Testing Triangle-Freeness in General Graphs

Noga Alon [*] Tali Kaufman [†] Michael Krivelevich [‡] Dana Ron [§]

Abstract

In this paper we consider the problem of testing whether a graph is triangle-free, and more generally, whether it is H-free, for a fixed subgraph H. The algorithm should accept graphs that are triangle-free and reject graphs that are far from being triangle-free in the sense that a constant fraction of the edges should be removed in order to obtain a triangle-free graph. The algorithm is allowed a small probability of error.

This problem has been studied quite extensively in the past, but the focus was on dense graphs, that is, when $d = \Theta(n)$, where d is the average degree in the graph and n is the number of vertices. Here we study the complexity of the problem in general graphs, that is, for varying d.

Our main finding 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 $\nu(n) = o(1)$. Since when $d = \Theta(n)$ the number of queries sufficient for testing has been known to be independent of n, we observe an abrupt, *threshold-like* behavior of the complexity of testing around n. This lower bound holds for testing H-freeness of every non-bipartite subgraph H.

Additionally we provide sub-linear upper bounds for testing triangle-freeness that are at most quadratic in the stated lower bounds, and we describe a transformation from certain one-sided error lower bounds for testing subgraph-freeness to two-sided error lower bounds.

Finally, in the course of our analysis we show that dense random Cayley graphs behave like quasi-random graphs in the sense that relatively large subsets of vertices have the "correct" edge density. The result for subsets of this size cannot be obtained from the known spectral techniques that only supply such estimates for much larger subsets.

1 Introduction

In this work we consider the problem of testing subgraph-freeness, and in particular triangle-freeness, in general graphs. Let n denote the number of vertices in the graph, let d denote the average degree, and let $d_{\max}$ denote the maximum degree. Given a distance parameter $\epsilon > 0$, we would like to design an algorithm that distinguishes with high probability between the case that the graph contains no triangles and the case in which more than $\epsilon \cdot nd$ edges should be removed so that no triangles remain. To this end we allow the algorithm query access to the graph. In particular, for any vertex of its choice, the algorithm may ask for the degree of the vertex, it may ask to get the i-th neighbor of the vertex for every $i \leq n$ (if the vertex has less than i neighbors then a null answer is returned), and it may ask whether there is an edge between any two vertices.

Subgraph-freeness, and more specifically, triangle-freeness, is one of the most basic problems studied in property testing. The interest in this problem is both due to the fact that triangle-freeness is a fundamental and simple graph property, and it is due to the relation between triangle-freeness and the study of dense sets of integers with no three-term arithmetic progression.

Dense graphs. Most of the focus in previous works was on testing triangle-freeness in dense graphs, that is, when $d = \Theta(n)$. The authors of [2] showed that it is possible to test triangle-freeness in dense graphs using a number of queries that is *independent* of n, and is of tower-type behavior in $1/\epsilon$. Alon [1] proved that a super-polynomial dependence on $1/\epsilon$ is necessary for testing subgraph-freeness of all non-bipartite subgraphs. When the fixed subgraph is bipartite then $O(1/\epsilon)$ queries suffice [1]. It is also observed in [1] (and much earlier, though implicitly, in [14]) that the problem of testing triangle-freeness is intimately related to the

[*]Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel, and Institute for Advanced Study, Princeton, NJ 08540, USA. E-mail: nogaa@post.tau.ac.il. Research supported in part by a USA Israel BSF grant, by a grant from the Israel Science Foundation and by the Von Neumann Fund.

[†]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: kaufmant@post.tau.ac.il, This work is part of the author's Ph.D. thesis prepared at Tel Aviv University under the supervision of Prof. Noga Alon, and Prof. Michael Krivelevich. Research done in part while the author visited the Radcliffe Institute for Advanced Study, Harvard University.

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

[§]Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: danar@eng.tau.ac.il. Research done in part while the author was a fellow at the Radcliffe Institute for Advanced Study, Harvard University.

famous (and very hard) problem of the existence of dense sets of integers without a three-term arithmetic progression. Alon's lower bound, which was proved for one-sided error algorithms, was extended in [7] to two-sided error algorithms. Other related results include [6].

Bounded-Degree graphs. In the other extreme, as was observed in [9], when $d_{\max} = O(1)$ then $O(1/\epsilon)$ queries suffice for testing triangle-freeness. More generally, $O(d^\tau/\epsilon)$ queries suffice for testing H-freeness in graphs with maximum degree $O(d)$, where τ is the diameter of H.

General graphs. In this work we study the complexity of testing triangle-freeness of graphs that lie between the two extremes. Namely, we would like to understand the dependence of the query complexity on the average degree d, and we do not want to necessarily assume that $d_{\max} = O(d)$. In the latter aspect we follow the work [13] on testing the diameter of sparse, but unbounded-degree, graphs, and in both aspects we follow the work [12] on testing bipartiteness of general graphs. Note that the fact that the graph has varying degrees makes the task of testing triangle-freeness significantly harder. Consider for example sparse graphs, that is, graphs with average degree $d = O(1)$. As we mentioned before, when $d_{\max} = O(1)$, $O(1/\epsilon)$ queries suffice for testing triangle-freeness. However, our work shows that when $d_{\max} = \Theta(n)$, $\Omega(\sqrt{n})$ queries are required for testing triangle-freeness.

Our contributions. The main contributions of this paper, on a qualitative level, are as follows:

- We discover a threshold-type behavior in testing H-freeness, for every non-bipartite fixed graph H: whenever $d = O(n^{1-\nu(n)})$, where $\nu(n)$ is a function that satisfies $\nu(n) = o(1)$, the number of queries that are necessary to test H-freeness is $\Omega(n^{1/3})$, while, as discussed above, for $d = \Theta(n)$ the query complexity is a function of ϵ only. This is in sharp contrast with the results of [12], where a smooth behavior of the complexity of testing bipartiteness as a function of d was described;
- We provide a transformation from lower bounds for testing H-freeness using one-sided error algorithms to those for two-sided error algorithms; though the suggested transformation carries some technical restrictions, it is general enough to capture a variety of lower bounds of this sort;
- We give quantitative lower and upper bounds for testing triangle-freeness in general graphs;
- We show that the edge distribution in random Cayley graphs is close to that of truly random graphs of the same edge density. This is proven by direct combinatorial and probabilistic arguments, without relying on the eigenvalue machinery, which is incapable of proving such results for subsets that are too small. Although we need this result for property testing purposes, we feel it is of enough independent interest to be stated here.

