Comparing the strength of query types in property testing: The case of testing k-colorability

Ido Ben-Eliezer [*] Tali Kaufman [†] Michael Krivelevich [‡] Dana Ron [§]

Abstract

We study the power of four query models in the context of property testing in general graphs, where our main case study is the problem of testing k-colorability. Two query types, which have been studied extensively in the past, are *pair queries* and *neighbor queries*. The former corresponds to asking whether there is an edge between any particular pair of vertices, and the latter to asking for the i'th neighbor of a particular vertex. We show that while for pair queries, testing k-colorability requires a number of queries that is a monotone decreasing function in the average degree d, the query complexity in the case of neighbor queries remains roughly the same for every density and for large values of k. We also consider a combined model that allows both types of queries, and we propose a new, stronger, query model, which is related to the field of Group Testing. We give one-sided error upper and lower bounds for all the models, where the bounds are nearly tight for three of the models. In some of the cases our lower bounds extend to two-sided error algorithms.

The problem of testing k-colorability was previously studied in the contexts of dense and sparse graphs, and in our proofs we unify approaches from those cases, and also provide some new tools and techniques which may be of independent interest.

1 Introduction

Property testing [12, 17] deals with the problem of deciding if a certain object has a prespecified property P or is far (i.e., differs significantly) from any object that has P. For a given distance parameter ϵ, the algorithm should *accept* graphs that have the property, and should *reject* graphs that are ϵ-far from having the property with respect to some predetermined distance measure. The algorithm is given query access to the object, and it should make the decision after observing only a small part of the object. We consider randomized algorithms that are allowed a small constant error probability. If the algorithm always accepts graphs that satisfy P then it has *one-sided error*, otherwise it has *two-sided error*. Property testing has been studied in recent years in many contexts, including graphs, boolean functions and geometric problems (see [11, 16] for surveys). In this work our interest lies in comparing the power of different query types with the same distance measure. To this end, we compare the power of the models for the problem of testing k-colorability for $k \geq 3$, which was previously studied in the dense and sparse contexts but not in general graphs.

1.1 Graph Testing Models and Query Types.

Property testing of graphs has previously been studied in several models. The models differ in two (quite related) aspects. The first is the distance measure between graphs, which determines the meaning of being ϵ-far from having the property. The second is the type of queries that the algorithm is allowed. In what follows, let n be the number of vertices in the graph.

The dense model. In this model (first studied in [12]) it is assumed that the graph is represented by its $n \times n$ adjacency matrix. A testing algorithm may perform queries into the matrix. Thus, queries are of the form "is there an edge between the pair of vertices u and v". We refer to these queries as *pair queries*. In this model the distance between graphs is the fraction of entries on which their adjacency matrices differ. A graph is ϵ-far from having a property P if it is necessary to make more than $\epsilon \cdot n^2$ modifications in the matrix to create a graph having the property.

The bounded-degree model. In this model (first studied in [13]) it is assumed that the graph is represented by incidence lists of bounded length $d_{\max}$, where $d_{\max}$ is the maximum degree in the graph. In this model queries are of the form: "who is the i'th neighbor of vertex v", and we refer to these queries as *neighbor*

[*] School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: idobene@post.tau.ac.il.

[†] Institute for Advanced Study, Princeton, New Jersey. E-mail:kaufmant@ias.edu.

[‡] School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@post.tau.ac.il. Research supported in part by a USA-Israel BSF grant and by a grant from the Israel Science Foundation.

[§] School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: danar@eng.tau.ac.il. Research supported in part by a grant from the Israel Science Foundation.

queries[1]. A graph is ϵ-far from having a property P if it is necessary to make more than $\epsilon d_{\max} n$ modifications in the incidence lists to create a graph having the property.

The combined model. The first model described above is clearly appropriate for testing dense graphs and the second model for testing sparse, bounded-degree graphs. When dealing with general graphs (that may be sparse, but with varying degrees, or may be neither dense nor sparse), a different model is required. Let d be the average degree in the graph. It was suggested in [15] to define the distance that a graph has to a property simply as the fraction of necessary edge modifications taken with respect to $d \cdot n$ (which is roughly the number of edges in the graph). This distance measure does not depend on the graph representation. In [14] it was suggested to allow the algorithm to use both pair queries and neighbor queries, as well as *degree queries* ("what is the degree of a vertex v").[2]

A new query type: The group query. Here we suggest a new query type that extends pair queries. The queries are of the form "is there at least one edge between a vertex u and a set of vertices S". We refer to these queries as *group queries*. The study of such queries are motivated by the field of Group Testing (see, e.g., [9]), where similar queries are allowed. Problems of group testing can be found in various fields such as Statistics and Biology. Neighbor queries can also be emulated using group queries (see Appendix A), and as we will discuss later, this model is strictly stronger than the combined model.

1.2 Related Work. The problem of testing k-colorability was previously studied in the dense model and in the bounded-degree model. In the dense model, Rödl and Duke [10] proved implicitly that k-colorability is testable using a number of queries that is independent of the graph size, but is a tower function of $1/\epsilon$. Goldreich et al. [12] proved that it suffices to sample an induced subgraph on $\tilde{O}(k^2/\epsilon^3)$ uniformly selected vertices, and Alon and Krivelevich [4] proved that it suffices to sample a subgraph on $\tilde{O}(k/\epsilon^2)$ vertices.

In the bounded-degree model, Bogdanov et. al. [7] proved that in order to test 3-colorability, it is necessary to perform $\Omega(n)$ queries. Recently it was proved by Czumaj et al. [8] that many natural properties, including k-colorability, are testable using a number of queries that depends only on $1/\epsilon$ (though possibly exponentially) when the graph has certain separating (non-expansion) properties.

