# Resolution Complexity of Independent Sets in Random Graphs

Paul Beame*
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
beame@cs.washington.edu

Russell Impagliazzo†
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114
russell@cs.ucsd.edu

Ashish Sabharwal*
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
ashish@cs.washington.edu

## Abstract

*We consider the problem of providing a resolution proof of the statement that a given graph with $n$ vertices and $\Delta n$ edges does* not *contain an independent set of size $k$. For randomly chosen graphs with constant $\Delta$, we show that such proofs almost surely require size exponential in $n$. Further, for $\Delta = o(n^{1/5})$ and any $k \leq n/5$, we show that these proofs almost surely require size $2^{n^\delta}$ for some global constant $\delta > 0$, even though the largest independent set in graphs with $\Delta \approx n^{1/5}$ is much smaller than $n/5$. Our result shows that almost all instances of the independent set problem are hard for resolution. It also provides a lower bound on the running time of a certain class of search algorithms for finding a largest independent set in a given graph.*

## 1. Introduction

The problem of determining if a given graph contains an independent set of a certain size is NP-complete and thus the dual problem of determining non-existence of independent sets of that size in a given graph is co-NP-complete. For a graph chosen at random, its independent sets have nice combinatorial properties; the size of the largest such independent set can be described in terms of simple graph parameters [6, 15]. This gives us a good range of sizes such that graphs chosen at random almost surely do not have an independent set of size in this range. We study the problem of proving this fact under a fixed proof system.

Influenced by algorithms of Tarjan and Trojanowski [19, 20] for finding maximum independent sets, Chvátal [8] devised a specialized proof system for the independent set problem and proved almost certain exponential lower bounds for proofs of non-existence of large independent sets in random graphs with a linear number of edges in this system. Chvátal's system also captures the more recent improved algorithms of Jian [16] and Robson [18], the latter of which has the current record for such algorithms.

Given a graph $G$ and an integer $k$, we consider encoding the existence of an independent set of size $k$ in $G$ as a CNF formula and examine the proof complexity of such formulas under resolution proofs. Resolution on one of the encodings we present captures the behavior of a fairly broad class of search algorithms on the corresponding graphs. In particular, the bounds we prove for resolution complexity also imply similar lower bounds in Chvátal's system. We show that given a randomly chosen graph with not too many edges, almost surely a resolution proof of the statement that this graph does not have an independent set of a certain size must be exponential. This gives us an exponential lower bound on the running time of a natural class of algorithms for searching for a largest independent set in a given graph. This is also interesting because it shows lower bounds for random formulas with significant structure as opposed to the unstructured random $k$-CNF formulas for which resolution bounds have previously been shown [9, 3]. In this sense, it adds to the random graph $k$-coloring exam-

* Research supported by NSF Award CCR-9800124
† Research supported by NSF Award CCR-9734911



1093-0159/01 $10.00 © 2001 IEEE

ple for which exponential resolution lower bounds are already known [2].

Instead of looking at the general problem of disproving the existence of *any* large independent set in a graph, we focus only on a restricted class of independent sets that we call block-respecting independent sets. We prove that even ruling out this smaller class of independent sets requires exponential size resolution proofs. These restricted independent sets are simply the ones obtained by dividing the $n$ vertices of the given graph into $k$ blocks of equal size (assuming $n$ is a multiple of $k$) and choosing one vertex from each block. Since it is easier to rule out a smaller class of independent sets, the lower bounds we obtain for the restricted version are stronger in the sense that they imply lower bounds for the general problem. Further, the independent set problem encoded in terms of block-respecting independent sets also captures Chvátal's proof system and hence our lower bounds also apply to proofs in his system, as well as to a number of algorithms that his system captures [16, 18, 19, 20]. Towards the end, we give a simple upper bound for the general problem based on expected sizes of independent sets in random graphs.

Most known resolution complexity lower bounds can be proved using a general technique that is shown by Ben-Sasson and Wigderson [5] and is derived from earlier papers by Haken [13] and Clegg, Edmonds and Impagliazzo [10]. It is based on a relationship between the size of resolution proofs and the *width*, the length of the longest clause, in such proofs. It uses the property that any resolution proof for a given problem must contain clauses that are minimally implied by a middle-size fraction of the relevant input clauses and that any such derived clauses must have large width. Therefore we begin by analyzing the width of any resolution proof saying that a graph does not have an independent set of a certain size and then apply the width-size relationship of [5] to get good lower bounds on tree-like and general resolution proof sizes.

The proof of the width bound can be broadly divided into two parts, both of which use the fact that random graphs are almost surely locally sparse. We first show that the minimum number $s$ of input clauses that are needed for any refutation of the problem is large for most graphs. We then use combinatorial properties of independent sets in random graphs to say that any clause minimally implied by a middle-size subset of these $s$ clauses has to be large. These together allow us to say that the width of any such proof has to be large.

## 2. Resolution and DLL Proofs

A propositional formula $F$ is said to be in conjunctive normal form (CNF) if it is a conjunction of *clauses*, where each clause is a disjunction of *literals* and each literal is either a variable or its negation. Resolution is a very simple proof system for CNF formulas. It forms the basis of most popular systems for practical theorem proving. Lower bounds on resolution proof sizes thus have a bearing on the running time of these algorithms. The construction of Tseitin [21] can be used to efficiently convert any given propositional formula to one in CNF form. Hence we don't lose much by restricting ourselves to CNF formulas.

### 2.1. General Resolution

To prove that a given input CNF formula $F$ is unsatisfiable, we start with the original input clauses of $F$ and repeatedly pick pairs of clauses to apply the *resolution rule*: given $(A \vee x)$ and $(B \vee \neg x)$, one can derive $(A \vee B)$. It is clear that a derived clause will be satisfied by any assignment that satisfies both the parent clauses. Since we are interested in proving that $F$ is unsatisfiable, the goal is to start with $F$ and derive the empty (and trivially unsatisfiable) clause $\Lambda$.

Let $\phi$ be a set of clauses. A *resolution derivation* from $\phi$ is a sequence of clauses $\pi = C_1, C_2, \ldots, C_s$ where each clause $C_i$ is either an element of $\phi$ or is derived by applying the resolution rule to two clauses $C_j$ and $C_k$, $j, k < i$, occurring earlier in $\pi$. A resolution derivation of the empty clause $\Lambda$ from $\phi$ is called a *refutation* or *proof* of $\phi$. Given any resolution refutation, we can associate with it in a natural way a directed acyclic graph where edges go from parent clauses to the clause obtained by resolving them on a certain literal. The special case where this graph is a tree is referred to as *tree-like* resolution and is discussed in section 2.2.

Let $F$ be a set of clauses encoding a given problem as a CNF formula and $P$ a resolution proof. The *size* of the proof $P$, $s(P)$, is the number of clauses appearing in $P$. Define the *resolution complexity of* $F$, $Res(F)$, to be the minimum of $s(P)$ over all proofs $P$ of $F$; if no such proofs exist we define $Res(F) = \infty$. As shown in [5] and we will see in section 2.3, resolution complexity is intimately related to another measure of proofs, their width. Let the *width* of a clause be the number of literals occurring in it. Given $F$ and proof $P$ we define $w(F)$ and $w(P)$ to be the maximum of the widths of all clauses in $F$ and $P$, respectively. Now define width$(F)$ to be the minimum of $w(P)$ over all proofs $P$ of $F$. To prove a lower bound on $Res(F)$, it is sufficient to prove a lower bound on width$(F) - w(F)$.

### 2.2. Davis-Putnam (DLL) Procedure and Tree-like Resolution

The Davis-Putnam or DLL procedure [12] is both a proof system and a collection of algorithms for finding proofs. A simple Davis-Putnam algorithm to refute a CNF formula $F$ is to repeatedly pick a variable $x$ of $F$ and recursively refute both $F|_{x \leftarrow 0}$ and $F|_{x \leftarrow 1}$. Variants of this algorithm form the most widely used family of complete algorithms for formula satisfiability. As a proof system, the DLL procedure forms a special case of resolution where the proof graph is a tree, that is, any clause that is used more than once in the proof must be re-derived. The *tree-resolution complexity* of a given formula $F$ is defined to be the minimum of $s(P)$, the size of $P$, over all *tree-like* resolution proofs $P$ of $F$.

We can think of DLL refutations as trees where we branch at each node based on the value of a variable. DLL refutations can be easily converted into tree-like resolution proofs of essentially the same size, and vice versa (see, for example, [3]). Given this correspondence, we will use these two terms interchangeably and denote the tree-resolution complexity of a formula $F$ by $DLL(F)$. This correspondence also shows that a lower bound on the size of tree-like resolution proofs also gets us a lower bound on DLL refutation sizes and hence on the running time of DLL type algorithms. The transcript of any algorithm for finding a largest independent set in a given graph is also a proof that no independent set of a larger size exists. Our results show that the running time of any such DLL type algorithms working on randomly chosen input graphs with not too many edges will almost certainly be exponential.

Even though we described DLL algorithms here as working on propositional formulas, they capture a much more general class of algorithms that are based on branching and backtracking. For instance, basic algorithms for finding a largest independent set such as that of Tarjan [19] branch on each vertex by either including it in the current independent set and deleting itself and all its neighbors from further consideration, or excluding it from the current independent set and recursively finding a largest independent set in the remaining graph. More complicated algorithms such as that of Tarjan and Trojanowski [20] branch in a similar manner on small subsets of vertices, reusing subproblems already solved. Such algorithms also fall under the category of resolution type (not necessarily tree-like) algorithms and our lower bounds apply to them as well.

### 2.3. The Width-Size Relationship

Our proof uses the relationship between the size and the width of a resolution proof given by Ben-Sasson and Wigderson [5]. Let $F$ be any unsatisfiable CNF formula over $n$ variables with initial width $w(F)$. Ben-Sasson and Wigderson showed that any short proof of unsatisfiability of $F$ can be converted to one of small width, thus showing that a lower bound on the width of any resolution proof gets us a lower bound on the size of such a proof.