1.1 A lower bound and a sharp threshold

Our main result is:

THEOREM 1. *There is a lower bound of $\Omega(n^{1/3})$ for testing triangle-freeness in general graphs. The lower bound holds for algorithms that are allowed two-sided error, and for every d that is upper bounded by $O(n^{1-\nu(n)})$ where $\nu(n) = \frac{\log\log\log n + 4}{\log\log n}$. For some values of d the lower bound reaches $\Omega(n^{1/2})$.*

Theorem 1 is actually the union of three lower bounds (whose one-sided error versions are stated in Lemmas 1, 2 and 3), which are applied to different values of d. The exact expression for the lower bound is

$$\Omega\Big(\max\Big\{\sqrt{n/d}, \min\{d, n/d\}, \min\big\{\sqrt{d}, n^{2/3}/d^{1/3}\big\}\cdot n^{-o(1)}\Big\}\Big)$$

For a schematic illustration see Figure 1.

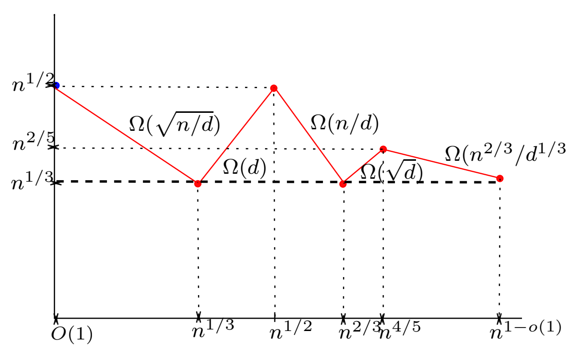

Figure 1: A schematic illustration of our lower bounds. The x-axis represents d and the y-axis represents the lower bound. Notice that the lower bound lies entirely above the horizontal line at height $n^{1/3}$.

Recall that when $d = \Theta(n)$ then testing can be performed using a number of queries that is independent of n [2]. Thus we observe a sharp transition between our lower bound of $\Omega(n^{1/3})$ that holds until $d = n^{1-\nu(n)}$ (recall that $\nu(n) = o(1)$), and the upper bound at $d = \Theta(n)$, which does not depend on n. The exact behavior of the complexity of testing triangle-freeness when $n^{1-\nu(n)} \leq d \leq n$ remains open.

Using techniques that were previously applied in [1] it is possible to extend Theorem 1 to testing subgraph-freeness of other non-bipartite subgraphs.

Theorem 2. *There is a lower bound of $\Omega(n^{1/3})$ for testing H-freeness in general graphs where H is a fixed non-bipartite graph. The lower bound holds for algorithms that are allowed two-sided error, and for every d that is upper bounded by $n^{1-\nu(n)}$ (for $\nu(n) = o(1)$ as in Theorem 1). For some values of d the lower bound reaches $\Omega(n^{1/2})$.*

We wish to note that the difference between the complexity of testing bipartite and non-bipartite graphs is caused by the difference in the behavior of their Turán numbers – they are subquadratic for the former and quadratic for the latter. A more detailed discussion can be found in [1].

1.2 Upper bounds

We show that for every graph density, there exists an algorithm for testing triangle-freeness whose query complexity is sublinear in n. Furthermore, the upper bound is always at most quadratic in the corresponding lower bound.

Theorem 3. *There is an upper bound of $O(n^{6/7})$ for testing triangle-freeness in general graphs for every value of d. The upper bound can go down to $O(n^{1/2})$ for some values of d. In all cases the upper bound is at most quadratic in the lower bound that holds for that density. If $d_{\max} = O(d)$ then the upper bound is $O(n^{4/5})$ for all values of d.*

The exact expression for our upper bound is $O\left(\min\left\{\sqrt{nd}/\epsilon^{3/2}, (n^{4/3}/d^{2/3})/\epsilon^2\right\}\right)$, where in the case that $d_{\max} = O(d)$, the first term is replaced by d/ϵ.

1.2.1 Tight results.

There are two cases in which our lower and upper bounds are tight. The first case is graphs in which $d_{\max} = O(d)$ and $d \leq \sqrt{n}$. For this case the complexity is $\Theta(d)$ (for constant ϵ). The second case is general sparse graphs, that is, graphs for which $d = \Theta(1)$. For these graphs the complexity is $\Theta(\sqrt{n})$.

1.3 Our techniques

Behrend Graphs and Cayley graphs. In the proof of our third lower bound (Lemma 3), we build on graphs that are known as Behrend graphs, which were previously used in the context of testing triangle-freeness in [1]. Here we prove that random Behrend graphs have a certain property that we can exploit in order to obtain our lower bound. Behrend graphs are variants of well studied Cayley graphs, and our proof concerning properties of random Behrend graphs extends to Cayley graphs.

Specifically, we show that for dense random Cayley graphs the edge density in relatively large induced subgraphs is close to the edge density of the whole graph. It was previously shown [5] that random Cayley graphs are expanders and hence have the property that the density of every induced subgraph on sufficiently many vertices is very close to the density of the graph. However, the known techniques for proving this property are based on estimating the second eigenvalue of the graph's adjacency matrix, and do not supply any informative bounds for sets of vertices that are much smaller than the number of vertices divided by the square root of the degree. Our results for Cayley graphs apply both for Cayley graphs over abelian and non-abelian groups, while Behrend graphs were considered only in an abelian setting. Our techniques are somewhat reminiscent of those of [4, 10].

A reduction from one-sided error lower bounds to two-sided error lower bounds. We obtain our two main lower bounds by first establishing lower bounds that hold for one-sided error algorithms. We then prove a transformation from one-sided error lower bounds to two-sided error lower bounds that hold under certain assumptions, and apply it to obtain our two-sided error lower bounds. This transformation may be of use in future lower bound proofs for subgraph freeness. We note that in [7] a transformation was given in the case of dense graphs, but it is not applicable in general.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with n vertices labeled $1, \ldots, n$, and let d denote the average degree in G, where we assume that $d = \Omega(1)$.[1] For each vertex $v \in V$ let $\deg(v)$ denote the degree of vertex v. The edges incident to v (and their end-points, the neighbors of v), are labeled from 1 to $\deg(v)$. Note that each edge has two, possibly different, labels, one with respect to each of its end-points. For a graph G and a subset of vertices $U \subseteq V$, we refer to the edges in the subgraph of G that is induced by U as the edges *spanned* by U in G.