Testing bipartitness (i.e., k-colorability for $k = 2$) of general graphs, has previously been studied in the combined model [14]. The complexity of the algorithm presented in [14] is $\tilde{O}(\min\{\sqrt{n}, n/d\})$, and the lower bound given is $\Omega(\min\{\sqrt{n}, n/d\})$. The proof of the lower bound in [14] implies the necessity of both neighbor queries and pair queries. Specifically, if only neighbor queries are allowed then $\Omega(\sqrt{n})$ queries are necessary, and if only pair queries are allowed then $\Omega(n/d)$ queries are necessary.

Testing triangle-freeness (and more generally, subgraph-freeness) of general graphs was also studied in the combined model [3]. The main result in [3] is a lower bound of $\Omega(n^{1/3})$ on the necessary number of queries that holds for every $d < n^{1-\nu(n)}$, where $\nu(n) = o(1)$.

1.3 Our Results. In this work we are interested in studying the power of the different types of queries when testing k-colorability of general graphs for a fixed $k \geq 3$. One motivation for this investigation is that while the distance measure between graphs can be defined independently of the graph representation, the queries that the algorithm can perform do depend on the representation. Namely, allowing both pair queries and neighbor queries assumes that we have access to both types of graph representations, which is not necessarily the case. Our second motivation is purely complexity theoretic: understanding the strength of each query type separately.

In what follows we consider testing models for general graphs that are defined by the type of queries they allow, where the distance measure in all models is as defined in the combined query model. We also allow to use degree queries in all the models. Our results are stated in terms of the dependence on n and d. In all our upper bounds the dependence on both k and $1/\epsilon$ is polynomial. The bounds are for the query complexity in the different models. Since deciding k-colorability ($k \geq 3$) is hard, the running time of our algorithms is exponential in the query complexity. The results are summarized in Table 1.

THEOREM 1.1. *The following holds for testing k-colorability in the pair query model:*

1. *There exists a one-sided error tester that uses $\tilde{O}((\frac{n}{d})^2)$ queries.*
2. *Every one-sided error tester requires $\Omega((\frac{n}{d})^2)$ queries.*

THEOREM 1.2. *The following holds for testing k-colorability in the group query model:*

[1] No assumption is made about the order of the neighbors of a vertex.

[2] It is possible to emulate a degree query by $\log n$ neighbor queries, but degree queries are allowed for the sake of simplicity.

	Pair Queries	Neighbor Queries	Pair&Neighbor Queries	Group Queries
Upper Bound	$\tilde{O}((\frac{n}{d})^2)$	$O(n)$	$\min\{\tilde{O}((\frac{n}{d})^2), O(n)\}$	$\tilde{O}(\frac{n}{d})$
Lower Bound	$\Omega((\frac{n}{d})^2)$	$\Omega(n^{1-\frac{1}{\lceil (k+1)/2 \rceil}})$ $\Omega(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}})$ if $k \geq 6$	$\Omega(\frac{n}{d})$ also for 2-sided error	$\Omega(\frac{n}{d})$ also for 2-sided error

Table 1: Results for one-sided error testing of k-colorability

1. *There exists a one-sided error tester that uses $\tilde{O}(\frac{n}{d})$ queries.*
2. *Every tester requires $\Omega(\frac{n}{d})$ queries.*

Theorem 1.3. *The following holds for testing k-colorability in the neighbor query model:*

1. *There exists a one-sided error tester that uses $O(n)$ queries.*
2. *Every tester requires $\Omega\left(\max\{\frac{n}{d}, \sqrt{n}\}\right)$ queries.*
3. *Every one-sided error tester requires $\Omega(n^{1-\frac{1}{\lceil (k+1)/2 \rceil}})$ queries.*
4. *Every one-sided error tester for $k \geq 6$ requires $\Omega(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}})$ queries.*

By combining Theorems 1.1, 1.2, 1.3 and the fact that neighbor queries can be emulated using a logarithmic number of group queries (Claim A.1), we get the next corollary.

Corollary 1.1. *The following holds for testing k-colorability in the combined query model:*

1. *There exists a one-sided error tester that uses $\min((\tilde{O}(\frac{n}{d})^2), O(n))$ queries.*
2. *Every tester requires $\Omega(\frac{n}{d})$ queries.*

Discussion of our results. While the query complexity in the pair query model and the group query model is a monotone decreasing function of d, the query complexity in the neighbor query model remains roughly the same for every value of d and large values of k. We note that a similar phenomenon is observed in the case of testing bipartiteness. It follows that the pair query model is weaker than the neighbor query model for small values of d, and becomes stronger as d becomes larger. The extreme case is $d = \Theta(n)$, where in the pair query model many natural properties are testable using a constant number of queries (see e.g. [12, 1, 2]).

Pair and neighbor queries can be emulated by group queries (a single query for a pair query and a logarithmic number of queries for a neighbor query, see Claim A.1), and hence the group query model is the strongest model we consider. In fact, the lower bound proof for this model holds for a wide range of query types, and the upper bound is tight. Using this query type for testing bipartiteness, it is possible to obtain a lower query complexity than in the combined query model. Since the bounds for testing bipartiteness in the combined model are tight, even for two-sided error testers, it follows that the group query model is strictly stronger than the combined model when testing bipartiteness. We also show that for the case of testing k-colorability and a certain distribution over graphs, the combined model is strictly stronger than the optimum of the pair query and the neighbor query models.

In our proofs we extend and unify methods from previous works, and present several new methods. It turns out that procedures for sampling edges are useful in problems of property testing in general graphs, and here we give more efficient procedures. In the lower bounds section we use a new representation of graphs. The combination of this representation and balls and bins techniques enables to prove lower bounds in this model, and in particular in the problem of testing k-colorability.

Due to space limitations, some proofs do not appear in this extended abstract and can be found in the full version of this work [6].