**Proposition 1 ([5]).** $DLL(F) \geq 2^{\text{width}(F)-w(F)}$.

**Proposition 2 ([5]).** *For* $c = 1/(9 \ln 2)$, $Res(F) \geq 2^{c(\text{width}(F)-w(F))^2/n}$.

## 3. Independent Sets

For any undirected graph $G = (V, E)$, let $n \stackrel{\text{def}}{=} |V|$ and $m \stackrel{\text{def}}{=} |E|$. A *k-independent set* in $G$ is a set of $k$ vertices no two of which are adjacent. We will describe reasonable ways of encoding in clausal form the statement that $G$ has a $k$-independent set. Their refutations will then be proofs that $G$ does *not* contain any $k$-independent set. We will be interested in size bounds for such proofs. Our results will be in terms of $n$, $k$ and the graph edge density $\Delta$ defined as $\Delta \stackrel{\text{def}}{=} m/n$.

### 3.1. Random Graph Models

Combinatorial properties of random graphs have been studied well, for instance in [6, 15]. We say a property holds *almost certainly* in a random graph if it holds with probability $1 - o(1)$ in the number of vertices when the graph is selected at random according to some distribution. A property of a graphs is called *monotone* if adding more edges to a graph that has the property cannot prevent it from having that property anymore. It is called *anti-monotone* if deleting edges from the graph that has this property cannot prevent it from having the property anymore. Thus containing an independent set of a certain size is an anti-monotone property.

There are various models saying how to pick a graph with $n$ vertices at random. One could choose a graph from the distribution $\mathcal{G}_{n,p}$ where each of the $\binom{n}{2}$ edges is chosen independently with probability $p$. The resulting graph has $m$ edges on average if $p = m/\binom{n}{2}$. One could also use the distribution $\mathcal{G}_n^m$ where each of the $\binom{\binom{n}{2}}{m}$ graphs with $m$ edges have equal probability of being chosen. This will guarantee that the resulting graph has exactly $m$ edges. A third distribution, which we denote by $\mathcal{G}_{n,m}$, is to pick $m$ edges uniformly at random with replacement and ignore duplicates. The resulting graph has $m - o(m)$ edges with probability $1 - o(1)$ in $n$.

As shown, for example, in [1], if $p = m/\binom{n}{2}$, then the almost certain monotone and anti-monotone properties of graphs are the same under all three models up to a change from $m$ to $m \pm o(m)$. The third distribution $\mathcal{G}_{n,m}$ is the easiest to use in our case. We will therefore use this throughout, but our results also apply to $\mathcal{G}_{n,p}$ and $\mathcal{G}_n^m$. We will use the notation $G \sim \mathcal{G}_{n,m}$ to denote a graph $G$ picked randomly from this distribution.

### 3.2. Independent Set Sizes

For a graph $G$ with $n$ vertices and $\Delta n$ edges, the average degree over all vertices is $2\Delta$. At least half of these vertices, the ones with low degree, must have degree at most twice the average, otherwise the high-degree half fraction of vertices by themselves will contribute more than $2\Delta$ to the average. Consider the subgraph $G'$ induced by these $1/2$ low-degree vertices. This subgraph $G'$ must have maximum degree at most $4\Delta$. Let us repeatedly apply the following procedure to $G'$ until no vertices are left: pick a vertex $v$ of $G'$ arbitrarily, put it in a set $I$ and remove $v$ and all its neighbors from $G'$. It is easy to see that the set $I$ is an independent set of $G'$ of size at least $\frac{n/2}{1+4\Delta}$ – each step removes a vertex and at most $4\Delta$ neighbors of it. Moreover, since $G'$ is an induced subgraph of $G$, $I$ is also an independent set of $G$. This gives us the following simple bound:

**Observation 1.** *Any graph with $n$ vertices and $\Delta n$ edges has an independent set of size $\frac{n}{2+8\Delta}$.*

Sizes of largest independent sets in random graphs are in fact known to high accuracy [6, 15]. For $\epsilon > 0$, let $k_{\pm\epsilon}$ be defined as follows:

$$k_{\pm\epsilon} = \lfloor \frac{2n}{\Delta}(\ln \frac{\Delta}{\ln \Delta} + 1 - \ln 2 \pm \epsilon) \rfloor$$

Then we know the following:

**Proposition 3 ([15]).** *For any $\epsilon > 0$, there exists a constant $C_\epsilon$ such that for $C_\epsilon \leq \Delta \leq n/\log^2 n$, asymptotically almost all graphs chosen at random from $\mathcal{G}_{n,\Delta n}$ have the largest independent set size between $k_{-\epsilon}$ and $k_{+\epsilon}$.*

## 4. Encoding Independent Sets as Formulas

In order to use a propositional proof system to prove that a graph does not have an independent set of a particular size, we first need to formulate the problem as a propositional formula. This is complicated by the difficulty of counting set sizes using CNF formulas. One natural way to encode the independent set problem is to have variables that say which vertices are in the independent set and auxiliary variables that count the number of vertices in this independent set. We will discuss this encoding in section 4.1. The existence of two different types of variables makes this encoding more difficult to reason about directly. A second encoding, derived from this counting-based encoding, is described in section 4.2. It uses a mapping from the vertices of the graph to $k$ additional nodes as an alternate to straightforward counting and uses variables of only one type. This is essentially the same encoding as the one used by Bonet, Pitassi and Raz [7] for the clique problem, except that in our case we need to add an extra set of clauses called ordering clauses to make the lower bounds non-trivial. Section 4.3 finally describes a much simpler encoding which we analyze directly for our lower bounds. This encoding discusses only a restricted class of independent sets that we call *block-respecting independent sets*, for which the problem of counting the set size is trivial so it has only one type of variables that say which vertices are in the independent set. Refutation of this third encoding rules out the existence only of this smaller class of independent sets. Intuitively, this should be easier to do than ruling out all possible independent sets. In fact, we show that its resolution and DLL refutations are bounded in size by those of the mapping encoding and at worst a small amount larger than those of the counting encoding, so we can translate our lower bounds for this third encoding to each of the other encodings. Further, we give upper bounds for the two general encodings which also apply to the simpler block-respecting independent set encoding.

We will identify the vertex set of the input graph with $\{1, 2, \ldots, n\}$. Each encoding will be defined over variables from one or more of the following three categories:

1. $x_v, 1 \leq v \leq n$, which will be TRUE iff vertex $v$ is chosen by the truth assignment to be in the independent set,
2. $y_{v,i}, 0 \leq i \leq v \leq n$, $0 \leq i \leq k$, which will be TRUE iff precisely $i$ of the first $v$ vertices are chosen in the independent set, and
3. $z_{v,i}, 1 \leq v \leq n, 1 \leq i \leq k$, which will be TRUE iff vertex $v$ is chosen as the $i^{th}$ node of the independent set.

### 4.1. Encoding Based on Counting

We construct the *counting encoding* $\alpha_{count}(G, k)$ of the independent set problem over variables $x_v$ and $y_{v,i}$ using the following three kinds of clauses:

**Edge Clauses** For each edge $(u, v)$, $\alpha_{count}(G, k)$ has one clause saying that not both $u$ and $v$ are selected; $\forall (u, v) \in E : (\neg x_u \vee \neg x_v) \in \alpha_{count}(G, k)$

**Size-$k$ Clause** There is a clause saying that the independent set chosen is of size $k$; $y_{n,k} \in \alpha_{count}(G, k)$

**Counting Clauses** There are clauses saying that variables $y_{v,i}$ correctly count the number of vertices chosen. For simplicity, we first write this condition not as a set of clauses but as more general propositional formulas. For the base case, $\alpha_{count}(G, k)$ contains $y_{0,0}$ and the clausal form of $(y_{v,0} \leftrightarrow (y_{v-1,0} \wedge \neg x_v))$ for $v \in \{1, \ldots n\}$. Further, $\forall i, v, 1 \leq i \leq v \leq n$, $1 \leq i \leq k$, $\alpha_{count}(G, k)$ contains the clausal form of $(y_{v,i} \leftrightarrow ((y_{v-1,i} \wedge \neg x_v) \vee (y_{v-1,i-1} \wedge x_v))$ unless $i = v$, in which case $\alpha_{count}(G, k)$ contains the clausal form of the simplified $(y_{i,i} \leftrightarrow (y_{i-1,i-1} \wedge x_i))$.

Translated into clauses, these conditions take the following form. Formulas defining $y_{v,0}$ for $v \geq 1$ translate into $\{(\neg y_{v,0} \vee y_{v-1,0}), (\neg y_{v,0} \vee \neg x_v), (y_{v,0} \vee \neg y_{v-1,0} \vee x_v)\}$. Further, formulas defining $y_{v,i}$ for $v > i \geq 1$ translate into $\{(y_{v,i} \vee \neg y_{v-1,i} \vee x_v), (y_{v,i} \vee \neg y_{v-1,i-1} \vee \neg x_v), (\neg y_{v,i} \vee y_{v-1,i} \vee y_{v-1,i-1}), (\neg y_{v,i} \vee y_{v-1,i} \vee x_v), (\neg y_{v,i} \vee y_{v-1,i-1} \vee \neg x_v)\}$ whereas in the case $i = v$ they translate into $\{(\neg y_{i,i} \vee y_{i-1,i-1}), (\neg y_{i,i} \vee x_i), (\neg x_i \vee \neg y_{i-1,i-1} \vee y_{i,i})\}$.