A graph G is said to be *triangle-free* if for every three vertices u, v, w in G, at least one of the three vertex-pairs (u, v), (v, w), or (w, u) is not an edge in G. A graph G is *ϵ-far from (being) triangle-free* if it is necessary to remove more than ϵnd edges from G in

[1] Our results can be extended to the case that $d = o(1)$ (that is, very sparse graphs). However, for the sake of simplicity, and since we believe that the very sparse case is of less interest, we assume that $d = \Omega(1)$.

order to obtain a triangle-free graph.[2]

A testing algorithm for triangle-freeness is required to accept with probability at least 2/3 every graph that is triangle-free and to reject with probability at least 2/3 every graph that is ϵ-far from being triangle-free, where ϵ is a given distance parameter. If the algorithm always accepts triangle-free graphs then it has *one-sided error*, otherwise it has *two-sided error*. In order to perform this task the testing algorithm is allowed the following types of queries:

- *Degree* queries: for any vertex u of its choice, the algorithm can obtain $\deg(u)$.
- *Neighbor* queries: for any vertex u and index $1 \leq i \leq \deg(u)$, the algorithm may obtain the i-th neighbor of vertex u.
- *Vertex-pair* queries: for any pair of vertices (u, v), the algorithm can query whether there is an edge between u and v in G.

3 A lower bound of $\Omega\left(\sqrt{n/d}\right)$

In this section we establish our first, and simplest lower bound.

LEMMA 1. *Every algorithm for testing triangle-freeness must perform $\Omega(\sqrt{n/d})$ queries. This lower bound holds for two-sided error algorithms as well.*

Proof. In order to prove a two-sided error lower bound of $\Omega(q)$ queries for testing triangle-freeness, it suffices to describe two families of graphs for which the following two conditions hold. (1) The graphs in the first family are all triangle-free, while the graphs in the second family are all $\Theta(1)$-far from being triangle-free. (2) Any algorithm that distinguishes with constant probability between a graph selected uniformly in one family, and a graph selected uniformly in the second family, must perform $\Omega(q)$ queries.

In particular, consider the following two families of graphs over n vertices and with average degree d. Each family is determined by a single graph, and consists of all possible $n!$ labelings of the vertices of the graph. Hence it suffices to describe the two graphs (one per family). In one graph there is a clique of size $\sqrt{nd}$, and in the other graph there is a complete bipartite graph between two sets of vertices, each of size $\sqrt{nd}$. In addition, in both graphs the remaining set of vertices spans a d-regular triangle-free graph. The second graph is clearly triangle-free, and it is not hard to verify that the first graph is $\Theta(1)$-far from being triangle-free. However, in order to distinguish between the two graphs (or more precisely, in order to distinguish between graphs that are selected uniformly from each of the two families), the algorithm must obtain a vertex in the clique / complete bipartite subgraph. To this end the algorithm must perform $\Omega(n/\sqrt{nd}) = \Omega(\sqrt{n/d})$ queries.

4 A lower bound of $\Omega(\min\{d, n/d\})$

Our next lower bound improves on the lower bound in Section 3 when $d > n^{1/3}$.

LEMMA 2. *Every one-sided error algorithm for testing triangle-freeness must perform $\Omega(\min\{d, n/d\})$ queries. This lower bound holds even when $d_{\max} = O(d)$.*

The lower bound in Lemma 2 is extended to two-sided error algorithms in Section 6. The proof of Lemma 2 appears in the full version of this paper [3]; here we give a brief sketch.

We prove the lemma by describing a distribution on graphs such that the following holds: On one hand almost all of its support is on graphs that are far from being triangle-free. On the other hand, if we select a graph according to the distribution, then every algorithm must perform $\Omega(\min\{d, n/d\})$ queries before it reveals a triangle with probability bounded away from zero. Since we currently focus on testing algorithms that have one-sided error, this implies a lower bound on the query complexity of such algorithms. We consider the case that $d = c \cdot \sqrt{n}$ for a particular constant $c > 0$. The argument is easily extended to both smaller and larger degrees.

The lower bound distribution, denoted D_Δ, is defined as follows: A graph is generated by first selecting a random partition of the vertices into equal-size subsets of size $n' = n/3$ denoted V_1, V_2, V_3. Next, between each pair of subsets, $d' = d/2 = c/2 \cdot \sqrt{n}$ random perfect matchings are selected. In order to show that with probability $1 - o(1)$ a graph chosen uniformly according to D_Δ is $\Omega(1)$-far from being triangle-free we show that with high probability a graph generated according to D_Δ contains $\alpha \cdot nd$ edge-disjoint triangles for some constant $0 < \alpha < 1$. It is not hard to show that the expected number of triangles that do not share an edge with any other triangle is $\Omega(nd)$. By exploiting the limited dependence between the different triangles, we can prove that there are $\Omega(nd)$ edge-disjoint triangles with high probability.

In order to prove that any algorithm must perform $\Omega(\min\{d, n/d\})$ queries before it reveals (with probability bounded away from zero) a triangle in a graph generated according to D_Δ, we apply an argument similar

[2] Since the number of edges in the graph is $(nd)/2$, the standard definition of ϵ-far would be that more than $(\epsilon nd)/2$ edges should be removed so that the graph becomes triangle-free. In order to simplify the presentation we slightly modify the definition.

to ones applied in other property testing lower bounds (see, e.g, [9, 12]). Specifically, we define a process that answers the algorithm's queries while generating a graph according to D_Δ. We show that if the algorithm performs $o(n/d)$ queries then the probability that any vertex-pair query is answered positively is $o(1)$, due to the density of the graph. If we ask $o(\sqrt{n})$ neighbor queries then due to the birthday paradox the probability that any cycle (in particular, a triangle) is detected is $o(1)$ as well.

5 An improved lower bound for high degrees

In this section we establish the following lemma, which improves on our previous lower bound of $\min\{d, n/d\}$ when the degree of the graph is at least $n^{2/3+o(1)}$.

LEMMA 3. *Every one-sided error testing algorithm for triangle-freeness must perform* $\Omega\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ *queries, where* $\nu(n) = \frac{\log\log\log n + 4}{\log\log n}$. *This lower bound holds even for d-regular graphs.*

In order to prove the lemma, here too we define a distribution over graphs that are far from being triangle free. We then prove a lower bound on the number of queries that are required in order to detect a triangle with probability bounded away from zero in a graph that is generated according to the distribution. As we shall see, it will actually be convenient to consider graphs over $3n$ vertices and degree $2d$.