2 Upper Bounds

In this section we establish the upper bounds in Theorems 1.1, 1.2 and 1.3. We shall need the following definition: We say that a graph G over n vertices is *almost regular* if the ratio between the maximum degree $d_{\max}$ in the graph and the average degree d is constant. This section is organized as follows. In the first subsection we establish that for every almost regular graph that is far from being k-colorable, a random induced subgraph of size $O(\frac{n}{d})$ is not k-colorable with high probability. In the second subsection we give efficient procedures for sampling edges almost uniformly in general graphs, and in the third subsection we give a framework for converting the bounds from general graphs to almost regular graphs. In the last subsection we show how to implement the reduction framework using various query types.

2.1 Almost Regular Graphs. For a graph $G = (V, E)$ and a subset of vertices $S \subseteq V$, we let $G[S]$ denote the subgraph of G that is induced by S. The upper bounds for group queries and for pair queries are based on the following theorem.

THEOREM 2.1. *Let G be an almost regular graph on n vertices with average degree d. If G is ϵ-far from being k-colorable for a constant ϵ then a random induced subgraph of size $\Theta(\frac{n}{d})$ is not k-colorable with high probability.*

The proof of Theorem 2.1 extends arguments presented in [4].

2.2 Sampling edges revisited. The problem of sampling edges plays a significant role in the context of property testing in general graphs, and in particular in this work. It is very easy to sample an edge using $O(\frac{n}{d})$ pair queries, by choosing randomly pairs of vertices until we find an edge. However, we need a more efficent way, and thus we relax our demands. Given a arbitrary value of $\delta > 0$, we would like to get every edge, apart than a set of edges of size at most δdn, with probability $\Theta(\frac{1}{dn})$. Such a procedure samples edges *almost uniformly*. In [14] it was shown how to sample an edge almost uniformly using $O(\sqrt{n})$ neighbor and degree queries. Here we present two algorithms that use neighbor and degree queries and sample edges almost uniformly. The first one uses $O(\sqrt{\frac{n}{d}})$ queries while the second one samples t edges using $\tilde{O}(t + \frac{n}{d})$ queries. In what follows we give the proof of the second lemma.

LEMMA 2.1. *There exists a procedure that samples an edge almost uniformly using $O(\sqrt{\frac{n}{d}})$ neighbor and degree queries.*

LEMMA 2.2. *There exists a procedure that samples t edges almost uniformly using $\tilde{O}(t + \frac{n}{d})$ neighbor and degree queries.*

Proof: Our procedure samples vertices with probability proportional to their degrees and proceeds as follows. In the first step, we build a random sample of the vertices by taking every vertex to a set S with probability $p = O(\frac{\log n}{d})$, and we query the degree of every vertex in S. Denote $d(S) = \sum_{u \in S} d(u)$. In the second step, we choose randomly and independently t vertices from S (with replacement), where every time a vertex $u \in S$ is chosen with probability $\frac{d(u)}{d(S)}$, and the result is a random neighbor of that vertex. First note that the number of edges with at least one endpoint of degree less than *deltad* is at most δdn. For every vertex v in the graph satisfying $d(v) > \delta d$, let L_v be the set of neighbors of v that were selected to S. Then $|L_v|$ is binomially distributed with parameters $d(v)$ and p. Therefore, with probability $O(\frac{1}{n^2})$, by Chernoff type bounds, $\frac{dp}{2} \leq |L_v| \leq 2dp$. We obtain that for every vertex w with degree d', the probability that we get w in a certain step is bounded between $\frac{d'}{2nd}$ and $\frac{2d'}{nd}$, as desired. ■

2.3 A reduction from the general case. We now describe a reduction from general graphs to almost regular graphs. It was proved in [14] that for every graph G with average degree d it is possible to construct (using a probabilistic procedure) a graph G' that is almost d-regular and has several additional properties that are useful in the context of testing bipartiteness. Here we extend the result to the problem of testing k-colorability.

LEMMA 2.3. *For every graph G over n vertices and average degree $d = \Omega(\log n)$, we can construct randomly a graph G' that has the following properties w.h.p:*

1. *G' has at most $2n$ vertices, the same number of edges, and maximal degree $d' < 2d$.*
2. *If G is k-colorable then G' is also k-colorable.*
3. *if G is ϵ-far from being k-colorable then G' is ϵ'-far being from being k-colorable for $\epsilon' = \Theta(\epsilon)$.*
4. *It is possible to sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G' using either $\tilde{O}(\frac{n}{d})$ group queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries on G.*

2.4 Establishing the upper bounds. In this subsection we show how to use the reduction (Lemma 2.3) to establish bounds for the pair query model and the group query model. In the neighbor query model we give an upper bound using a different (and simpler) approach.

Proof of Item 1 in Theorem 1.3: Our algorithm has two steps : In the first one, we perform a degree query on every vertex of the graph. Using this information, we can now choose a random edge with a single query (just choose a vertex with probability proportional to its degree, and choose a random edge incident to this vertex). In the second step sample $\frac{n \ln(k)}{2\epsilon} + \frac{1}{\epsilon}$ edges uniformly and independently. If the graph is k-colorable then trivially every subgraph will be k-colorable. On the other hand, if the graph is ϵ-far from being k-colorable, then every fixed k-partition has at least ϵdn violating edges. As the total number of edges is $\frac{nd}{2}$, the probability that none of the violating edges was selected is at most

$$(1-2\epsilon)^{(n\ln(k))/(2\epsilon)+1/\epsilon} < e^{-2} \cdot e^{-n\ln(k)} < \frac{1}{3}k^{-n} .$$

Using the union bound, with probability at least 2/3 every partition will have a violating edge in the sample, and thus the sample will not be k-colorable. ■