## 4.2. Encoding Based on Mapping

This encoding, denoted $\alpha_{map}(G, k)$, uses a mapping from $n$ vertices of $G$ to $k$ nodes of the independent set as an indirect way of counting the number of vertices chosen by a truth assignment to be in the independent set. It can be viewed as a set of constraints restricting the mapping (see Figure 1). It is defined over variables $z_{v,i}$ and has the following four kinds of clauses:

**Edge Clauses** For each edge $(u, v) \in E$, there is one clause saying that not both $u$ and $v$ are chosen in any independent set; $\forall (u, v) \in E, i, j \in \{1, \ldots, k\}, i \neq j : (\neg z_{u,i} \vee \neg z_{v,j}) \in \alpha_{map}(G, k)$

**Surjective Clauses** For each $i, 1 \leq i \leq k$, there is a clause saying that some vertex is chosen as the $i^{th}$ node of the independent set; $\forall i \in \{1, \ldots, k\} : (z_{1,i} \vee z_{2,i} \vee \ldots \vee z_{n,i}) \in \alpha_{map}(G, k)$

**Function Clauses** For each vertex, there are clauses saying that this vertex is not mapped to two nodes, i.e. it is not counted twice in the independent set; $\forall v \in \{1, \ldots, n\}, i, j \in \{1, \ldots, k\}, i \neq j : (\neg z_{v,i} \vee \neg z_{v,j}) \in \alpha_{map}(G, k)$

**1-1 Clauses** For each node in the independent set, there are clauses saying no two vertices map to this node; $\forall i \in \{1, \ldots, k\}, u, v \in \{1, \ldots, n\}, u \neq v : (\neg z_{u,i} \vee \neg z_{v,i}) \in \alpha_{map}(G, k)$

**Ordering Clauses** For every pair of consecutive vertices, there is a clause saying that these are not mapped in the reverse order. This, by associativity, implies that there is a unique mapping to $k$ nodes once we have chosen $k$ vertices to be in the independent set. $\forall u, v \in \{1, \ldots, n\}, i \in \{1, \ldots, k-1\}, u < v : (\neg z_{u,i+1} \vee \neg z_{v,i}) \in \alpha_{map}(G, k)$.

This encoding differs in spirit from the clique encoding of Bonet, Pitassi and Raz [7] only in that it has additional ordering clauses. By omitting such clauses, the encoding in [7] merely stated that there is a bijection from the independent set to a set of size $k$. Since they worked in the more powerful Cutting Planes proof system this method of counting was easy to reason about. However, in resolution, the well-known exponential lower bounds for the pigeonhole principle [13, 4, 17] imply that refuting such an encoding even for finding large independent sets in a trivial graph would be exponentially hard for resolution. Adding the ordering clauses makes counting easy and ensures that any lower bound we prove says something about the hardness of the independent set problem as a graph problem rather than merely as a problem of counting.

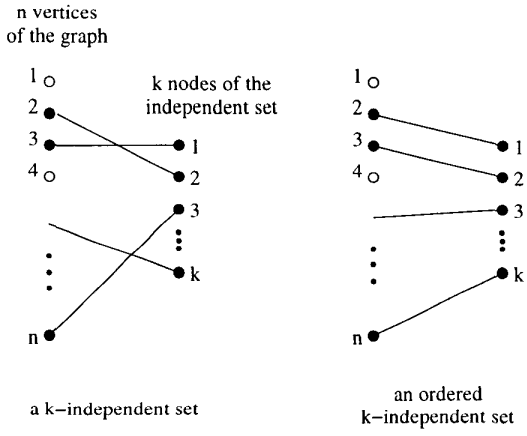


**Figure 1. Viewing independent sets as a mapping from $n$ vertices to $k$ nodes**

### 4.3. Encoding Using Block-respecting Independent Sets

We define a restricted class of independent sets which we call *block-respecting independent sets*. We will fix $b = n/k$ for the rest of the paper and assume for simplicity that $k$ divides $n$. Partition the vertices of $G$ into $k$ subsets of size $b$ each. We will refer to these subsets as *blocks*. A block-respecting independent set of size $k$ is an independent set in which precisely one vertex comes from each of these $k$ blocks (see Figure 2). The definition implicitly assumes a partitioning of the vertices of $G$ into $k$ equal size blocks. We note that the restriction that $k$ divides $n$ is only to make the presentation simple. We can extend this argument to all $k < n$ by letting each block have either $b$ or $b+1$ vertices for $b = \lfloor n/k \rfloor$. The calculations are nearly identical to what we present here. Clearly, if a graph does not contain *any* $k$-independent set, then it certainly does not contain any block-respecting independent set of size $k$ either.

We now define a CNF formula $\alpha_{block}(G,k)$ over variables $x_v$ which says that $G$ contains a block-respecting independent set. We will fix an arbitrary ordering of vertices such that the first $b$ vertices form the first block, the second $b$ vertices form the second block, and so on. Henceforth, in all references to $G$, we will implicitly assume this fixed ordering of vertices and division into $k$ blocks. Since this ordering is chosen arbitrarily, the bounds we derive hold for any ordering. $\alpha_{block}(G,k)$ contains the following three kinds of clauses:

**Edge Clauses** For each edge $(u,v)$, there is one clause saying that not both $u$ and $v$ are selected; $\forall (u,v) \in E:\ (\neg x_u \vee \neg x_v) \in \alpha_{block}(G,k)$

**Block Clauses** For each block, there is one clause saying that at least one of the vertices in the block is selected; $\forall\, i \in \{0,1,\ldots,k-1\}, (x_{bi+1} \vee x_{bi+2} \vee \ldots \vee x_{bi+b}) \in \alpha_{block}(G,k)$

**1-1 Clauses** For each block, there are clauses saying that at most one of the vertices in the block is selected; $\forall\, i \in \{0,\ldots,k-1\}, j,l \in \{1,\ldots,b\}, j \neq l:\ (\neg x_{bi+j} \vee \neg x_{bi+l}) \in \alpha_{block}(G,k)$

Clearly $\alpha_{block}(G,k)$ is satisfiable iff $G$ has a block-respecting independent set of size $k$.

### 4.4 Relationships Among Encodings

**Lemma 1.** *For any graph $G$ on $n$ vertices and any integer $k$ dividing $n$,*

$$Res(\alpha_{block}(G,k)) \leq b^2\, Res(\alpha_{count}(G,k)), \quad \textit{and}$$

$$DLL(\alpha_{block}(G,k)) \leq (2\, DLL(\alpha_{count}(G,k)))^{\log_2 2b}.$$

*Proof.* Resolution proofs of $\alpha_{count}(G,k)$ can be converted into proofs of $\alpha_{block}(G,k)$ by applying a restriction to the variables. We do this by starting with a fixed resolution proof of $\alpha_{count}(G,k)$ and setting some of the variables so that its initial clauses either transform into initial clauses of $\alpha_{block}(G,k)$ or are satisfied trivially. For each $i \in \{0,1,\ldots,k\}$, we simplify the proof by setting $y_{bi,i}$ = TRUE and $y_{bi,j}$ = FALSE for $j \neq i$. We also set all $y_{v,i}$ = FALSE if vertex $v$ does not belong to either block $i+1$ or block $i$ (no vertex belongs to block 0). Finally, for $1 \leq j \leq b$, we replace all occurrences of $y_{bi+j,i+1}$ and $\neg y_{bi+j,i}$ in the proof with $(x_{bi+1} \vee x_{bi+2} \vee \ldots \vee x_{bi+j})$ and all occurrences of $\neg y_{bi+j,i+1}$ and $y_{bi+j,i}$ with $(x_{bi+j+1} \vee x_{bi+j+2} \vee \ldots \vee x_{bi+b})$. Note that setting $y_{bi,i}$ = TRUE for each $i$ logically implies the rest of the restrictions we stated.

The edge clauses are the same in both encodings. The size-$k$ clause $y_{n,k}$ and the counting clause $y_{0,0}$ of $\alpha_{count}(G,k)$ are trivially satisfied. The following can also be easily verified by plugging in the substitutions for the $y$ variables. The counting clauses that define $y_{v,0}$ for $v \geq 1$ are either satisfied or translate into the first block clause $(x_1 \vee \ldots \vee x_b)$. Further, the counting clauses that define $y_{v,i}$ for $v \geq 1, i \geq 1$ are either satisfied or transform into the $i^{th}$ or the $(i+1)^{st}$ block clause, i.e. into $(x_{b(i-1)+1} \vee \ldots \vee x_{b(i-1)+b})$ or $(x_{bi+1} \vee \ldots \vee x_{bi+b})$. Hence, all initial clauses of $\alpha_{count}(G,k)$ are either trivially satisfied or transform into initial clauses of $\alpha_{block}(G,k)$.

Note that the substitutions for $y_{bi+j,i+1}$ and $y_{bi+j,i}$ replace these variables by a disjunction of at most $b$ positive literals. Any resolution step performed on these $y$'s in the original proof must now be converted into a set of equivalent resolution steps, which will blow up the size of the transformed refutation. More specifically, a step resolving clauses $(y \vee A)$ and $(\neg y \vee B)$ on the literal $y$ (where $y$ is either $y_{bi+j,i+1}$ or $y_{bi+j,i}$) will now be replaced by a set of resolution steps deriving $(A' \vee B')$ from clauses $(x_{u_1} \vee \ldots \vee x_{u_p} \vee A')$ and $(x_{v_1} \vee \ldots \vee x_{v_q} \vee B')$ and any initial clauses of $\alpha_{block}(G,k)$, where all $x$'s mentioned belong to the same block of $G$, $p+q = b$ and $A'$ and $B'$ correspond to the translated versions of $A$ and $B$ respectively.

The obvious way of doing this is to resolve the clause $(x_{u_1} \vee \ldots \vee x_{u_p} \vee A')$ with all 1-1 clauses $(\neg x_{u_i} \vee \neg x_{v_1})$ obtaining $(\neg x_{v_1} \vee A')$. Repeating this for all $x_{v_j}$'s gives us clauses $(\neg x_{v_j} \vee A')$. Note that this reuses $(x_{u_1} \vee \ldots \vee x_{u_p} \vee A')$ $q$ times and is therefore not tree-like. Resolving all $(\neg x_{v_j} \vee A')$ in turn with $(x_{v_1} \vee \ldots \vee x_{v_q} \vee B')$ gives us $(A' \vee B')$. This takes $pq + q < b^2$ steps. Hence the blow up in size for general resolution is at most a factor of $b^2$. Note that this procedure is symmetric in $A'$

and $B'$; we could also have picked the clause $(\neg y \vee B)$ to start with, in which case we would need $qp + p < b^2$ steps.