5.1 A variant of Behrend graphs

Our lower bound distribution builds on graphs that are variants of what are known as *Behrend Graphs* [8, 14, 15]. These graphs are defined by sets of integers that include no three-term arithmetic progression (abbreviated as 3AP). Namely, these are sets $X \subset \{1, \ldots, n\}$ such that for every three elements $x_1, x_2, x_3 \in X$, if $x_2 - x_1 = x_3 - x_2$ (i.e., $x_1 + x_3 = 2x_2$), then necessarily $x_1 = x_2 = x_3$. Below we describe a construction of such sets that are large (relative to n), and later explain how such sets determine Behrend graphs. Our construction of X uses similar ideas to those used in known constructions [8, 15] and gives a slightly weaker result. However, our alternative construction is somewhat simpler, and the size of the resulting set suffices for our purposes.

LEMMA 4. *For every sufficiently large n there exists a set* $X \subset \{1, \ldots, n\}$, $|X| \geq n^{1-\frac{\log\log\log n+4}{\log\log n}}$, *such that X contains no three-term arithmetic progression.*

Proof. Let $b = \log n$ and $k = \log n/\log b - 2$. Since $\log n/\log b = \log n/\log\log n$ we have that $k < b/2$ for every $n \geq 8$. We arbitrarily select a subset of k different numbers $\{x_1, \ldots, x_k\} \subset \{0, \ldots, b/2-1\}$ and define $X = \left\{\sum_{i=1}^{k} x_{\pi(i)} b^i \;:\; \pi \text{ is a permutation of } \{1, \ldots, k\}\right\}$. By the definition of X we have that $|X| = k!$. By using $z! > (z/e)^z$, we get that

$$(5.1)\quad |X| \;=\; k! \;=\; \left(\frac{\log n}{\log\log n}\right)! \;>\; n^{1-\frac{\log\log\log n+4}{\log\log n}}$$

Consider any three elements $u, v, w \in X$ such that $u + v = 2w$. By definition of X, these elements are of the form $u = \sum_{i=1}^{k} x_{\pi_u(i)} b^i$, $v = \sum_{i=1}^{k} x_{\pi_v(i)} b^i$ and $w = \sum_{i=1}^{k} x_{\pi_w(i)} b^i \in X$, where π_u, π_v, π_w are permutations over $\{1, \ldots, k\}$. Since $x_i < b/2$ for every $1 \leq i \leq k$, it must be the case that for each i, $x_{\pi_u(i)} + x_{\pi_v(i)} = 2x_{\pi_w(i)}$ This implies that for every i: $x^2_{\pi_u(i)} + x^2_{\pi_v(i)} \geq 2x^2_{\pi_w(i)}$ where the inequality is strict unless $x_{\pi_u(i)} = x_{\pi_v(i)} = x_{\pi_w(i)}$. If we sum over all i's and there is at least one index i for which the inequality is strict we get that $\sum_{i=1}^{k} x^2_{\pi_u(i)} + \sum_{i=1}^{k} x^2_{\pi_v(i)} > \sum_{i=1}^{k} 2x^2_{\pi_w(i)}$ which is a contradiction since we took permutations of the same numbers. Thus, we get that $u = v = w$.

Remark. In fact, the set constructed above is also 3AP-free when all calculations are performed modulo n. We will use this observation below.

Behrend graphs. Given a set $X \subset \{1, \ldots, n\}$ with no three-term arithmetic progression we define the Behrend graph BG_X as follows. It has $3n$ vertices that are partitioned into three equal parts: V_1, V_2, and V_3. For each $i \in \{1, 2, 3\}$ we associate with each vertex in V_i a different integer in $\{0, \ldots, n-1\}$. The edges of the graph are defined as follows:

- The edges between V_1 and V_2: For every $x \in X$ and $j \in \{0, \ldots, n-1\}$ there is an edge between $j \in V_1$ and $(j + x) \bmod n \in V_2$;
- The edges between V_2 and V_3: For every $x \in X$ and $j \in \{0, \ldots, n-1\}$ there is an edge between $(j + x) \bmod n \in V_2$ and $(j + 2x) \bmod n \in V_3$;
- The edges between V_1 and V_3: For every $x \in X$ and $j \in \{0, \ldots, n-1\}$ there is an edge between $j \in V_1$ and $(j + 2x) \bmod n \in V_3$.

We shall say that an edge between $j \in V_1$ and $j' \in V_2$ or between $j \in V_2$ and $j' \in V_3$ is *labeled* by x, if $j' = (j + x) \bmod n$, and we shall say that an edge between $j \in V_1$ and $j' \in V_3$ is *labeled* by x, if $j' = (j+2x) \bmod n$.

The graph BG_X is $2|X|$-regular and it contains $3|X|n$ edges. For every $j \in \{0, \ldots, n-1\}$ and $x \in X$, the graph contains a triangle $(j, (j + x) \bmod n, (j + 2x) \bmod n)$ where $j \in V_1$, $(j + x) \bmod n \in V_2$ and $(j + 2x) \bmod n \in V_3$. There are $n \cdot |X|$ such edge-disjoint triangles and every edge is part of one such triangle. Moreover, it is not hard to verify (based on the assumption that X is 3AP-free) that there are no other triangles in the graph.

5.2 The edge density of large sets in random Behrend graphs

In this subsection we prove the following lemma, which is central to the proof of Lemma 3. We shall use the following notation: For a subset $Y \subseteq X$ and a subset of vertices C in BG_Y, we let $e_Y(C)$ denote the number of edges spanned by C in BG_Y.

LEMMA 5. *Let $0 < \beta < \frac{1}{2}$ and $0 < \alpha \leq 1$ be such that $\alpha - 2\beta > \frac{1}{\log\log n}$, and let $X \subset \{1, \ldots, n\}$, $|X| \geq n^\beta$. Consider the random Behrend graph BG_Y obtained by choosing a random subset $Y \subseteq X$, $|Y| = d = \frac{|X|}{n^\beta}$. With high probability over the choice of Y, for every subset C of vertices in BG_Y where $|C| = n^\alpha$, we have $e_Y(C) \leq \frac{90}{\alpha-2\beta}\frac{n^{2\alpha}}{n^\beta}$ edges.*

The lemma states that for sufficiently large subsets C (i.e., for $|C| = n^\alpha$, where $\alpha - 2\beta$ is a constant), the number of edges $e_Y(C)$ is not much larger than its expected value. Note that the smaller we choose β (i.e., the larger we choose Y), the smaller can α be. Thus, the lemma can be applied to sets with of relatively small size.

