u \in U} \mathbb{P}(u \in Q, W \subseteq Q^c).$$

Let  $\alpha_Q = |Q|/n$ . Since  $U$  and  $W$  are chosen uniformly at random, we have

$$\mathbb{P}(u \in Q, W \subseteq Q^c) \leq \frac{|Q|}{n} \left( \frac{n - |Q|}{n - 1} \right)^{|W|} = \left( \frac{n}{n - 1} \right)^{|W|} \alpha_Q (1 - \alpha_Q)^{|W|}.$$

The term  $\alpha_Q(1 - \alpha_Q)^{|W|}$  is maximised by  $\alpha_Q = 1/(|W| + 1)$ , and so we get

$$\mathbb{P}(u \in Q, W \subseteq Q^c) \leq \left(\frac{n}{n-1}\right)^{|W|} \frac{1}{|W|+1} \left(\frac{|W|}{|W|+1}\right)^{|W|} \leq \frac{1}{|W|+1} \leq \frac{3}{\Delta}.$$

It follows that  $\mathbb{E}(\log D_Q) \leq 1 + 3|U|/\Delta \leq 4$  and therefore, by Markov's inequality,

$$\mathbb{P}(D \geq e^{16t}) = \mathbb{P}\left(\sum_{Q \in \mathcal{Q}} \log D_Q \geq 16t\right) \leq \frac{\mathbb{E}(\sum_{Q \in \mathcal{Q}} \log D_Q)}{16t} \leq \frac{4t}{16t} = \frac{1}{4}.$$

Overall, we get

$$\begin{aligned} \mathbb{P}(S \text{ reconstructs } G) &\leq \mathbb{P}(D \geq e^{16t}) + \mathbb{P}(S \text{ reconstructs } G \mid D \leq e^{16t}) \mathbb{P}(D \leq e^{16t}) \\ &\leq \frac{1}{4} + \frac{e^{16t}}{N}. \end{aligned}$$

Given that the success probability of  $\mathcal{A}$  is at least  $1/2$ , it follows by [Lemma 4.1](#) that  $e^{16t}/N \geq 1/4$ . If  $\Delta \leq n/3$ , this implies that

$$t \geq \frac{1}{16} \log\left(\frac{N}{4}\right) \geq \frac{\Delta^2}{144} \log\left(\frac{n-\Delta}{\Delta}\right) - \frac{1}{16} \log 4 = \Omega\left(\Delta^2 \log\left(\frac{n}{\Delta}\right)\right).$$

For  $\Delta \geq n/3$ , we can simply apply this argument with  $\Delta = n/3$  to obtain the desired lower bound.  $\square$

## 5 Open problems

In this paper, we improved the lower bounds on the number of queries required for graph reconstruction with maximal independent set queries. However, some gaps between the lower and upper bounds remain.

For deterministic non-adaptive algorithms, we showed that  $\Omega(\Delta^3 \log n / \log \Delta)$  queries are necessary while the upper bound  $\mathcal{O}(\Delta^3 \log n)$  was given by Konrad, O'Sullivan, and Traistaru [[KOT24](#)]. This is the same gap as the longstanding gap between the lower and upper bounds  $\Omega(r^{w+1} \log n / \log r) \leq t(n, w, r) \leq \mathcal{O}(r^{w+1} \log n)$  for cover-free families.

**Problem 5.1.** *Are there  $(1, r)$ -cover-free families  $\mathcal{F} \subseteq \mathcal{P}(t)$  with  $t \in \mathcal{O}(r^2 \log |\mathcal{F}| / \log r)$ ?*

For randomised non-adaptive algorithms, the upper bound of Konrad, O'Sullivan, and Traistaru [[KOT24](#)] is  $\mathcal{O}(\Delta^2 \log n)$ . We proved that such algorithms need at least  $\Omega(\Delta^2 \log(n/\Delta))$  queries, and ask whether this is tight.

**Problem 5.2.** *Is there a randomised non-adaptive algorithm which uses  $\mathcal{O}(\Delta^2 \log(n/\Delta))$  queries to reconstruct any graph of maximum degree  $\Delta$  with high probability?*

Finally, for adaptive algorithms, we have the same upper bounds as for non-adaptive algorithms in both the deterministic and randomised setting. However, our best lower bound drops to  $\Omega(\Delta^2 \log(n/\Delta) / \log \Delta)$  in both settings. This suggests that adaptive algorithms might perform better than their non-adaptive counterparts.

**Problem 5.3.** *Is there a deterministic adaptive algorithm that uses  $o(\Delta^3 \log(n/\Delta))$  queries to reconstruct any graph of maximum degree  $\Delta$ ?*