The tree-like case is somewhat trickier because we need to replicate clauses that are reused by the above procedure. We handle this using an idea similar to the one used in [10] for deriving the size-width relationship for tree-like resolution proofs. Let $newSize(s)$ denote the maximum over the sizes of all transformed tree-like proofs obtained from original tree-like proofs of size $s$ by applying the above procedure and creating enough duplicates to take care of reuse. We use induction to prove that $newSize(s) \leq (2s)^{\log_2 2b}$. For the base case, $newSize(1) = 1 \leq 2b$. For the inductive case, consider the subtree of the original proof that derives $(A \vee B)$ by resolving $(y \vee A)$ and $(\neg y \vee B)$ on the literal $y$ as above. Let this subtree be of size $s \geq 2$ and assume w.l.o.g. that the subtree deriving $(y \vee A)$ is of size $s_A \leq s/2$. By induction, the transformed version of this subtree deriving $(x_{u_1} \vee \ldots \vee x_{u_p} \vee A')$ is of size at most $newSize(s_A)$ and that of the other subtree deriving $(x_{v_1} \vee \ldots \vee x_{v_q} \vee B')$ is of size at most $newSize(s - s_A - 1)$. Choose $(x_{u_1} \vee \ldots x_{u_p} \vee A')$ as the clause to start the new derivation of $(A' \vee B')$ as described in the previous paragraph. The size of this refutation is at most $b \cdot newSize(s_A) + newSize(s - s_A - 1) + b^2$. Since we can do this for any original proof of size $s$, we must have $newSize(s) \leq b \cdot newSize(s_A) + newSize(s - s_A - 1) + b^2$ for $s \geq 2$ and $s_A \leq s/2$. It can be easily verified that $newSize(s) = 2bs\ b^{\log_2 s} = (2s)^{\log_2 2b}$ is a solution to this. We thus have the bound for the DLL case as well. □

**Lemma 2.** *For any graph $G$ on $n$ vertices and any integer $k$ dividing $n$,*

$$\begin{aligned} Res(\alpha_{block}(G,k)) &\leq Res(\alpha_{map}(G,k)), \quad \textit{and} \\ DLL(\alpha_{block}(G,k)) &\leq DLL(\alpha_{map}(G,k)). \end{aligned}$$

*Proof.* In the general encoding $\alpha_{map}(G,k)$, a vertex $v$ can potentially be chosen as the $i^{th}$ node of the $k$-independent set for *any* $i \in \{1,2,\ldots,k\}$. In the restricted encoding, however, vertex $v$ belonging to block $j$ can be thought of as either being selected as the $j^{th}$ node of the independent set or not being selected at all. Hence, if we start with a resolution (or DLL) refutation of $\alpha_{map}(G,k)$ and set $z_{v,i} = \text{FALSE}$ for $i \neq j$, we get a simplified refutation where the only variables are of the form $z_{v,j}$ where vertex $v$ belongs to block $j$. Renaming these $z_{v,j}$'s as $x_v$'s, we get a refutation in the variables of $\alpha_{block}(G,k)$ that is no larger in size than the original refutation of $\alpha_{map}(G,k)$.

All we now need to do is verify that this transformation either converts any given initial clause of $\alpha_{map}(G,k)$ to an initial clause of $\alpha_{block}(G,k)$ or satisfies it trivially. The transformed refutation will then be a refutation of $\alpha_{block}(G,k)$ itself. This reasoning is straightforward:

- Edge clauses $(\neg z_{u,i} \vee \neg z_{v,j})$ of $\alpha_{map}(G,k)$ that represented edge $(u,v) \in E$ with $u$ in block $i$ and $v$ in block $j$ transform into the corresponding edge clause $(\neg x_u \vee \neg x_v)$ of $\alpha_{block}(G,k)$. If vertex $u$ (or $v$) is not in block $i$ (or $j$, resp.), then the transformation sets $z_{u,i}$ (or $z_{v,j}$, resp.) to FALSE and the clause is trivially satisfied.

- Surjective clauses of $\alpha_{map}(G,k)$ clearly transform to the corresponding block clauses of $\alpha_{block}(G,k)$ – for the $i^{th}$ such clause, variables corresponding to vertices that do not belong to block $i$ are set to FALSE and simply vanish, and we are left with the $i^{th}$ block clause of $\alpha_{block}(G,k)$.

- It is easy to see that all function clauses and ordering clauses are trivially satisfied by the transformation.

- 1-1 clauses $(\neg z_{u,i} \vee \neg z_{v,i})$ of $\alpha_{map}(G,k)$ that involved vertices $u$ and $v$ both from block $i$ transform into the corresponding 1-1 clause $(\neg x_u \vee \neg x_v)$ of $\alpha_{block}(G,k)$. If vertex $u$ (or $v$) is not in block $i$, then the transformation sets $z_{u,i}$ (or $z_{v,i}$, resp.) to FALSE and the clause is trivially satisfied.

Thus, this transformed proof is a refutation of $\alpha_{block}(G,k)$ and the desired bounds follow. □

## 5. Simulating Chvátal's Proof System

We show that resolution on $\alpha_{block}(G,k)$ can simulate Chvátal's proofs [8] of non-existence of $k$-independent sets in $G$. This indirectly provides bounds on the running time of various algorithms for finding a largest independent set in a given graph. We first briefly describe his proof system. Let $(S,t)$ be the statement that the subgraph of $G$ induced by a vertex subset $S$ does not have an independent set of size $t$. $(\phi,1)$ is given as an axiom and the goal is to prove the statement $(V,k)$, where $V$ is the vertex set of $G$ and $k$ is given as input. Proofs in his system are based on the following property. Pick any vertex $v$. $v$ is either present in some largest independent set of $G$, or not. Therefore, $(S,t)$ is TRUE iff both $(S - N(v) - \{v\}, t-1)$ and $(S - \{v\}, t)$ are TRUE, where $N(v)$ is the set of neighbors of $v$. To prove $(S,t)$, one then recursively proves the two subproblems. We will denote by $Chv(G,k)$ the size of the smallest proof in Chvátal's system of the statement $(V(G),k)$.

A key idea for getting short proofs in Chvátal's system is to reuse proofs for subproblems. We say $(S,t)$ *dominates* $(S',t')$ iff $S \supseteq S'$ and $t \leq t'$. It is clear that once we have proved $(S,t)$, there is no need to provide a separate proof of $(S',t')$. Chvátal's system allows one to derive such an $(S',t')$ from $(S,t)$ in a single step using the *monotone rule*. Also, one can easily simplify the proof of $(S,t)$ to get a proof of $(S',t')$. This notion of reusing proofs for subproblems that are dominated by previously solved subproblems makes proofs in Chvátal's system look like directed acyclic graphs. As we will soon see, one can convert these proofs to corresponding resolution proofs by traversing the proof graph bottom-up and locally replacing each inference in Chvátal's system by a small number of resolution inferences.

We call a proof system for independent sets *monotone* if adding more edges to the original graph does not make it any harder to prove that there is no independent set of a certain size.

**Observation 2.** *Chvátal's proof system is monotone.*

*Proof.* Let $G'$ be a graph obtained by adding edges to a graph $G$. Let $V$ denote both the vertex set of $G$ and that of $G'$. In order to prove $(V,k)$ for the denser graph $G'$, we have to recursively prove $(V - N'(v) - \{v\}, k-1)$ and $(V - \{v\}, k)$ in $G'$, for some vertex $v$. Here $N'$ denotes the set of neighbors of $v$ in $G'$. Since $G'$ has all the edges of $G$ and some more, $V - N(v) - \{v\} \supseteq V - N'(v) - \{v\}$. Hence $(V - N(v) - \{v\}, k-1)$ in $G$ dominates $(V - N'(v) - \{v\}, k-1)$ in $G$. We can therefore rewrite the proof of $(V - N(v) - \{v\}, k-1)$ in $G$ to get a proof of $(V - N'(v) - \{v\}, k-1)$ in $G$. This is still not a proof of $(V - N'(v) - \{v\}, k-1)$ in $G'$ because the induced subgraph in $G'$ has a different set of edges than the induced graph in $G$. However, we can now inductively use the monotonicity of the smaller problem to convert the proof of $(V - N'(v) - \{v\}, k-1)$ in $G$ to one in $G'$. Using monotonicity inductively once again, we can also convert the proof of $(V - \{v\}, k)$ in $G$ to one in $G'$. Hence, by induction, we have a no larger proof of $(V,k)$ in $G'$ than we had in $G$. □

**Lemma 3.** *For any graph $G$ on $n$ vertices and any integer $k$ dividing $n$,*

$$Res(\alpha_{block}(G,k)) \leq 2n\, Chv(G,k).$$

*Proof.* Let $V$ denote the vertex set of $G$. Divide $V$ into $k$ blocks of equal size and let $G_{block}$ be the graph obtained by taking $G$ and adding all edges $(u,v)$ such that vertices $u$ and $v$ belong to the same block of $G$. In other words, $G_{block}$ is $G$ with modified to contain a clique on each block. By the monotonicity of Chvátal's system, we can convert the given proof of $(V,k)$ in $G$ to a proof of $(V,k)$ in $G_{block}$ that is no larger than that in $G$.

We will start with a fixed proof of $(V,k)$ in $G_{block}$ in Chvátal's system and use it to guide the construction of a resolution refutation for $\alpha_{block}(G,k)$. Observe that without loss of generality, for any statement $(S,t)$ in the proof, $t$ is at least the number of blocks of $G$ containing vertices in $S$. This is so because it is true for the final statement $(V,k)$ and if it is true for $(S,t)$, then it is also true for both $(S - \{v\}, t)$ and $(S - N(v) - \{v\}, t-1)$ from which $(S,t)$ is derived. We will call $(S,t)$ a *trivial* statement if $t$ is strictly bigger than the number of blocks of $G$ containing vertices in $S$. Notice that the initial statement $(\phi, 1)$ of the proof is trivial, whereas the final statement $(V,k)$ is not. Also, all statements derived by applying the monotone rule are non-trivial.

The resolution proof we will construct will have a clause associated with each non-trivial statement $(S,t)$ occurring in the proof. This clause will be a subclause of the clause $C_S \stackrel{\text{def}}{=} (\bigvee_{u \in N_S} x_u)$ where $N_S$ is the set of all vertices in $V - S$ that are in blocks of $G$ containing at least one vertex of $S$. We will construct our resolution proof inductively, going bottom-up through the non-trivial statements of the given proof. Note that the clause associated in this manner with $(V,k)$ will be the empty clause and hence we have a refutation.