Before proving the lemma we introduce some notation and prove two claims. For a subset $W \subseteq V_1 \cup V_2$, $|W| = s$, consider the subgraph of BG_X induced by W. Let

$$\Delta(W) = \{(j_2 - j_1) \bmod n \;:\; j_1 \in W_1,\ j_2 \in W_2, \text{ and } (j_2 - j_1) \bmod n \in X\} \tag{5.2}$$

denote the set of *differences* in W. That is, it is the set of labels of the edges between W_1 and W_2 in BG_X. Obviously, $|\Delta(W)| \leq s^2$. For every difference $x \in \Delta(W)$, we define the *multiplicity* of x in W as the number of edges in BG_X between vertices in W_1 and vertices in W_2 that are labeled by x.

Let $k = \frac{5}{\alpha-2\beta}$. For β and α that satisfy the condition of the lemma ($\alpha - 2\beta > \frac{1}{\log\log n}$) we have that $k \leq 5\log\log n$. We shall say that W is *good* if no difference in $\Delta(W)$ has multiplicity higher than k in W.

CLAIM 6. *With high probability over the choice of $Y \subseteq X$, for every good W such that $|W| = s \geq n^\beta \log n$, we have that $e_Y(W) \leq \frac{2ks^2}{n^\beta}$.*

Claim 6 is easily established by using known bounds on the tail of the Hypergeometric distribution (see, e.g., [11, Page 29]). We also need the following claim.

CLAIM 7. *Let C be a subset of $V_1 \cup V_2$, such that $|C| = n^\alpha$. Suppose we uniformly and independently select $W \subset C$, $|W| = n^\beta \log n$. Then the probability that W is* not good *is at most $\frac{1}{n^\beta}$.*

Proof. Note that by the definition of Behrend graphs, the edges between vertices in C that are labeled by a specific difference, form a matching. When we choose a random subset $W \subset C$, the probability that there exists a single difference of C (i.e. an element of $\Delta(C)$) that has multiplicity at least $k+1$ in W is bounded by $|C|^2|C|^{k+1}\binom{|C|-(2k+2)}{|W|-(2k+2)}\binom{|C|}{|W|}^{-1}$. Now,

$$\frac{|C|^2|C|^{k+1}\binom{|C|-(2k+2)}{|W|-(2k+2)}}{\binom{|C|}{|W|}} \leq |C|^2 \cdot |C|^{k+1} \cdot \left(\frac{|W|}{|C|}\right)^{2k+2}$$

$$\leq n^{2\alpha}\left(\frac{\log^2 n}{n^{\alpha-2\beta}}\right)^{\frac{5}{\alpha-2\beta}} \leq \frac{1}{n^2} \tag{5.3}$$

The last expression is upper bounded by $\frac{1}{n^\beta}$ as required.

Proof of Lemma 5. Consider a set C of vertices of BG_Y such that $|C| = n^\alpha$. Let $C_i = C \cap V_i$ for $1 \leq i \leq 3$, and let $C = C_1 \cup C_2$. We will show that almost surely the number of edges between C_1 and C_2 is at most $\frac{30}{\alpha-2\beta}\frac{n^{2\alpha}}{n^\beta}$. The argument for the number of edges between C_2 and C_3 and between C_1 and C_3 is analogous, and hence the lemma follows. We shall prove the claim for every C_1, C_2 such that $|C_1 \cup C_2| = n^\alpha$. Clearly this implies that it holds for every C_1, C_2, s.t. $|C_1 \cup C_2| \leq n^\alpha$.

By Claim 6, with high probability over the choice of Y the following holds. For every good W, $W \subset C$, such that $|W| = s \geq n^\beta \log n$, the number of edges spanned by W in BG_Y is at most $2ks^2n^{-\beta}$. Assume from now on that the selected Y has this property. We shall use Claim 7 to derive an upper bound on the number of edges in BG_Y that are spanned by the vertices of C.

By our assumption on Y, if W is good and $|W| = s \geq n^\beta \log n$ then $e_Y(W) \leq 2ks^2n^{-\beta}$. Clearly, $e_Y(W) \leq s^2$. If we uniformly at random select $W \subset C$, such that $|W| = s$ then $\mathrm{Exp}[e_Y(W)] \geq \frac{1}{2} \cdot e_Y(C) \cdot \frac{s^2}{n^{2\alpha}}$. We stress that the expectation is taken only over the choice of W and not over the choice of Y. Now,

$$\begin{aligned}
&\mathrm{Exp}[e_Y(W)] \\
&\quad = \mathrm{Exp}[e_Y(W) \mid W \text{ is good }] \cdot \Pr[W \text{ is good }] \\
&\quad + \mathrm{Exp}[e_Y(W) \mid W \text{ is not good }] \cdot \Pr[W \text{ is not good }] \\
&\quad \leq 2ks^2 \cdot n^{-\beta} + s^2 \cdot n^{-\beta} = (2k+1)s^2 \cdot n^{-\beta}
\end{aligned} \tag{5.4}$$

It follows that $e_Y(C) \leq (2k+1) \cdot 2|C|^2 \cdot n^{-\beta} \leq 5k|C|^2 \cdot n^{-\beta}$ Since $k = \frac{5}{\alpha-2\beta}$, the lemma follows.

As a corollary of Lemma 5 we get:

Corollary 8. *Let $0 < \beta < \frac{1}{2}$ and $X \subset \{1,\ldots,n\}$ where $|X| \geq n^{1-\nu(n)}$ for $\nu(n) = \frac{\log\log\log n+4}{\log\log n}$. Consider the random Behrend graph BG_Y obtained by choosing a random subset $Y \subseteq X$, $|Y| = d = \frac{|X|}{n^\beta}$. With high probability over the choice of Y, for every subset C of vertices in BG_Y such that $|C| \leq \min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}$, the following bound applies: $|C| \cdot e_Y(C) \leq n^{1-\nu(n)}$.*

The proof of Corollary 8, which follows from Lemma 5 by a simple case analysis, appears in the full version of this paper [3].

5.3 The lower bound distribution $BG(n, d)$

Let $X \subset [n]$ be a set with no three-term arithmetic progression, as constructed in Subsection 5.1, such that $|X| = n^{1-\nu(n)}$ (where $\nu(n) = \frac{\log\log\log n+4}{\log\log n}$). Consider the Behrend graph, denoted BG_X, whose set of generators is X. Recall that BG_X, which is a graph over $3n$ vertices, contains $|X| \cdot n$ edge-disjoint triangles: every edge belongs to exactly one triangle, and every triangle corresponds to some $x \in X$.