Proof of Item 1 in Theorem 1.1 and Item 1 in Theorem 1.2: Consider first the case that $d = o(\log n)$. For this case, in the pair query model just query all the pairs of vertices in the graph. In the group query model simply use the tester of the neighbor query model, emulating neighbor queries using group queries (Claim A.1). Otherwise ($d = \Omega(\log n)$), given a graph G, let G' be the graph obtained by Lemma 2.3. By the properties of Lemma 2.3 and by Theorem 2.1, in order to test k-colorability we need to sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G'. By the last item of the lemma this can be done using either $\tilde{O}(\frac{n}{d})$ group queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries, and the statement follows. ■

3 Lower Bounds

In this section we prove the lower bounds in Theorems 1.1, 1.2 and 1.3. In the first subsection, we describe two constructions, including a probabilistic construction of a graph that is far from being colorable yet every induced subgraph of linear size is 3-colorable. Then we use those constructions to prove bounds for various models.

3.1 The Constructions. We want to generate a sparse graph that is far from being k-colorable, yet every induced subgraph of linear size is 3-colorable. We show that a graph from the distribution $G(n,p)$ satisfies these conditions for appropriate p with high probability.

LEMMA 3.1. *For every constant value of k there exist constants $\alpha, c(k) > 0$ such that the following properties hold for a random graph from the distribution $G(n, p = \frac{c(k)}{n})$ with high probability.*

- *It has average degree bounded by a constant.*
- *It is $\Theta(1)$-far from being k-colorable.*
- *Every induced subgraph of size αn is 3-colorable.*

Given a graph G' over n' vertices and m' edges, a *d-blowup* of G' is a graph G obtained in the following manner. Every vertex v' of G' is transformed into a cluster of d vertices in G. Thus the number of vertices in G is $n = n'd$. Every edge of G' is transformed into d^2 edges in G that form a complete bipartite graph between the corresponding clusters, and hence the number of edges in G is $m = m'd^2$. In this work we use a blowup to build graphs on n vertices and average degree d that have similar properties to sparse graphs of order $\Theta(\frac{n}{d})$. Note that G contains d^2 edge-disjoint copies of G', and hence if G' is ϵ-far from being k-colorable then G is also ϵ-far from being k-colorable.

In what follows we describe a random process that creates a random bipartite multigraph. We then use this process to create a graph that is similar to a blowup of a graph, but instead of a full bipartite graph between every two connected clusters we have a random bipartite multigraph.

The Process $P_{s,d}$. The process creates a random bipartite graph over two disjoint sets A and B where $|A| = |B| = s$ and maximal degree d (we assume that $d < \lambda n$ for some constant λ) in the following manner. We maintain two matching tables, T_A and T_B, each of size $s \times d$. For every $1 \le i \le s, 1 \le j \le d$, the cell $T_{A_{i,j}}$ (respectively, the cell $T_{B_{i,j}}$) represents the j'th neighbor of the i'th vertex in A (respectively, in B). In the process we choose at every step a free cell in T_A or T_B and choose randomly free cell from the other table, and match the two cells. It is quite easy to see that the order in which we choose the cells doesn't change the distribution (see also [14]). In the end of the process we get a random bipartite multigraph. We have the following:

CLAIM 3.1. *Let G be a bipartite graph obtained by applying the process $P_{s,d}$ on the sets A and B. There exist constants $\beta, \gamma > 0$ such that for every fixed sets $A' \subseteq A, B' \subseteq B, |A'|, |B'| > \beta s$ with high probability $|e(A', B')| > \gamma sd$. Moreover, every vertex has $\Omega(d)$ different neighbors with high probability.*

CLAIM 3.2. *Let H be a graph of constant size that is not k-colorable, and let G be a graph obtained by transforming every vertex of H into $\Theta(n)$ vertices, and every two connected clusters are transformed to a random bipartite graph obtained by the process $P_{s,d}$. With high probability, the obtained graph is $\Theta(1)$-far from being k-colorable.*

3.2 Lower Bounds for Group Queries. In this section we prove lower bounds for the group query model, and since we have proved that the group query model is essentially stronger than the combined model, our results apply also to the latter one. In fact, the results here are quite general and can be applied to a wide range of query models. We start with a simple consequence of the construction of Lemma 3.1 to get a one-sided error lower bound, and then use a more complicated argument to get a two-sided error bound. Recall that a k-critical graph is a graph that is not $(k-1)$-colorable, but removal of any single edge or vertex will give a $(k-1)$-colorable graph. Every k-critical graph has minimal degree $(k-1)$, and every non-$(k-1)$-colorable graph contains a k-critical graph.

COROLLARY 3.1. *Every one-sided error algorithm for testing k-colorability must perform $\Omega(\frac{n}{d})$ group queries.*

Proof: Every one-sided error algorithm must accept whenever the subgraph it has queried is k-colorable. Take a graph of order $\Theta(\frac{n}{d})$ obtained as described in Lemma 3.1, and blow it up by factor d. Every induced subgraph of size $O(\frac{n}{d})$ is k-colorable, and thus in order to find a $(k+1)$-critical subgraph we need to get at least $\Omega(\frac{n}{d})$ positive answers. Every query returns at most one positive answer, and therefore we need to perform that number of queries in order to find an evidence that the graph is not k-colorable. ■

Next we build on a reduction from [7] to get a lower bound of $\Omega(\frac{n}{d})$ for two-sided algorithms in the group query model. The lower bound in [7] is for the case $k = 3$, and therefore we start with a reduction for general k.