Suppose $(S,t)$ is non-trivial and is derived in the original proof by applying the branching rule to vertex $v \in S$. Then we write our target clause $C_S$ as $(C_S^b \vee C_S^r)$ where $C_S^L$ is the disjunction of all variables corresponding to vertices of $N_S$ that are in the same block as $v$ and $C_S^R$ is the disjunction of all variables corresponding to vertices of $N_S$ that are in the remaining blocks. $v \in S$, Before deriving the subclause of $C_S$, we will derive two clauses $Cl_1$ and $Cl_2$ as follows depending on the properties of the inference that produced $(S,t)$:

**Case 1**: Both $(S - \{v\}, t)$ and $(S - N(v) - \{v\}, t-1)$ are trivial. It is easy to see that since $(S,t)$ is non-trivial, if $(S - \{v\}, t)$ is trivial then $v$ is the only vertex of $S$ in its block. We let $Cl_1$ be the initial block clause for the block containing $v$ which is precisely $(x_v \vee C_S^b)$. The fact that $(S - N(v) - \{v\}, t-1)$ is also trivial implies that the neighbors of $v$ include not only every vertex of $S$ appearing in the block containing $v$ but also all vertices in $S \cap B$ where $B$ is some other block that does not contain $v$. Resolving the block clause for block $B$ with all edge clauses $(\neg x_v \vee \neg x_u)$ for $u \in S \cap B$ gives us a subclause $Cl_2$ of $(\neg x_v \vee C_S^r)$.

**Case 2**: $(S - \{v\}, t)$ is trivial but $(S - N(v) - \{v\}, t-1)$ is non-trivial. We set $Cl_1$ exactly as in case 1. Given that $(S - N(v) - \{v\}, t-1)$ is non-trivial, we have by inductive assumption a subclause of $C_{S - N(v) - \{v\}}$. Since the given proof applies to $G_{block}$, $N(v) \cup v$ con-

tains every vertex in the block containing $v$ as well as all neighbors of $v$ in $G$ that are not in $v$'s block. Therefore the subclause of $C_{S-N(v)-\{v\}}$ we have by induction, is a subclause of $(C^r_S \vee x_{u_1} \vee \ldots \vee x_{u_p})$, where each $u_i$ is a neighbor of $v$ in $S$ in blocks other than $v$'s block. We derive a new clause $Cl_2$ by resolving this clause with all edge clauses $(\neg x_v \vee \neg x_{u_i})$. Observe that $Cl_2$ is a subclause of $(\neg x_v \vee C^r_S)$.

**Case 3**: $(S - \{v\}, t)$ is non-trivial but $(S - N(v) - \{v\}, t-1)$ is trivial. We set $Cl_2$ as in case 1. Since $(S - \{v\}, t)$ is non-trivial, we have by induction a subclause $Cl_2$ of $C_{S-\{v\}}$, i.e. a subclause of $(x_v \vee C_S)$.

**Case 4**: Both $(S-\{v\}, t)$ and $(S-N(v)-\{v\}, t-1)$ are non-trivial. In this case, we derive $Cl_1$ as in case 3 and $Cl_2$ as in case 2.

It is easy to verify that $Cl_1$ is a subclause of $(x_v \vee C_S)$ and $Cl_2$ is a subclause of $(\neg x_v \vee C^r_S)$. If either $Cl_1$ or $Cl_2$ does not mention $x_v$ at all, then we already have the desired subclause of $C_S$. Otherwise we resolve $Cl_1$ with $Cl_2$ to get a subclause of $C_S$. This completes the construction. Given any statement in the original proof, it takes at most $2n$ steps to derive the subclause associated with it in the resolution proof, given that we have already derived the corresponding subclauses for the two branches of that statement. This gives the bound on the size of the constructed resolution refutation. □

It follows that our bounds apply to Chvátal's system and hence also to many algorithms for finding a largest independent set in a given graph that are captured by his proof system [16, 18, 19, 20].

## 6. Key Concepts for Lower Bounds

This section defines key concepts that will be used in the lower bound argument given in the next section. We will fix a graph $G$ and a partition of its $n$ vertices into $k$ subsets of size $b$ each. For any edge $(u, v)$ in $G$, we will call it an *inter-block edge* if $u$ and $v$ belong to different blocks of $G$ and an *intra-block edge* otherwise.

### 6.1. Critical Truth Assignment

We call a truth assignment to variables of $\alpha_{block}(G, k)$ *critical* if it sets exactly one variable in each block to TRUE. Critical truth assignments clearly satisfy all block, 1-1 and intra-block edge clauses but may leave some inter-block edge clauses unsatisfied. It is easy to see that if $\alpha_{block}(G, k)$ does have a satisfying assignment, it also has a satisfying *critical* assignment – we can simply reset all but one TRUE variable in each block to FALSE.

Fix a critical truth assignment $\gamma$. Then for each $i \in \{0, 1, \ldots, k-1\}$ and each $j \in \{1, 2, \ldots, b\}$, $\gamma(\neg x_{bi+j})$ = TRUE iff $\gamma(x_{bi+1} \vee \ldots \vee x_{bi+j-1} \vee x_{bi+j+1} \vee \ldots \vee x_{bi+b})$ = TRUE. We can use this equivalence to convert any clause $C$ in the variables appearing in $\alpha_{block}(G, k)$ into a clause $C^+$ in which every variable occurs positively by replacing each negated variable by the disjunction of the other variables in the block. Observe that $C$ and $C^+$ are equivalent under all critical truth assignments.

### 6.2. Block Graph

It will be useful to look at the block multi-graph of $G$, denoted $B(G)$, obtained by identifying all vertices that belong to the same block in $G$ and removing any self-loops that are thus generated. $B(G)$ contains exactly $k$ nodes and possibly multiple edges between pairs of nodes. The degree of a node in $B(G)$ is the number of inter-block edges touching the corresponding block of $G$. Given the natural correspondence between $G$ and $B(G)$, we will write *nodes of* $B(G)$ and *blocks of* $G$ interchangeably.

### 6.3. Block Induced Subgraphs and Boundary

Let $H$ be a subgraph of $G$ and $S$ a subset of blocks of $G$. We say that $H$ is *block induced by* $S$ if it is the subgraph of $G$ induced by all vertices present in the blocks $S$. Clearly, if $H$ is block induced by $S$, then $B(H)$ is induced by $S$ in $B(G)$. $H$ will be called a *block induced subgraph* of $G$ if there exists a subset $S$ of blocks such that $H$ is block induced by $S$. Further, if $H$ is a block induced subgraph, then we can find a *unique* minimal block set $S$ such that $H$ is block induced by $S$. This $S$ simply contains all blocks which have non-zero degree in $B(H)$. With each block induced subgraph, we will associate such a minimal $S$ and say that the subgraph is induced by $|S|$ blocks.

We define the *boundary* of a block induced subgraph $H$, denoted $\beta(H)$, to be the set of nodes of $B(H)$ (blocks of $H$) which have degree (number of inter-block edges, respectively) between 1 and $b-1$. The following property will be useful in obtaining a lower bound on the width of clauses implied by subgraphs.

**Observation 3.** *Given any boundary block, we can find two vertices $u$ and $v$ in it such that $u$ has no inter-block edges and $v$ has at least one.*

### 6.4. Minimal Implication and Block Width

For a block induced subgraph $H$ of $G$, let $E(H)$ denote the conjunction of the edge clauses of $\alpha_{block}(G, k)$

corresponding to the edges of $H$. We say that $H$ *critically implies* a clause $C$ iff $E(H) \rightarrow C$ is true for all critical truth assignments to the variables of $\alpha_{block}(G,k)$. Let subgraph $H$ be induced by the block set $S$. We say that $H$ *minimally implies* $C$ if $H$ critically implies $C$ and no subgraph of $G$ which is induced by a proper subset of $S$ critically implies $C$.

The *block width* of a clause $C$ with respect to $G$, denoted $w^G_{block}(C)$, is the number of different blocks of $G$ the variables appearing in $C$ come from.

**Observation 4.** *For any clause $C$ and any block induced subgraph $H$ of $G$,*

1. *$H$ minimally implies $C$ iff $H$ minimally implies $C^+$.*
2. $w(C) \geq w^G_{block}(C) = w^G_{block}(C^+)$

### 6.5. Clause Complexity

Let $C$ be a clause over variables of $\alpha_{block}(G,k)$. The *complexity of $C$*, denoted $\mu_G(C)$, is the minimum over the sizes of subsets $S$ of blocks such that subgraph $H$ induced by $S$ critically implies $C$. In other words, it is the minimum number of blocks we need to look at so as to make the block induced subgraph critically imply $C$. In the following, we state some simple properties of the complexity measure $\mu_G$.

**Observation 5.** *Let $G$ be a graph and $\Lambda$ denote the empty clause. Then*

1. *For an* initial clause $C$, *i.e. for $C \in \alpha_{block}(G,k)$, $\mu_G(C) \leq 2$.*
2. *$\mu_G(\Lambda)$ is the number of blocks in the smallest block induced subgraph of $G$ that has no block-respecting independent set of size $k$.*
3. Subadditive property*: If clause $C$ is a resolvant of clauses $C_1$ and $C_2$, then $\mu_G(C) \leq \mu_G(C_1) + \mu_G(C_2)$.*

*Proof.* Each initial clause is either an edge clause, a block clause or a 1-1 clause. Any critical truth assignment, by definition, satisfies all block, 1-1 and intra-block edge clauses. Further, an edge clause corresponding to an inter-block edge $(u,v)$ is implied by the subgraph induced by the two blocks to which $u$ and $v$ belong. Hence, complexity of an initial clause is at most 2, proving part 1.

Part 2 is trivially true by definition of $\mu_G$. Part 3 follows from the simple observation that if $G_1$ critically implies $C_1$, $G_2$ critically implies $C_2$ and both $G_1$ and $G_2$ are block induced subgraphs, then $G_{1\cup 2}$, defined as the block graph induced by the union of the blocks $G_1$ and $G_2$ are induced by, critically implies both $C_1$ and $C_2$, and hence critically implies $C$. □

## 7. Proof of Lower Bounds

We use combinatorial properties of block graphs and independent sets to obtain a lower bound on the size of resolution refutations for a given graph in terms of its expansion properties. Next, we argue that random graphs almost surely have good expansion properties. Section 8 combines these two to obtain an almost certain lower bound for random graphs.