For each subset $Y \subset X$, such that $|Y| = d$ we consider the subgraph of BG_X that contains all its vertices but only the edges labeled by differences $y \in Y$. This (sub-)graph contains $n \cdot |Y| = nd$ edge-disjoint triangles and is hence ϵ-far from being triangle free for any $0 < \epsilon < 1/3$. Next we apply a permutation π on the names of the vertices. More precisely, π consists of 3 permutations, $\pi^{(b)}$, $b \in \{1,2,3\}$, each over $\{0,\ldots,n-1\}$. If we denote each vertex v in BG_X by a pair (b,i) where $b \in \{1,2,3\}$ is the index of the subset that the vertex belongs to and $i \in \{0,\ldots,n-1\}$, then $\pi(v) = \pi(b,i) = (b, \pi^{(b)}(i))$. We denote the resulting graph by $BG_{Y,\pi}$.

A graph is generated according to the distribution $BG(n,d)$ by uniformly selecting Y and π and outputting the resulting graph $BG_{Y,\pi}$. We also assume that the edges incident to a vertex v are ordered randomly in the incidence list of v. For the sake of simplicity, we do not include these random labelings in the notation.

5.4 Proving Lemma 3

In order to prove Lemma 3 it remains to show that any algorithm which is given query access to a graph generated according to $BG(n,d)$ must perform $\Omega\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ queries in order to detect a triangle with probability bounded away from zero. We wish to stress that the algorithm can be much more powerful/sophisticated potentially than just one sampling a random set of the input and looking for a triangle inside. Details can be found in the full version of this paper [3]. Here we provide a sketch.

As in the proof of Lemma 2, we define a process that answers queries of the algorithm while generating a graph according to $BG(n,d)$. In fact, the process will provide the algorithm not only with answers to its neighbor and vertex-pair queries but also with the "identity" of the vertices involved. That is, the algorithm will be provided with $\pi^{-1}(v)$ for vertices v that appeared in its queries or answers to them. The main observation is the following. Let E_t denote the set of edges that the algorithm has observed up till time t. On one hand, by applying Corollary 8 we can upper bound $|E_t|$ (with high probability). On the other hand, since the triangles in the generated graph are edge disjoint, in order to get a triangle, the algorithm must hit one of the $|E_t|$ vertices that closes a triangle with the edges it has observed. Given the definition of the process (which is consistent with the generation of graphs according to $D_\triangle$), we are able to bound the probability that such an event occurs. Further details appear in the full version of this paper [3].

6 From 1-sided error to 2-sided error

In this section we establish that under certain conditions, a one-sided error lower bound for triangle-freeness can be transformed into a two-sided error lower bound. Since these conditions hold for our one-sided error lower bounds, we obtain two-sided error lower bounds.

Theorem 4. *Let $D_\triangle$ be a distribution over graphs with n vertices and average degree d, and let $q(n,d)$ be a function of these parameters. Assume the following holds:*

1. *With probability $1-o(1)$ a graph selected according to $D_\triangle$ is ϵ-far from being triangle-free for some constant ϵ.*
2. *One of the following two conditions holds:*
   - *In all graphs in the support of $D_\triangle$, the triangles are edge-disjoint, and for any algorithm A, the probability that A reveals a triangle in a graph selected according to $D_\triangle$ using $o(q(n,d))$ queries is less than $2/3$.*
   - *For any algorithm A, the probability that A reveals a cycle (of any length) in a graph selected according to $D_\triangle$ using $o(q(n,d))$ queries is less than $2/3$.*

Then any two-sided error algorithm for testing triangle-freeness that has success probability at least $5/6$ must perform $\Omega(q(n/2,d))$ queries.

In what follows we show that if the first condition of Theorem 4 applies, then the theorem holds. The complete proof of this theorem appears in the full version of this paper [3].

Proof. Given the distribution D_Δ we define two distributions over graphs that have $n' = 2n$ vertices and average degree d. One distribution, denoted D'_Δ, generates graphs that are ϵ-far from being triangle free, and the other distribution, denoted $D_{\bar{\Delta}}$ generates graphs that are triangle free. Assume, contrary to what is claimed in the theorem that there exists a two-sided error algorithm A' for testing triangle freeness that performs $o(q(n'/2, d))$ queries and has success probability at least $5/6$. Then, in particular, using $o(q(n'/2, d)) = o(q(n/d))$ queries, A' should be able to distinguish with sufficiently high probability between graphs generated by D'_Δ and graphs generated by $D_{\bar{\Delta}}$. We shall show that we can then use A' to obtain an algorithm A that performs $o(q(n, d))$ queries and with probability at least $2/3$ reveals a triangle in a random graph generated according to D_Δ.

Defining the two distributions. In both distributions, a graph G' over $n' = 2n$ vertices is generated by first selecting a graph G from D_Δ. Every vertex v in G is replaced by two vertices, v_0 and v_1. Every edge (u, v) in G is replaced by two edges: either the two edges (u_0, v_0) and (u_1, v_1) (so that they are "in parallel") or the two edges (u_0, v_1) and (u_1, v_0) (so that they are "crossing"). If v is the j'th neighbor of u and u is the ℓ'th neighbor of v, then in both cases we maintain the ordering on neighbors. Namely, in the case of parallel edges we have that v_0 is the j'th neighbor of u_0 and v_1 is the j'th neighbor of u_1, u_0 is the ℓ'th neighbor of v_0 and u_1 is the ℓ'th neighbor of v_1 (an analogous correspondence holds for crossing edges). The difference between the distributions is in the choice (distribution on the choice) between the above two options.

Recall that the triangles in G are edge-disjoint. Hence, for each triangle in G, the edges between the corresponding vertices in G' can be determined independently from the edges that belong to other triangles. Consider a particular triangle (u, v, w) in G. There are $2^3 = 8$ ways to select the edges between the vertices $u_0, u_1, v_0, v_1, w_0, w_1$ (depending on whether we select parallel or crossing edges). In 4 of these ways we get 2 edge-disjoint triangles (e.g., (u_0, v_0, w_0) and (u_1, v_1, w_1)), and in 4 of these ways we get a single cycle of length 6 (e.g. $(u_0, v_0, w_0, u_1, v_1, w_1)$). The graph generated by D'_Δ simply selects one of the former 4 ways uniformly, and the graph generated by $D_{\bar{\Delta}}$ selects one of the latter 4 ways uniformly.

The basic, but important observation is that for both distributions the following holds: If we consider any edge that belongs to a particular triangle in G, then the probability that the corresponding pair of edges in G' are parallel is equal to the probability that they are crossing. Moreover, this remains true if we condition on any other (single) edge in the triangle being transformed to either parallel or crossing edges. Independence breaks down only when we consider all 3 edges in a triangle. We shall refer to this observation as the *Independence Observation*.