LEMMA 3.2. *For any n and constant $k \geq 4$ there exists a bounded degree graph G' over $O(n)$ vertices with a set $S \subseteq V(G')$ of size n such that the following holds. For every bounded degree graph G on n vertices, if we replace $G'[S]$ by an induced copy of G then:*

- *If G is 3-colorable then G' is k-colorable.*
- *If G is $\Theta(1)$-far from being 3-colorable then G' is $\Theta(1)$-far from being k-colorable*

Proof: We show that a graph G' chosen from a certain distribution satisfies the properties with high probability, and it will follow that such a graph exists. Consider first the following distribution of graphs on $(k-3)n$ vertices. The vertices are partitioned into $k-3$ sets $U_1, U_2, \ldots U_{k-3}$ of equal size. Every two vertices from different clusters are connected by an edge with probability $\frac{1}{n}$, and every cluster forms an independent set of size n. Clearly, every graph from such a distribution is $(k-3)$-colorable. We will show that with high probability it is $\Theta(1)$-far from being $(k-4)$-colorable. First, with high probability such a graph has at most $k^2 n$ edges. Consider a coloring χ of the vertices with $k-4$ colors. Call a pair of vertices *dangerous* with respect to χ if they are from different clusters but have the same color in χ. For every color class with $n+y$ vertices, there are at least $\binom{y}{2}$ dangerous pairs. Hence, by the convexity of the function $\binom{x}{2}$, for every coloring there at least $(k-4)\binom{\frac{n}{k-4}}{2} \geq \frac{n^2}{2k}$ dangerous pairs. Therefore, by Chernoff-type bounds (see [5]), for every fixed coloring, the probability that there are less than $n/(4k)$ violating edges is much less than k^{-n} for sufficiently large n. Using the union bound, we get that almost always a graph from this distribution is $\Theta(1)$-far from being $(k-4)$-colorable.

Known results from the theory of random graphs (see [5]) imply that we may remove a small number of edges incident to high degree vertices so as to get a graph with degrees bounded by a constant. Therefore, we obtain a graph H that is (1) $(k-3)$-colorable, (2) $\Theta(1)$-far from being $(k-4)$-colorable and (3) has degrees bounded by a constant.

We now define a graph G' over $O(n)$ vertices, and a subset $S \subset V(G')$ of size n as required in the lemma (recall that the claim in the lemma is about replacing $G'[S]$ by an induced copy of a graph G). G' consists of a copy of H, and the set S, which is an independent set, where the edges between $V(H)$ and S are defined as follows. Every pair of vertices $v_1 \in V(H)$ and $v_2 \in S$ are connected with probability $\Theta(\frac{1}{n})$. This ensures that with high probability, for every choice of subsets $X \subset V(H)$ and $Y \subset S$ such that both have size n/c for some sufficiently large constant c, there are $\Omega(n)$ edges between X and Y. Assume from this point on that this is in fact true.

For any fixed choice of a graph G over n vertices, let G'' be the graph that results from replacing $G[S]$ with G. Since H is $(k-3)$-colorable, if G is 3-colorable then G'' is k-colorable, as required. We thus turn to the case that G is ϵ-far from being 3-colorable for $\epsilon = \Theta(1)$.

Consider any coloring χ of $V(G'')$ with k colors, and let $C_1, \ldots, C_k$ be the corresponding color classes. Recall that H is ϵ'-far from being $(k-4)$-colorable, for $\epsilon' = \Theta(1)$. Denote $C_i \cap V(H)$ by C_i^H and $C_i \cap V(G)$ by C_i^G. Let $\epsilon'' = \frac{1}{2k} \min\{\epsilon, \epsilon'\}$, and suppose there exists a color class C_i such that $|C_i^H| \geq \epsilon'' n$ and $|C_i^G| \geq \epsilon'' n$. In such a case, by the foregoing discussion, there are $\Omega(n)$ edges between vertices in $V(H)$ and vertices in $V(G)$, and so there is a constant fraction of violating edges with respect to the coloring χ.

Otherwise, for each C_i, let C_i' be the larger between C_i^H and C_i^G, and observe that $|\bigcup_i (C_i \setminus C_i')| \leq \frac{1}{2} \min\{\epsilon, \epsilon'\} n$. If $V(G)$ contains at most 3 subsets $C_{i_1}', C_{i_2}', C_{i_3}'$, then, since G is ϵ-far from 3-colorable, there must be a constant fraction of violating edges with respect to χ. But if $V(G)$ contains more than 3 such subsets, then $V(H)$ must contain at most $k-4$ such subsets. Since H is ϵ'-far from being $(k-4)$-colorable, in this case too there must be a constant fraction of violating edges with respect to χ.

Finally, we note again that we may remove a small number of edges that are incident to large-degree vertices so as to obtain a bounded degree-graph with the desired properties, and the proof is completed. ■

Using the lemma, we can now prove the lower bound:

Proof of Item 2 in Theorem 1.2: We first prove the bound for $k = 3$, and in the end of the proof we explain how to adjust it to other values of k. The authors of [7] show that there exist two distributions of b-regular graphs (for a constant b) over n' vertices with the following properties. The first distribution, denoted as D'_{3col} contains graphs that are all 3-colorable. The second distribution, denoted as $D'_{\text{3col-far}}$ contains graphs that are all $\Theta(1)$-far from being 3-colorable with high probability. Moreover, $|D'_{\text{3col-far}}| = |D'_{\text{3col}}|$ and the distributions are statistically indistinguishable by a tester that has observed $o(n')$ vertices of the input graph.

We create two distributions, $D_{\text{3col}}, D_{\text{3col-far}}$ that are created by a d-blowup of every graph in the original distributions $D'_{\text{3col}}, D'_{\text{3col-far}}$. Suppose that such a tester T' exists, we create the following tester T that distinguishes between graphs from D'_{3col} and $D'_{\text{3col-far}}$ using vertices from $o(n')$ different clusters. Given a graph G' on n' vertices, denote a d-blowup of G' by G. Clearly, every single query to G can be emulated using a single query to G'. Use now the tester T' to distinguish between the distributions D_{3col} and $D_{\text{3col-far}}$, using vertices from $o(n') = o(\frac{n}{d})$ different clusters. If T' answers that the graph G was created by the distribution D_{3col}, T answers that G' was created by the distribution D'_{3col}, and otherwise it answers that the graph created by the distribution $D'_{\text{3col-far}}$.