### 7.1. Relating Proof Size to Graph Expansion

**Lemma 4.** *Let $C$ be a clause in the variables of $\alpha_{block}(G,k)$. If $H$ is a block induced subgraph of $G$ that minimally implies $C$, then $w^G_{block}(C) \geq |\beta(H)|$.*

*Proof.* By Observation 4, it suffices to assume that every literal of $C$ is positive.

We will use the toggling property of block-respecting independent sets (Figure 2) to show that each boundary block of $H$ contributes to $C$ every positive literal that corresponds to a vertex in it with no inter-block edges. In particular, at least one literal from every boundary block appears in $C$.

Fix a boundary block $B$. From Observation 3, $B$ must contain two vertices $u$ and $v$ such that $u$ has no inter-block edge and $v$ has an inter-block edge $(v,w)$. If we let $H_B$ be the block induced subgraph that has all edges of $H$ except those that have an endpoint in block $B$, then $H_B$ is a strict subgraph of $H$ and by minimality, cannot critically imply $C$. Therefore, there must exist a critical truth assignment $\gamma$ such that $\gamma(E(H_B))$ is TRUE but $\gamma(C)$ is FALSE. We can think of $\gamma$ as picking exactly one vertex from each block. Note that $\gamma$ cannot pick vertex $u$ from block $B$ because then the lack of edges with endpoints in $B$ won't matter. $\gamma$ will not violate any edge clauses of $H$ itself. In other words, $\gamma(E(H))$ will be TRUE which would imply that $\gamma(C)$ is TRUE – a contradiction. Therefore for suitable choices of $v$ and $w$, $\gamma$ picks $v$ from block $B$ and its neighbor $w$ from some other block.

Now create another critical truth assignment $\gamma'$ which differs from $\gamma$ only in that it picks vertex $u$ from block $B$ whereas $\gamma$ didn't. Since $\gamma'$ is identical to $\gamma$ in blocks other than $B$ and $\gamma(E(H_B))$ is TRUE, $\gamma'(E(H_B))$ must also be TRUE. Moreover, since $\gamma'$ picked vertex $u$ from block $B$ and $u$ does not have any inter-block edges, $\gamma'$ cannot violate any edge with an endpoint in $B$. Therefore, even $\gamma'(E(H))$ is TRUE, implying that $\gamma'(C)$ is TRUE.

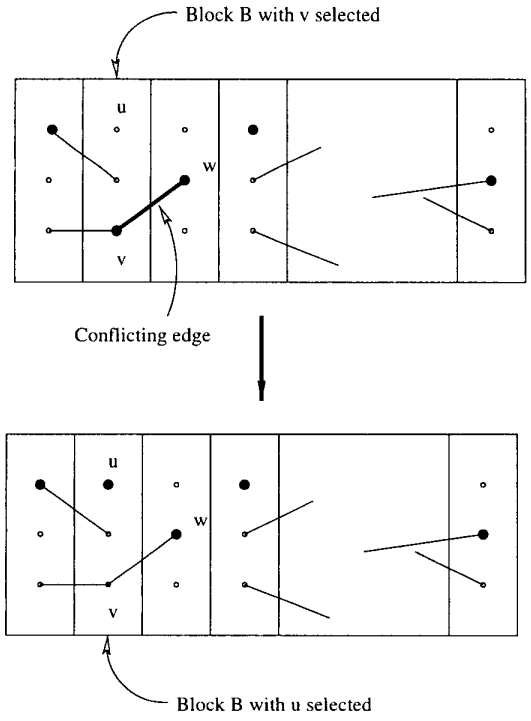


**Figure 2. Toggling property of block-respecting independent sets (selected vertices are shown in bold)**

We now have two critical truth assignments $\gamma$ and $\gamma'$ that differ only in that the latter sets $x_u$ to TRUE whereas the former doesn't, and that the latter satisfies $C$ while the former doesn't. This is what we earlier referred to as the toggling property. Since $C$ contains only positive literals, this can happen only if $x_u \in C$. □

If we start with $G$ and keep throwing edges away, the resulting subgraph will surely contain a block-respecting independent set after enough edges are gone. Let $s+1$ be the minimum number of blocks such that some subgraph of $G$ induced by $s+1$ blocks does not have a $k$-independent set. Now define the *sub-critical expansion*, $e(G)$, of $G$ to be the maximum over all $t, 2 \leq t \leq s$, of the minimum boundary size of any subgraph $H$ of $G$ induced by $t'$ blocks, where $t/2 < t' \leq t$.

**Lemma 5.** *Any resolution refutation of $\alpha_{block}(G,k)$ must contain a clause of width at least $e(G)$.*

*Proof.* Let $t$ be chosen as in the definition of $e(G)$ and let $\pi$ be a resolution refutation of $\alpha_{block}(G,k)$. By Observation 5.2, $\mu_G(\Lambda) = s+1$. Further, Observation 5.1 says that any initial clause has complexity at most 2. Therefore for $2 < t \leq s$ there exists a clause $C$ in $\pi$ such that $\mu_G(C) > t \geq 2$ and no ancestor of $C$ has complexity greater than $t$.

Since $\mu_G(C) > 2$, $C$ cannot be an initial clause. It must then be a resolvant of two parent clauses $C_1$ and $C_2$. By Observation 5.3 and the fact that no ancestor of $C$ has complexity greater than $t$, one of these clauses, say $C_1$, must have $\mu_G(C_1)$ between $(t+1)/2$ and $t$. If $H$ is a block induced subgraph that witnesses the value of $\mu_G(C_1)$, then by Lemma 4, $w^G_{width}(C_1) \geq |\beta(H)|$. From Observation 4, this implies $w(C_1) \geq |\beta(H)|$. By definition of $e(G)$, $|\beta(H)| \geq e(G)$. Thus $w(C_1) \geq e(G)$ as required. □

**Lemma 6.** *If $G$ is a graph with vertices divided into $k$ blocks of size $b = n/k$ each and $\alpha_{block}(G,k)$ is the CNF formula saying that $G$ has a block-respecting independent set of size $k$, then for the constant $c > 0$ in Proposition 2,*

$$\begin{aligned} Res(\alpha_{block}(G,k)) &\geq 2^{c(e(G)-b)^2/n}, \text{ and} \\ DLL(\alpha_{block}(G,k)) &\geq 2^{e(G)-b}. \end{aligned}$$

*Proof.* This follows immediately from Lemma 5 and Propositions 2 and 1 by observing that the initial width of $\alpha_{block}(G,k)$ is $b$. □

## 7.2. Lower Bounding Sub-critical Expansion

Throughout this section, the probabilities we mention will be with respect to the random choice of the graph $G$ from distribution $\mathcal{G}_{n,m}$ for some parameters $n$ and $m$. Let $B(G)$ be a block graph corresponding to $G$ with block size $b$. We call $B(G)$ $(r,q)$-*dense* if some subgraph of $G$ induced by $r$ blocks (i.e. some subgraph of $B(G)$ with $r$ nodes) contains at least $q$ edges. The following Lemma shows that for almost all random graphs $G$, $B(G)$ is locally sparse.

**Lemma 7.** *Let $G \sim \mathcal{G}_{n,m}$ and $B(G)$ be a corresponding block graph with block size $b$. Let $\Delta \stackrel{\text{def}}{=} m/n$. Then for $r, q \geq 1$,*

$$\Pr[B(G) \text{ is } (r,q)\text{−dense}] < \left(\frac{ne}{br}\right)^r \left(\frac{eb^2r^2\Delta}{nq}\right)^q$$

*Proof.* Let $R$ be a subset of $r$ nodes of $B(G)$. $R$ then corresponds to $br$ vertices of $G$. A given edge of $G$ transforms into an edge of $B(G)$ with its endpoints in two different blocks in $R$ with probability $p = (br)(br-r)/(n(n-1)) < (br/n)^2$. For $G \sim \mathcal{G}_{n,m}$, the number of edges contained in $R$ has the binomial distribution $B(m,p)$. Hence, the probability that at least $q$ edges of $B(G)$ are contained in $R$ is

$$\begin{aligned}\Pr[B(m,p) \geq q] &\leq \binom{m}{q} p^q \\ &< \left(\frac{em}{q}\right)^q \left(\frac{b^2r^2}{n^2}\right)^q \\ &= \left(\frac{eb^2r^2\Delta}{nq}\right)^q\end{aligned}$$

Summing this over all the $\binom{n/b}{r} \leq (ne/br)^r$ $r$-subsets, $R$, of nodes of $B(G)$, we obtain

$$\Pr[B(G) \text{ is } (r,q)\text{−dense}] < \left(\frac{ne}{br}\right)^r \left(\frac{eb^2r^2\Delta}{nq}\right)^q.$$

□

We use this local sparseness property of almost all random graphs to prove that the smallest block induced subgraph one needs to consider for proving that $G$ does not have a block-respecting independent set of size $k$ is almost surely large. This shows that any resolution proof that $G$ does not have a block-respecting independent set of size $k$ will have to consider at least a constant fraction of blocks when $G$ has does not have too many edges. More precisely,

**Lemma 8.** *Let $m, n, b$ be integers with $\Delta \stackrel{\text{def}}{=} m/n$ and $b \geq 3$ and $k = n/b$. There exists a constant $C$ such that if $s < Cn/(b\Delta^{b/(b-2)})$, then the probability that $G \sim \mathcal{G}_{n,m}$ contains a subgraph induced by at most $s$ blocks that has no block-respecting independent set of size $k$ is $o(1)$ in $s$.*

*Proof.* The probability that $G$ contains a subgraph induced by at most $s$ blocks that has no block-respecting independent set of size $k$ is the same as the probability that there is some *minimal* subgraph $H$ of $G$ which is induced by $r \leq s$ blocks and has no block-respecting independent set of size $k$. Since clauses corresponding to $H$ are unsatisfiable, $H$ critically implies the empty clause $\Lambda$ which is of width 0. It follows from Lemma 4 that $H$ has no boundary blocks. Further, since $H$ is minimal, it does not have any blocks with no inter-block edges. It follows that each of the $r$ blocks that induce $H$ must have at least $b$ inter-block edges. Hence, the subgraph of $B(G)$ with the $r$ nodes corresponding to the $r$ blocks that induce $H$ must have at least $br/2$ edges.