Using a two-sided error algorithm to find triangles. Let A' be a two-sided error algorithm for testing triangle freeness that performs $o(q(n'/2, d))$ queries when testing graphs over $n' = 2n$ vertices and has success probability at least $5/6$. We next show how to use it in order to detect triangles in a graph G over n vertices that is generated randomly according to D_Δ. The idea is that by performing queries to G and flipping some coins, we shall actually be emulating the execution of A' on graphs generated by either D'_Δ or $D_{\bar{\Delta}}$. Since A' is supposed to test graphs over $n' = 2n$ vertices, we denote the vertices in the queries it performs by $\{v_{1,0}, v_{1,1}, \ldots, v_{n,0}, v_{n,1}\}$.

Thus let G be a graph generated according to D_Δ. Algorithm A (whose goal is to detect a triangle in G) runs A' as a subroutine and answers its queries by performing queries to G and transforming the answers to the queries in an appropriate manner described below. In this process A maintains a *knowledge graph*, denoted $\hat{G}$, which contains all the edges it has observed in G as well as the "non-edges" (i.e., pairs (u, v) that do not have an edge between them). In addition, A records all the answers it has already given to A'.

Whenever A' performs a degree query for a vertex $v_{i,b}$ ($b \in \{0, 1\}$), algorithm A queries the degree of v_i and returns it as an answer. Whenever A' performs a vertex-pair query $(v_{i,b}, v_{j,b'})$ ($b, b' \in \{0, 1\}$), if (u, v) is an edge or a non-edge in the knowledge graph $\hat{G}$ then the answer to A' is determined. If this is not the case then A performs the vertex-pair query (v_i, v_j). If the answer is that there is no edge between the two vertices, then the answer given to A' is "no" as well. If the answer is that there is an edge, then there are two cases. If this edge closes a triangle with two other edges in $\hat{G}$ then A terminates successfully. Otherwise, with probability $1/2$ A answers that there is an edge between $v_{i,b}$ and $v_{j,b'}$ and with probability $1/2$ it answers that there is no such edge. In addition, in the former case A' is provided with the information concerning which neighbor is $v_{j,b'}$ of $v_{i,b}$.

Whenever A' performs a neighbor query $(v_{i,b}, \ell)$ (that does not correspond to an edge already in $\hat{G}$), algorithm A performs the neighbor query (v_i, ℓ). Let the answer be (v_j, t). Namely, there is an edge between v_i and v_j, where v_j is the ℓ'th neighbor of v_i, and v_i is the t'th neighbor of v_j. Here too, if a triangle in G is detected then A terminates successfully. Otherwise it answers the query of A' in an analogous manner to the way a vertex-pair query is answered. If A' terminates before A has found a triangle, then A terminates unsuccessfully.

Completing the proof. Since A always terminates when or before it finds a triangle, by the Independence Observation, the distribution on the answers it gives to the queries of A' is exactly the one we would get if the queries of A' were answered by a graph that is selected either according to D'_{Δ} or according to $D_{\bar{\Delta}}$. We claim that this implies that the probability that A' terminates before A finds a triangle (thus causing A to terminate unsuccessfully) is less than $1/3$. Here the probability is taken over the choice of G, the coin flips of A and the possible coin flips of A'.

Assume, contrary to the claim, that the probability that A' terminates before A finds a triangle is at least $1/3$. Consider the distribution over graphs that results from selecting with probability $1/2$ a graph G' according to D'_{Δ}, and with probability $1/2$ a graph G' according to $D_{\bar{\Delta}}$. By our counter-assumption (and the Independence Observation) the probability that A' terminates before it sees three edges of the form (v_{i,b_1}, v_{j,b_2}), (v_{j,b_3}, v_{k,b_4}) and (v_{k,b_5}, v_{i,b_6}) (where $b_1, \ldots, b_6 \in \{0,1\}$) is greater than $1/3$. In such a case, the distribution on the answers to the queries of A' (and hence on its queries conditioned on these answers) is the same if the graph G' is selected according to D'_{Δ} or according to $D_{\bar{\Delta}}$. Therefore, the probability that A' terminates with an incorrect output, is greater than $1/6$. But this contradicts our assumption on A'.

Since the number of queries performed by A before it terminates is upper bounded by the number of queries performed by A', the theorem follows.

Since the distributions that are defined for our one-sided error lower bounds, which are stated in Lemmas 2 and 3, are as required by Theorem 4, we get the following corollary.

Corollary 9. *Any algorithm for testing triangle-freeness must perform*

$$\Omega\left(\max\left\{\min\{d, n/d\}, \min\left\{\sqrt{d}, n^{2/3}/d^{1/3}\right\} \cdot n^{-\nu(n)}\right\}\right)$$

queries. This lower bound holds even if the algorithm is allowed two-sided error and $d_{\max} = O(d)$.

7 Upper bounds

7.1 An upper bound of $O(\sqrt{nd}/\epsilon^{3/2})$ for general graphs

Lemma 10. *It is possible to test triangle-freeness by performing $O(\sqrt{nd}/\epsilon^{3/2})$ queries. If $d_{\max} = O(d)$ then $O(d/\epsilon)$ queries suffice.*

Proof. Let G be a graph with average degree d over n vertices that is ϵ-far from being triangle-free. By definition, G must contain at least ϵnd edges that belong to triangles. If $d_{\max} = O(d)$ then by uniformly selecting $\Theta(1/\epsilon)$ vertices and for each uniformly selecting an incident edge, with high probability we obtain an edge that belongs to a triangle. Conditioned on this event, if we now perform all $O(d)$ neighbor queries to the end-points of each selected edge, we reveal a triangle.

If the maximum degree of the graph differs significantly from its average degree, then the above argument cannot be applied: First, the suggested edge selection process might not select with sufficiently high probability an edge that belongs to a triangle. Second, even if we obtain such an edge, its end-points might have a very high degree. To address these issues, we first introduce some notation.

We say that a vertex has *high degree* if its degree is more than $c\sqrt{nd}$ (where we set c momentarily). We shall say that an edge is *covered* by these high degree vertices, if *both* its end-points have high degree. But the high-degree vertices can cover at most $\left((1/c)\sqrt{nd}\right)^2 = (1/c^2)nd$ edges. Hence, among the edges that belong to triangles, there are at least $(\epsilon - (1/c^2))nd$ edges that have at least one end-point with degree at most $c\sqrt{nd}$. If we set $c = \sqrt{2/\epsilon}$ then we have at least $(\epsilon/2)nd$ such edges.