We note that since each vertex in each graph created by the distributions $D_{\text{3col}}, D_{\text{3col-far}}$ has neighbors from exactly b different clusters, our knowledge graph for a tester that performs $o(n')$ group queries is contained in a subgraph touching only $o(n'b) = o(n')$ different clusters, and thus indeed our tester can't distinguish between the two distributions using $o(n')$ queries.

Finally, for every $k > 3$, we can adjust the proof as follows. We replace the distributions D'_{3col} and $D'_{\text{3col-far}}$ by two new distributions $D_{\text{k-col}}$ and $D_{\text{k-col-far}}$. Every graph G from the distributions is replaced by a graph G' given by Lemma 3.2. The new graph has $O(n)$ vertices, degrees bounded by a constant. If the graph is generated by $D_{\text{k-col}}$ it is k-colorable and otherwise it is $\Theta(1)$-far from being k-colorable. We may also assume that the tester knows in advance all the edges in G' except the edges in the induced copy of G. Since the graph G' is fixed (except the induced copy of G), the only way to separate between the two distributions is by the induced copy of G, and thus we may apply the above argument and the theorem follows. ■

3.3 A Lower Bound for Pair Queries. Next we prove a one-sided error lower bound for the pair query model. Roughly speaking, using the blowup technique, it is enough to prove a lower bound for sparse graphs of order $\frac{n}{d}$, since every such graph can be transformed to an almost regular graph of average density $\Theta(d)$.

Proof of Item 2 in Theorem 1.1: We define a random process P that builds a graph G while interacting with the algorithm. In the beginning of the algorithm, the process divides the vertices into equal clusters of size d. We may also assume that the tester knows that the input is a d-blowup of another graph, and it also knows a single vertex from every cluster. Clearly, any optimal tester will perform pair queries only between those vertices, and will not require more information. Hence, we may assume that the tester works on a sparse graph of size $\frac{n}{d}$ instead of a graph of size n. While the tester performs a pair query between two vertices, the process returns a positive answer with probability $p = 8dk^3/n$ independently of the other queries as in Lemma 3.1. At the end of the algorithm's execution, the process completes all other edges in G using the same distribution, thus creating a graph from the distribution $G(n/d, p)$. By Lemma 3.1, with high probability every minimal evidence that G is not k-colorable has at least cn/d vertices for a constant c. Such an evidence is a $(k+1)$-critical graph and therefore every vertex must have minimal degree k and the total number of edges is at least $\frac{kcn}{2d}$, and this is also a lower bound for the required number of positive answers. The probability that a single query returns a positive answer is p independently of other answers. Therefore, the probability that in $\frac{kcn}{6dp}$ queries the tester will get at least $\frac{kcn}{2d}$ positive answers is at most $1/3$ by Markov's inequality. Hence every tester has to perform at least $\frac{cn^2}{48k^2d^2}$ pair queries and the theorem follows. ■

3.4 Lower Bounds for Neighbor Queries. We use a representation of the knowledge graph as a di-graph, where if we query a random neighbor of a vertex v and find a vertex u then we direct an edge from v to u. For a fixed orientation O, the indegree of a vertex v is the number of edges that are directed into v. We start with two easy claims.

CLAIM 3.3. *Let T be a graph of order n with average degree d, and consider an orientation of H. There exists a vertex with indegree at least $\lceil d/2 \rceil$.*

CLAIM 3.4. *Let H be a K_4-free graph that is not k-colorable ($k \geq 6$), and let O be an orientation of H.*

The set of vertices with indegree at least $\lceil \frac{k}{2} \rceil - 1$ with respect to O induces a non-3-colorable graph.

Proof: Denote the set of vertices with indegree at least $\lceil k/2 \rceil - 1$ by C and the complement of C by U. In U every vertex has indegree at most $\lceil k/2 \rceil - 2$. Suppose that $H[U]$ is not $(k-3)$-colorable, then it must contain a $(k-2)$-critical subgraph. Since that subgraph doesn't contain a copy of K_4, by Brooks' Theorem (see e.g., [18]), it has average degree more than $k-3$. By the previous claim, such a subgraph must contain a vertex with indegree more than $\lceil (k-3)/2 \rceil$; the latter quantity is at least $\lceil k/2 \rceil - 1$, a contradiction. If $H[C]$ is 3-colorable, we get that the whole graph can be colored with k colors, which contradicts our assumption and thus completes the proof. ∎

We want to show that in order to find a vertex with large indegree in our knowledge graph, we need to perform many neighbor queries. To this end, consider the following simple game $R(n, p, \ell, c)$: We have p players, every one has a bag with n balls, labeled by $1, 2, \ldots, n$. In every step, we may order a player to take a ball from his bag, and he gets a random ball, without replacement. In every round, the probability that we get a ball with a certain label is at most $\frac{c}{n'}$, where n' is the number of balls left in the bag (that is, the sampling is not necessarily uniform but it is almost uniform). Our goal is to have ℓ players that have a ball with the same label, and in the following claim we bound the number of rounds that such a game will take.

Claim 3.5. *With high probability, the game $R(n, p, \ell, c)$ takes $\Omega(n^{1-1/\ell})$ rounds for a constant c.*

We first prove a two-sided error bound that follows immediately from previous results, and then use the game terminology to prove a better one-sided error bound for the case $k \geq 6$.