Thus, the probability that $G$ contains such a block induced subgraph $H$ is at most

$$\sum_{r=1}^{s} \Pr[B(G) \text{ is } (r, br/2)\text{−dense}].$$

By Lemma 7, we have $\Pr[B(G) \text{ is } (r, br/2)\text{−dense}] < D(r)$ where

$$\begin{aligned}D(r) &= \left(\frac{ne}{br}\right)^r \left(\frac{2ebr\Delta}{n}\right)^{br/2} \\ &= \left(\frac{ne}{b}\left(\frac{2eb\Delta}{n}\right)^{b/2} r^{(b-2)/2}\right)^r \\ &= \left(Q(b,n,\Delta)\, r^{(b-2)/2}\right)^r\end{aligned}$$

for $Q(b,n,\Delta) = (ne/b)(2eb\Delta/n)^{b/2}$. Now

$$\begin{aligned}\frac{D(r+1)}{D(r)} &= \frac{(Q(b,n,\Delta)\,(r+1)^{(b-2)/2})^{r+1}}{(Q(b,n,\Delta)\, r^{(b-2)/2})^r} \\ &= Q(b,n,\Delta)\,(r+1)^{(b-2)/2} \left(\frac{r+1}{r}\right)^{r(b-2)/2} \\ &\leq Q(b,n,\Delta)\,(r+1)^{(b-2)/2}\, e^{(b-2)/2} \\ &\leq (ne/b)(2eb\Delta/n)^{b/2}\,(e(r+1))^{(b-2)/2} \\ &= (1/b)(2e^2b\Delta)^{b/2}\,((r+1)/n)^{(b-2)/2}\end{aligned}$$

This quantity is at most $1/2$ when

$$\begin{aligned}\frac{r+1}{n} &\leq \left(\frac{b}{2}\right)^{2/(b-2)} \frac{1}{(2e^2b\Delta)^{b/(b-2)}} \\ &= \left(\frac{b/2}{2e^2b}\right)^{2/(b-2)} \frac{1}{2e^2b\,\Delta^{b/(b-2)}} \\ &= \left(\frac{1}{4e^2}\right)^{2/(b-2)} \frac{1}{2e^2b\,\Delta^{b/(b-2)}}\end{aligned}$$

For $b \geq 3$, $(1/4e^2)^{2/(b-2)} \geq 1/(16e^4)$. Hence it suffices to have $(r+1)/n \leq 1/(16e^4\, 2e^2b\Delta^{b/(b-2)}) = 1/(32e^5\, b\Delta^{b/(b-2)})$. Therefore, $D(r+1)/D(r) \leq 1/2$ for $1 \leq r < Cn/(b\Delta^{b/(b-2)})$, where $C \stackrel{\text{def}}{=} 1/(32e^5)$ is a constant. Let $s+1 = Cn/(b\Delta^{b/(b-2)})$. Then the probability that $G$ contains such a block induced subgraph is bounded above by a geometric series in $r$ with common ratio $1/2$. It is therefore at most twice the largest term of the series which is less than $D(1)$. Rewriting $D(1)$ in terms of $s$ using the fact that $\Delta = (\frac{Cn}{b(s+1)})^{(b-2)/b}$ gives

$$\begin{aligned}D(1) &= \left(\frac{ne}{b}\right)\left(\frac{2eb\Delta}{n}\right)^{b/2} \\ &= \frac{1}{(s+1)^{(b-2)/2}}\, 32e^6 \left(\frac{1}{16e^4}\right)^{b/2}\end{aligned}$$

which for $b \geq 3$ is $o(1)$ in $s$ as required. □

We again use the local sparseness property to prove that any subgraph induced by not too many as well as not too few blocks has large boundary for almost all

random graphs $G$. The intuition is that for sparse subgraphs, most blocks have degree $< b$ whereas for dense subgraphs, most blocks have degree $> 0$. Hence, if we look at middle size subgraphs, it is very likely that some large fraction of blocks will have degree between 1 and $b-1$, and will therefore belong to the boundary.

**Lemma 9.** *Let $m, n, b$ be integers with $\Delta \stackrel{\text{def}}{=} m/n$ and $b \geq 5$. For $0 < \epsilon \leq 1/2$, let $b' \stackrel{\text{def}}{=} b - (b-1)\epsilon$. There exists a constant $c$ such that if $t \leq cn/(b\Delta^{\frac{b'}{b'-2}})$, then the probability that $G \sim \mathcal{G}_{n,m}$ has a subgraph $H$ induced by $r$ blocks, $t/2 < r \leq t$, with $\beta(H) \leq \epsilon r$ is $o(1)$ in $t$.*

*Proof.* Fix $b, \epsilon$ and $r$ satisfying all conditions of the Lemma. Since $H$ is induced by $r$ blocks, by definition, all $r$ blocks inducing $H$ must have non-zero degree in $B(H)$. Moreover, if $H$ has at most $\epsilon r$ boundary blocks, the other $(1-\epsilon)r$ blocks of non-zero degree inducing it must have degree at least $b$. Hence, the $r$ nodes of $B(G)$ that induce $H$ form a subgraph with at least $(\epsilon r + (1-\epsilon)rb)/2 = b'r/2$ edges. Therefore, $H$ has at most $\epsilon r$ boundary blocks only if $B(G)$ is $(r, b'r/2)$-dense. Thus by Lemma 7, the probability that such an $H$ exists is at most

$$\begin{aligned}
&\Pr[B(G) \text{ is } (r, b'r)\text{−dense}] \\
&< \left(\frac{ne}{br}\right)^r \left(\frac{2eb^2r^2\Delta}{nb'r}\right)^{b'r/2} \\
&= \left(\frac{e}{b}\left(\frac{2eb^2\Delta}{b'}\right)^{b'/2}\left(\frac{r}{n}\right)^{(b'-2)/2}\right)^r
\end{aligned}$$

Since $r > t/2$, it will suffice to obtain an upper bound on this probability that is exponentially small in $r$. We first note that since $\epsilon \leq 1/2$, $b'$ must be at least $(b+1)/2$. Moreover, since $b \geq 5$, $b'$ must be at least 3. Rearranging terms now gives that $\Pr[B(G) \text{ is } (r, b'r)\text{−dense}] \leq 2^{-r}$ when

$$\begin{aligned}
\frac{r}{n} &\leq \left(\frac{b}{2e}\right)^{2/(b'-2)}\left(\frac{b'}{2eb^2\Delta}\right)^{b'/(b'-2)} \\
&= \left(\frac{b'}{4e^2b}\right)^{2/(b'-2)} \frac{b'}{2eb^2\ \Delta^{b'/(b'-2)}} \\
&\leq \left(\frac{1}{8e^2}\right)^{2/(b'-2)} \frac{1}{4eb\ \Delta^{b'/(b'-2)}}
\end{aligned}$$

since $b' \geq (b+1)/2$.

For $b' \geq 3$, we see that $1/(8e^2)^{2/(b'-2)}$ is at least $1/(64e^4)$. Hence, it suffices to have $r/n \leq 1/(64e^4\ 4eb\Delta^{b'/(b'-2)}) = 1/(256e^5 b\ \Delta^{b'/(b'-2)})$. Therefore, the probability that $B(G)$ is $(r, b'r)$-dense is at most $2^{-r}$ for $r \leq cn/(b\Delta^{b'/(b'-2)})$, where $c \stackrel{\text{def}}{=} 1/(256e^5)$ is a constant. It follows that the probability that there exists such an $H$ with $t/2 < r \leq t$ is at most $\sum_{r=\lceil (t+1)/2 \rceil}^{t} 2^{-r}$. This sum is $o(1)$ in $t$ as required. □

Combining Lemmas 8 and 9 we obtain the following lower bound on sub-critical expansion:

**Lemma 10.** *Let $m, n, b$ be integers with $\Delta \stackrel{\text{def}}{=} m/n$ and $b \geq 5$. For $0 < \epsilon \leq 1/2$, let $b' \stackrel{\text{def}}{=} b - (b-1)\epsilon$ and $W \stackrel{\text{def}}{=} n/(b\Delta^{b'/(b'-2)})$. Then there exists a constant $c_\epsilon$ depending only on $\epsilon$ such that the probability that $G \sim \mathcal{G}_{n,m}$ has $e(G) < c_\epsilon W$ is $o(1)$ in $W$.*

*Proof.* Let $C$ be the constant from Lemma 8, $c$ be the one from Lemma 9 and $c^*$ be the minimum of these two. By Lemma 8, if $G \sim \mathcal{G}_{n,m}$ and $s+1 = Cn/(b\Delta^{b/(b-2)})$, then the probability that a subgraph $H$ of $G$ induced by at most $s$ blocks does not have a block-respecting independent set of size $k$ is $o(1)$ in $s$, which is $o(1)$ in $W$. Now let $t = c^*W$, which is at most $s$ because of our choice of $c^*$ and the fact that $b' < b$. By Lemma 9, the probability that $G \sim \mathcal{G}_{n,m}$ has a subgraph $H$ induced by $r$ blocks, $t/2 < r \leq t$, that has at most $\epsilon r$ boundary blocks is $o(1)$ in $t$, and thus $o(1)$ in $W$.