In order to obtain one of these edges, we would like to sample edges uniformly in G. In fact, it suffices to sample edges "almost uniformly" as defined in [12]. In [12] an algorithm is described that uses $\tilde{O}(\sqrt{n/\delta})$ queries to a graph G and for which the following holds: For all but at most $\delta/4$-fraction of the edges of G the probability that the edge is selected is at least $\frac{1}{32nd}$. We refer to this algorithm as "Edge-Select". By definition of the algorithm, if we set $\delta = \epsilon$, we get that there are at least $(\epsilon/4)nd$ edges that can be returned by "Edge-Select" such that these edges belong to triangles and have at least one end-point with degree at most $\sqrt{2/\epsilon}\sqrt{nd}$. It follows that at a cost of $O(\sqrt{n}/\epsilon^{3/2})$ queries we obtain such an edge with a high constant probability. Thus the algorithm for detecting a triangle runs "Edge-Select" $\Theta(1/\epsilon)$ times. For each selected edge, if it has one end-point with degree less than $\sqrt{2/\epsilon} \cdot \sqrt{nd}$ then it asks all neighbor queries for that vertex, and for each of them it asks all pair queries with the other end point. (If both end-points have high degree then the algorithm does nothing).

7.2 An improved upper bound for relatively dense general graphs

Lemma 11. *It is possible to test triangle-freeness by performing* $O\left(\max\left\{\frac{n^{4/3}}{\epsilon^{2/3}d^{2/3}}, \frac{d_{\max}^2}{\epsilon^2 d^2}\right\}\right)$ *queries.*

Corollary 12. *It is possible to test triangle-freeness of graphs with average degree* $d = \Omega(\sqrt{n})$ *by performing* $O\left(\frac{n^{4/3}}{d^{2/3}\epsilon^2}\right)$ *queries.*

Proof. Let G be a graph over n vertices with average degree d and maximum degree $d_{\max}$ that is ϵ-far from being triangle-free. We shall show that if we take a uniform sample of $\Theta\left(\max\left\{\frac{n^{2/3}}{\epsilon^{1/3}d^{1/3}}, \frac{d_{\max}}{\epsilon d}\right\}\right)$ vertices of G, and ask vertex-pair queries between all pairs in the sample, then a triangle is detected with probability at least 2/3.

Since G is ϵ-far from being triangle-free, it must contain at least ϵnd triples of vertices that form a triangle. This lower bound on the number of triangles implies that the expected number of triangles in a set of s uniformly selected vertices is at least $s^3 \cdot \frac{\epsilon nd}{n^3}$. It follows that for $s \geq n^{2/3}/(\epsilon d)^{1/3}$, the expected number of triangles spanned by the sample is at least 1. This unfortunately does not imply in general that a uniform sample of $s = \Omega(n^{2/3}/(\epsilon d)^{1/3})$ vertices spans a triangle with probability at least 2/3. Rather, the size of the sample should depend on the ratio between $d_{\max}$ and d.

Let $s = c \cdot \max\left(\frac{n^{2/3}}{(\epsilon d)^{1/3}}, \frac{d_{\max}}{\epsilon d}\right)$, where $c > 0$ is a sufficiently large constant. Since G is ϵ-far from being triangle-free, it easily follows that G must contain a family T of $(\epsilon nd)/3$ pairwise edge-disjoint triangles. Fix such a family, and for every $v \in V(G)$, let $d_T(v)$ be the number of triangles in T containing v; obviously, $d_T(v) \leq d(v)/2 \leq d_{\max}/2$. We sample a set S of s vertices of G uniformly at random. Let X be the random variable counting the number of triangles of T spanned by S. Due to the Chebyshev inequality, it is enough to prove that $\mathrm{Exp}[X]$ is at least a large constant, and the ratio $\mathrm{Var}[X]/\mathrm{Exp}^2[X]$ is at most a small enough constant. We will estimate both quantities.

Observe that each triangle of T falls into S with probability $(1+o(1))s^3/n^3$. It follows that

$$\mathrm{Exp}[X] = (1+o(1))\frac{s^3}{n^3}|T| = \Theta\left(\frac{\epsilon d s^3}{n^2}\right) . \tag{7.5}$$

Thus, taking c large enough, we get: $\mathrm{Exp}[X]$ is large enough, too. Also,

$$\begin{aligned} \mathrm{Var}[X] &\leq \sum_{\substack{t,t' \in T \\ t \cap t' \neq \emptyset}} \Pr[t, t' \subset S] \\ &= \sum_{v \in V(G)} \binom{d_T(v)}{2} \frac{(1+o(1))s^5}{n^5} \qquad (7.6) \end{aligned}$$

(the latter estimate is due to the fact that, since T is pairwise edge-disjoint, for any $t, t' \in T$ with $t \cap t' \neq \emptyset$, the union $t \cup t'$ contains exactly five vertices). Recall that $d_T(v) \leq d_{\max}$. Due to convexity, we get:

$$\mathrm{Var}[X] = O\left(\frac{\epsilon nd}{d_{\max}} \cdot d_{\max}^2\right)\frac{s^5}{n^5} . \tag{7.7}$$

Using the assumption $s = \Omega\left(\frac{d_{\max}}{\epsilon d}\right)$, we derive: $\mathrm{Var}[X]/\mathrm{Exp}^2[X]$ is small enough, as required.

References

[1] N. Alon. Testing subgraphs in large graphs. *Random Structures and Algorithms*, 21(3–4):359–370, 2002.

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

[3] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle-freeness in general graphs. Available from http://www.math.tau.ac.il/~krivelev, 2005.

[4] N. Alon and A. Orlitsky. Repeated communication and Ramsey graphs. *IEEE Transactions on Information Theory*, 41:1276–1289, 1995.

[5] N. Alon and Y. Roichman. Random Cayley graphs and expanders. *Random Structures and Algorithms*, 5:271–284, 1994.

[6] N. Alon and A. Shapira. A characterization of easily testable induced subgraphs. In *Proceedings of the 15th SODA*, 2004.

[7] N. Alon and A. Shapira. Testing subgraphs in directed graphs. *JCSS*, 69:354–482, 2004.

[8] F. A. Behrend. On sets of integers which contain no three terms in arithmetic progression. *Proc. National Academy of Sciences USA*, 32:331–332, 1946.

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

[10] B. Green. Counting sets with small sumset, and the clique number of random Cayley graphs. *Combinatorica*, 25:307–326, 2005.

[11] S. Janson, T. Łuczak, and A. Ruciński. *Random graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

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

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

[14] I. Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely), Coll. Math. Soc. J. Bolyai 18*, 2:939–945, 1976.

[15] R. Salem and D. C. Spencer. On sets of integers which contain no three terms in arithmetical progression. *Proc. National Academy of Sciences USA*, 28:561–563, 1942.