Proof of Item 2 in Theorem 1.3: As the group query model is stronger than the neighbor query model, by Theorem 1.2 we get a lower bound of $\Omega(\frac{n}{d})$ queries. In [14] it was proved that every algorithm that performs only neighbor queries requires $\Omega(\sqrt{n})$ queries. The claim follows. ∎

Proof of Item 3 in Theorem 1.3: Let H_k be a fixed $(k+1)$-critical graph which is not K_{k+1}. We define a process P that creates a graph G while interacting with the tester. Initially, the graph G contains a cluster of $n' = \Theta(n)$ vertices for every vertex in H. Every vertex in G will be connected to $d' = \Theta(d)$ neighbors from every incident cluster. We may also assume that the tester knows in advance the structure of H, and that the neighbors in the incidence list of every vertex are orderd by their clusters. Therefore, the tester knows in advance before performing a query from which cluster the neighbor will be. Every pair of incident clusters are connected by a random bipartite graph as defined by the process $P_{s,d}$: We maintain for each such pair of clusters two tables of size $n' \times d'$, and when the tester asks for the i'th neighbor of a vertex v we match the appropriate cell from the relevant table to a random free cell in the second table. At the end of the algorithm's execution, the process P completes all the bipartite graphs between every two connected clusters in an arbitrary way. By Claim 3.2 the obtained graph is not k-colorable with high probability.

During the execution of the tester, we represent our knowledge graph as a directed graph, where if we perform a neighbor query from a vertex v and get a vertex u, we orient the edge from v to u. As every minimal evidence is a $(k+1)$-critical graph that is not a clique, by Brooks' Theorem it has average degree more than k. By Claim 3.3 it has at least one vertex with indegree at least $\lceil (k+1)/2 \rceil$.

We now wish to prove that in order to find a single vertex in the graph with such indegree, we need to perform many queries. We represent every vertex v as a possible label $b(v)$ and as a player $p(v)$. When we perform a neighbor query from a vertex v and find a vertex u, we say that the player $p(v)$ gets a ball that is labeled $b(u)$. We note that the incidence lists of every vertex are ordered randomly, and that every edge in H_k was replaced by a random almost regular bipartite graph. If the tester performs $o(n)$ queries, then the probability of every given round to give a ball with a certain label is at most $\Theta(\frac{1}{n})$. Therefore, we have an $R(n', n', \lceil (k+1)/2 \rceil, \Theta(1))$ game, and thus by Claim 3.5, with high probability one needs to perform $\Omega(n^{1-\frac{1}{\lceil (k+1)/2 \rceil}})$ neighbor queries in order to find a neighbor with the required indegree, and the proof follows. ∎

Proof of Item 4 in Theorem 1.3: By Lemma 3.1 there exists a graph of order $\Theta(\frac{n}{d})$ that is ϵ-far from being k-colorable yet every induced subgraph of size $\Omega(\frac{n}{d})$ is 3-colorable (in particular, the graph doesn't contain K_4 as a subgraph). Let G be a d-blowup of such a graph, then G is also K_4-free and ϵ-far from being k-colorable. Let H be a directed graph that represents our knowledge graph. If H is an evidence that the graph G is not k-colorable, then H contains a $(k+1)$-critical subgraph. By Claim 3.4, the set of vertices in H with indegree at least $\lceil k/2 \rceil - 1$ is not 3-colorable, and therefore by the properties of the construction, it has vertices from $\Theta(\frac{n}{d})$ different clusters. We now use the balls and labels approach to prove that we need many queries in order to create a single vertex with high indegree.

We represent every cluster as a game with d possible labels, and again each query that gets a vertex from the cluster corresponds to a round in the game such that one of the players gets a the corresponding label. The list of bins is ordered randomly, but note that we can assume that the tester knows in advance the cluster name of every vertex in the list, and thus every query is a round in a certain game. For every such game, if it has $o(d)$ rounds, the probability of every label is at most $\Theta(\frac{1}{d})$. We have $R(\Theta(d), d, \lceil k/2 \rceil - 1, \Theta(1))$ game, and thus by Claim 3.5 in order to find a vertex with indegree at least t in our directed knowledge graph, with high probability we need to perform at least $\Omega(d^{1-\frac{1}{t}})$ neighbor queries. Combining those two facts, since we must find vertices in $\Theta(\frac{n}{d})$ clusters with indegree at least $\lceil k/2 \rceil - 1$, with high probability we need at least $\Omega(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}})$ neighbor queries, and the theorem follows. ■

3.5 On the power of the combined model. Let T be the distribution of d-blowups of graphs given by Lemma 3.1. We saw that for this distribution, there exist lower bounds for the pair query and neighbor query models of $\Omega((\frac{n}{d})^2)$ and $\Omega(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}})$ (if $k \geq 6$), respectively. Here we show a one-sided error algorithm in the combined model that uses $\tilde{O}(\frac{n}{\sqrt{d}})$ queries for this distribution. This shows that the combined model is stronger than the optimum of the pair query and neighbor query models for the problem of k-colorability.

Proposition 3.1. *There exists an algorithm such that given a graph $G \in T$ that is not k-colorable reveals a non k-colorable subgraph using $\tilde{O}(\frac{n}{\sqrt{d}})$ in the combined model with high probability.*

Here we give the main idea of the algorithm. We sample a set of vertices that contains at least one vertex from every cluster, and then create an identifier for every vertex by sampling $\tilde{O}(\sqrt{d})$ random neighbors. With high probablity, every pair of vertices that belong to the same cluster will have a common neighbor in their identifiers. Pairs of vertices from different clusters may also have a common neighbor, but there won't be too many such pairs. Using pair queries we can distinguish between the two cases. By this observation, we can recover the clusters graph as follows: First, we take a sample of size $\tilde{O}(\frac{n}{d})$ vertices, and so with high probability we have at least one vertex from every cluster. By creating an identifier for every vertex, we can choose from that sample a single vertex from every cluster. Denote this set by S. In the second step, we take $O(\log(n))$ random neighbors for every vertex in S, and then create an identifier for every such neighbor. Using these identifiers, we know with high probability which clusters are connected, and thus we can recover the clusters graph with high probability. Finally, using pair queries we can check that the clusters graph is indeed a subgraph, and thus we have a one-sided error algorithm for this distribution.