## References

- [AA05] NOGA ALON and VERA ASODI (2005). [Learning a hidden subgraph](#). *SIAM Journal on Discrete Mathematics* **18**(4), 697–712. [↑1](#)
- [AB16] ALI Z. ABDI and NADER H. BSHOUTY (2016). [Lower bounds for cover-free families](#). *Electronic Journal of Combinatorics* **23**(2), Paper 2.45, 7. [↑4](#)
- [AB19] HASAN ABASI and NADER H. BSHOUTY (2019). [On learning graphs with edge-detecting queries](#). *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, PMLR, vol. 98, 3–30. [↑1](#)
- [ABK<sup>+</sup>04] NOGA ALON, RICHARD BEIGEL, SIMON KASIF, STEVEN RUDICH, and BENNY SUDAKOV (2004). [Learning a hidden matching](#). *SIAM Journal on Computing* **33**(2), 487–501. [↑1](#)
- [ABM14] HASAN ABASI, NADER H. BSHOUTY, and HANNA MAZZAWI (2014). [On exact learning monotone DNF from membership queries](#). *Algorithmic learning theory, Lecture Notes in Computer Science*, vol. 8776, 111–124. [↑1](#)
- [AC06] DANA ANGLUIN and JIANG CHEN (2006). [Learning a hidden hypergraph](#). *Journal of Machine Learning Research* **7**, 2215–2236. [↑1](#)
- [AC08] DANA ANGLUIN and JIANG CHEN (2008). [Learning a hidden graph using  \$O\(\log n\)\$  queries per edge](#). *Journal of Computer and System Sciences* **74**(4), 546–556. [↑1](#)
- [AJS19] MATTHEW ALDRIDGE, OLIVER JOHNSON, and JONATHAN SCARLETT (2019). Group testing: an information theory perspective. *Foundations and Trends in Communications and Information Theory* **15**(3-4), 196–392. [↑3](#)
- [DFFT95] MARTIN DYER, TREVOR FENNER, ALAN FRIEZE, and ANDREW THOMASON (1995). [On key storage in secure networks](#). *Journal of Cryptology* **8**, 189–200. [↑3](#)
- [DH00] DING-ZHU DU and FRANK K. HWANG (2000). Combinatorial group testing and its applications, *Series on Applied Mathematics*, vol. 12. Second edn. (World Scientific Publishing Co., Inc., River Edge, NJ). [↑3](#)
- [DH06] DING-ZHU DU and FRANK K. HWANG (2006). [Pooling designs and nonadaptive group testing](#), *Series on Applied Mathematics*, vol. 18 (World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ). Important tools for DNA sequencing. [↑3](#)
- [DR82] A. G. D’YACHKOV and V. V. RYKOV (1982). Bounds on the length of disjunctive codes. *Problems of Information Transmission* **18**(3), 166–171. [↑3](#)
- [Eng96] KONRAD ENGEL (1996). [Interval packing and covering in the Boolean lattice](#). *Combinatorics, Probability and Computing* **5**(4), 373–384. [↑3](#)
- [Fei06] URIEL FEIGE (2006). [On sums of independent random variables with unbounded variance and estimating the average degree in a graph](#). *SIAM Journal on Computing* **35**(4), 964–984. [↑1](#)
- [Für96] ZOLTÁN FÜREDI (1996). [On  \$r\$ -cover-free families](#). *Journal of Combinatorial Theory, Series A* **73**(1), 172–173. [↑3](#)
- [GR08] ODED GOLDBREICH and DANA RON (2008). [Approximating average parameters of graphs](#). *Random Structures & Algorithms* **32**(4), 473–493. [↑1](#)
- [HS87] F. K. HWANG and V. T. SÓS (1987). Non-adaptive hypergeometric group testing. *Studia Scientiarum Mathematicarum Hungarica* **22**, 257–263. [↑3](#)
- [KK20] LIDIYA KHALIDAH BINTI KHALIL and CHRISTIAN KONRAD (2020). [Constructing large matchings via query access to a maximal matching oracle](#). *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 182, Art. No. 26, 15. [↑1](#)

- [KNS23] CHRISTIAN KONRAD, KHEERAN K. NAIDU, and ARUN STEWARD (2023). [Maximum matching via maximal matching queries](#). *40th International Symposium on Theoretical Aspects of Computer Science, Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 254, Art. No. 41, 22. [↑1](#)
- [KOT24] CHRISTIAN KONRAD, CONOR O’SULLIVAN, and VICTOR TRAISTARU (2024). [Graph Reconstruction via MIS Queries](#). arXiv:2401.05845. [↑1, 2, 6, 7, 10](#)
- [KS64] WILLIAM KAUTZ and ROY SINGLETON (1964). Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory* **10**(4), 363–377. [↑3](#)
- [MP88] CHRIS J. MITCHELL and FRED C. PIPER (1988). [Key storage in secure networks](#). *Discrete Applied Mathematics* **21**(3), 215–228. [↑3](#)
- [MW04] X. MA and R. WEI (2004). [On a bound of cover-free families](#). *Designs, Codes and Cryptography* **32**(1-3), 303–321. [↑4](#)
- [Rus94] MIKLÓS RUSZINKÓ (1994). [On the upper bound of the size of the  \$r\$ -cover-free families](#). *Journal of Combinatorial Theory. Series A* **66**(2), 302–310. [↑3](#)
- [STW00] D. R. STINSON, TRAN VAN TRUNG, and R. WEI (2000). [Secure frameproof codes, key distribution patterns, group testing algorithms and related structures](#). *Journal of Statistical Planning and Inference* **86**(2), 595–617. [↑3](#)
- [SW04] D. R. STINSON and R. WEI (2004). [Generalized cover-free families](#). *Discrete Mathematics* **279**(1-3), 463–477. [↑3](#)
- [SWZ00] D. R. STINSON, R. WEI, and L. ZHU (2000). [Some new bounds for cover-free families](#). *Journal of Combinatorial Theory. Series A* **90**(1), 224–234. [↑3](#)
- [Wei06] RUIZHONG WEI (2006). [On cover-free families](#). arXiv:2303.17524. [↑3](#)