4 Concluding Remarks and Open Questions

Our upper and lower bounds are nearly tight for the problem of one-sided error testing of k-colorablity in the pair query model and in the neighbor query model. In the group query model our bounds are tight even for two-sided error testers. It seems plausible to assume that the lower bounds for the two-sided error case in the pair query model and in the neighbor query model are $\Omega((\frac{n}{d})^2)$ and $\Omega(n)$, respectively.

The exact power of the combined model of testing k-colorablity is still not known. We gave evidence that it may be weaker than the group query model, as it is for the problem of testing bipartiteness, and may be stronger than the optimum of the pair and neighbor models (as it is for the distribution used in Section 3).

Finally, the exact query complexity of various models is still not determined for many problems, including problems that were already studied in the combined model. For example, it may be interesting to determine the query complexity of testing bipartiteness in the pair query model.

Acknowledgment. We would like to thank Elad Verbin for helpful discussions that led to better results on sampling edges.

References

[1] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.

[2] N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. In *Proc. of the 38 ACM STOC*, pages 251–260, 2007.

[3] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle freeness in general graphs. In *Proceedings of the 17th Symposium on Discrete Algorithms (SODA'06)*, pages 279–288, 2006.

[4] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Math*, 15(2):211–227, 2002.

[5] N. Alon and J. Spencer. *The probablistic method.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[6] I. Ben-Eliezer, T. Kaufman, M. Krivelevich, and D. Ron. Comparing the strength of query

types in property testing: The case of testing k-colorability. Manuscript, available from `http://www.eng.tau.ac.il/~danar/papers.html`, 2007.

[7] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded degree graphs. In *Proceedings of the Forty-Third Annual Symposium on Foundations of Computer Science*, pages 93–102, 2002.

[8] A. Czumaj, A. Shapira, and C. Sohler. Testing hereditary properties of non-expanding bounded-degree graphs. Manuscript, 2007.

[9] D. Du and F. Hwang. *Combinatorial group testing and its applications*. World Scientific, 1993.

[10] R. Duke and V. Rödl. On graphs with small subgraphs of large chromatic number. *Graphs Combin*, 1:91–96, 1985.

[11] E. Fischer. The art of uninformed decisions: A primer to property testing. *The Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[12] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *JACM*, 45(4):653–750, 1998.

[13] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

[14] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, 2004.

[15] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2):165–183, 2002.

[16] D. Ron. Property testing. In *Handbook of Randomized Computing*, Volume II, Chapter 15, pages 597–649. Edited by S. Rajasekaran, P. M. Pardalos, J.H. Reif and J. Rolim, Kluwer Academic Publishers.

[17] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[18] D. West. *Introduction to Graph Theory.* Prentice Hall, 2000.

A Group Queries

In this section we prove two simple claims that demonstrate the power of group queries. We start by emulating the standard queries by group queries:

Claim A.1. *A pair query can be emulated using a single group query, while a neighbor query can be emulated using a logarithmic number of group queries.*

Proof: Clearly, a pair query is a special case of group queries, where the set we query for has a single element. Next we show how to emulate a neighbor query. For a given vertex v in order to emulate a neighbor query for v we need to show how to find a random neighbor of v. This can be done in the following manner : Order the vertices of G as the leaves of a full binary tree randomly. For every inner node t denote by $S(t)$ the set of vertices which t is an ancestor of their corresponding leaves, and denote by $C(t)$ the two direct children of t. Now, apply the following procedure, starting from the root: Given a node t, use group query on $S(t')$ for every $t' \in C(t)$. If the two queries return negative answer then v does not have neighbors at all. If one of the queries returns a positive answer, apply the procedure recursively in the node that returns positive answer. Otherwise, choose one of the children randomly and apply the procedure recursively on it. The procedure ends when it finds a single leaf, and since the vertices were orderd randomly in the leaves, this procedure gives a random neighbor in a logarithmic number of queries. ■

Next we use a similar idea to show how to find efficiently all the edges of an induced subgraph using group queries.

Claim A.2. *It is possible to find all the edges of a random induced subgraph using $\tilde{O}(n'+m')$ group queries where n' is the number of vertices and m' is the number of edges in the induced subgraph.*

Proof: Let U be a set of vertices of size n'. For every vertex $v \in U$, we find all its neighbors in U by the following procedure: Order all the vertices of U as the leaves of a full binary tree. For every inner node t denote by $A(t)$ the set of leaves in the subtree rooted at t. Our aim is to find all the leaves which represent neighbors of v. Starting from the root, if the node is a leaf we perform a group query (that is actually a pair query) to check if the corresponding vertex is a neighbor of v. Otherwise, perform a group query between v and $A(t)$. If the answer is 'false' return nothing, and otherwise apply the same procedure recusively on the two direct children of t, and return the union of their results. It is quite easy to see that this procedure returns all the neighbors of v, and for every neighbor we perform a logarithmic number group queries to find it. Thus, the total number of queries is $\tilde{O}(n' + m')$, as desired. ■

We note that it is quite easy to see that by using this procedure we obtain more efficient testers for the problems of testing bipartiteness (see [14]) and testing triangle freeness (see [3]). Since the upper and lower bounds in the case of testing bipartiteness are tight, this shows that the group query model is strictly stronger than the combined model.