It follows that with probability $1 - o(1)$ in $W$, every subgraph of $G$ induced by $r$ blocks with $t/2 < r \leq t \leq s$ has at least $\epsilon r \geq \epsilon t/2 = \epsilon c^* W/2$ boundary blocks. Letting $c_\epsilon = \epsilon c^*/2$ yields the desired bound on $e(G)$. □

## 8. Main Results

### 8.1. Lower Bounds

**Theorem 1.** *Let $m, n, k$ be integers with $\Delta \stackrel{\text{def}}{=} m/n$, $k \leq n/5$ and $n$ a multiple of $k$. If $G \sim \mathcal{G}_{n,m}$ and $\Delta = o(n^{1/5})$, then for each $0 < \epsilon \leq 1/2$, there exists a global constants $C_\epsilon$, $C'_\epsilon$ such that with probability $1 - o(1)$ in $n$,*

$$\begin{aligned}
Res(\alpha_{block}(G,k)) &\geq 2^{C_\epsilon k^2/(n\Delta^{2+2\delta})}, \quad \text{and} \\
DLL(\alpha_{block}(G,k)) &\geq 2^{C'_\epsilon k/\Delta^{1+\delta}}
\end{aligned}$$

*where $\delta = \frac{2k}{n-(n-k)\epsilon+2k}$.*

*Proof.* Let $b = n/k \geq 5$, $b' = b - (b-1)\epsilon$ and $W = n/(b\Delta^{b'/(b'-2)})$. From Lemma 10, there exists a constant $c_\epsilon$ such that with probability $1 - o(1)$ in $W$, $e(G) \geq c_\epsilon W$. Then by Lemma 6 $Res(\alpha_{block}(G,k)) \geq 2^{c(c_\epsilon W - b)^2/n}$ with probability $1 - o(1)$ in $W$ where $c = 1/(9 \ln 2)$.

We now show that our conditions imply that $W$ is large enough that this yields our claimed exponential lower bounds with probability $1-o(1)$ in $n$. Since $b \geq 5$ and $\epsilon \leq 1/2, b'/(b'-2) \leq 3$. Hence $W/b \geq n/(b^2\Delta^3)$. Note that for $b \geq 4\Delta$, $k$ is at most $n/(4\Delta)$ and by Observation 1, an independent set of size $k$ surely exists. In this case, there are simply no resolution refutations of $\alpha_{block}(G,k)$. Therefore, we can safely assume $b < 4\Delta$, in which case $W/b > n/(16\Delta^5)$. Since by assumption $\Delta = o(n^{1/5})$, there is some constant $C_\epsilon > 0$ such that $c(c_\epsilon W - b)^2 \geq C_\epsilon W^2$. Thus

$$\begin{aligned}
&\log_2(Res(\alpha_{block}(G,k))) \\
&\quad \geq C_\epsilon W^2/n \\
&\quad = C_\epsilon n^2/(nb^2\Delta^{\frac{2b'}{b'-2}}) \\
&\quad = C_\epsilon k^2/(n\Delta^{2+\frac{4}{b-(b-1)\epsilon+2}}) \\
&\quad = C_\epsilon k^2/(n\Delta^{2+\frac{4k}{n-(n-k)\epsilon+2k}}) \\
&\quad = C_\epsilon k^2/(n\Delta^{2+2\delta})
\end{aligned}$$

A similar calculation gives the $DLL$ bound. □

**Corollary 1.** *For $G, m, n, k, \Delta, \epsilon, \delta$ as in Theorem 1 there are constants $c_\epsilon$, $c'_\epsilon$, $c''_\epsilon$, $c'''_\epsilon$, $c''''_\epsilon > 0$, such that with probability $1-o(1)$ in $n$,*

$$\begin{aligned}
Res(\alpha_{map}(G,k)) &\geq 2^{c_\epsilon k^2/(n\Delta^{2+2\delta})}, \\
DLL(\alpha_{map}(G,k)) &\geq 2^{c'_\epsilon k/\Delta^{1+\delta}}, \\
Res(\alpha_{count}(G,k)) &\geq 2^{c''_\epsilon k^2/(n\Delta^{2+2\delta})}, \\
DLL(\alpha_{count}(G,k)) &\geq 2^{c'''_\epsilon k/(\Delta^{1+\delta}\log\Delta)}, \quad \textit{and} \\
Chv(G,k) &\geq 2^{c''''_\epsilon k^2/(n\Delta^{2+2\delta})}.
\end{aligned}$$

*Proof.* The bounds for $\alpha_{map}(G,k)$ follow directly from Theorem 1 and Lemma 2. For the bounds on $\alpha_{count}(G,k)$, observe that we can assume $b < 4\Delta$ because otherwise $k = n/b \leq n/(4\Delta)$ and from Observation 1, $G$ definitely contains a $k$-independent set. The numbers $b^2$ and $\log_2 2b$ from Lemma 1 are therefore bounded above by $16\Delta^2$ and $\log_2(8\Delta)$ respectively. Combining this with Theorem 1, we get the desired bounds. □

**Corollary 2.** *For almost all graphs $G = (V,E)$ with a linear number of edges and for any $k \leq |V|/5$, $Res(\alpha_{block}(G,k)) = 2^{\Omega(n)}$.*

*Proof.* Follows from Theorem 1 by setting $\Delta$ = constant and observing that the largest independent set in a graph with a linear number of edges is also linear in size almost surely. □

**Corollary 3.** *Let $m, n, k$ be integers with $\Delta \stackrel{\text{def}}{=} m/n$, $k \leq n/5$ and $n$ a multiple of $k$. If $G \sim \mathcal{G}_{n,m}$ and $\Delta = o(n^{1/5})$, then there are global constants $c$ and $c'$ such that with probability $1-o(1)$ in $n$,*

$$\begin{aligned}
Res(\alpha_{block}(G,k)) &\geq 2^{cn^{1/25}}, \quad \textit{and} \\
DLL(\alpha_{block}(G,k)) &\geq 2^{c'n^{11/25}}.
\end{aligned}$$

*Proof.* A trivial calculation shows that for $0 < \epsilon \leq 1/2$ and $k \leq n/5$, $\frac{4k}{n-(n-k)\epsilon+2k} \leq 4/5$, and this value is achieved for $\epsilon = 1/2$. If $k \leq n/(4\Delta)$, then from Observation 1, $G$ surely has a $k$-independent set and no resolution refutations of $\alpha_{block}(G,k)$ exist. Therefore, we can safely assume $k > n/(4\Delta)$. It then follows from Theorem 1 that $\log_2(Res(\alpha_{block}(G,k))) > C_{1/2}(n^2/(16\Delta^2))/(n\Delta^{2+4/5}) = C_{1/2}n/(16\Delta^{4+4/5})$. Since $\Delta = o(n^{1/5})$, this quantity is at least $cn^{1/25}$ for $c = \frac{C_{1/2}}{16}$. This gives us the resolution bound. A similar calculation gives the $DLL$ bound. □

## 8.2. Upper Bounds

**Theorem 2.** *Let $G$ be a graph with $n$ vertices and $m$ edges, $\Delta \stackrel{\text{def}}{=} m/n$ and $k$ be any integer for which $G$ does not have a $k$-independent set. Then*

$$DLL(\alpha_{map}(G,k)) = 2^{O(k\log(n/k))}$$

*This bound also holds when $\alpha_{map}(G,k)$ does not include 1-1 clauses.*

*Proof.* A straightforward way to disprove the existence of a $k$-independent set is to go through all $\binom{n}{k}$ subsets of vertices of size $k$ and use as evidence an edge from each subset. We will use this to derive a refutation of $\alpha_{map}(G,k)$.

To begin with, we apply trasitivity to derive all ordering clauses of the form $(\neg z_{u,j} \vee \neg z_{v,i})$ for $u < v$ and $i < j$. If $j = i+1$, this is simply one of the original ordering clauses. For $j = i+2$, we derive the new clause $(\neg z_{u,i+2} \vee \neg z_{v,i})$ as follows. Consider any $w \in \{1,2,\ldots,n\}$. If $u < w$, we have the ordering clause $(\neg z_{w,i+1} \vee \neg z_{u,i+2})$, and if $u \geq w$, then $v > w$ and we have the ordering clause $(\neg z_{v,i} \vee \neg z_{w,i+1})$. Resolving these $n$ ordering clauses (one for each $w$) with the surjective clause $(z_{1,i+1} \vee \ldots \vee z_{n,i+1})$ gives us the new ordering clause $(\neg z_{u,i+2} \vee \neg z_{v,i})$ associated with $u$ and $v$. This clearly requires only $n$ steps and we can do this for all $u < v$ and $j = i+2$. We now continue to apply this argument for $j = i+2, i+3, \ldots, k$ and derive all new ordering clauses in $n$ steps each.

We will construct a tree-like refutation starting with the initial clauses and the new ordering clauses we derived above. We claim that for any $i \in \{1,2,\ldots,k\}$ and for any $1 \leq v_i < v_{i+1} < \ldots < v_k \leq n$, we can derive a

clause that is a subclause of $(\neg z_{v_i,i} \vee \neg z_{v_{i+1},i+1} \vee \ldots \vee \neg z_{v_k,k})$. Let us first see how we will get a refutation given this claim. For $i = k$, the claim says that we can derive a subclause of $\neg z_{v_k,k}$ for all $1 \leq v_k \leq n$. If any of these is a strict subclause, we already have the empty clause. Otherwise, we have $\neg z_{v_k,k}$ for every $v_k$. Resolving all these with the surjective clause $(z_{1,k} \vee \ldots \vee z_{n,k})$ results in the empty clause.

We now prove the claim by induction on $i$. For the base case, fix $i = 1$. For any given $k$ vertices $v_1 < v_2 < \ldots < v_k$, pick an edge $(v_p, v_q)$ that witnesses the fact that these $k$ vertices do not form an independent set. The corresponding edge clause $(\neg z_{v_p,p} \vee \neg z_{v_q,q})$ then works as the required subclause.

For the inductive step, fix $v_{i+1} < v_{i+2} < \ldots < v_k$. We will derive a subclause of $(\neg z_{v_{i+1},i+1} \vee \neg z_{v_{i+2},i+2} \vee \ldots \neg z_{v_k,k})$. By induction, we can derive a subclause of $(\neg z_{v_i,i} \vee \neg z_{v_{i+1},i+1} \vee \ldots \vee z_{v_k,k})$ for any choice of $v_i < v_{i+1}$. If for some such $v_i$, $\neg z_{v_i,i}$ does not appear in the corresponding subclause, then the same subclause works here for the inductive step and we are done. Otherwise for every $v_i < v_{i+1}$, we have a subclause of $(\neg z_{v_i,i} \vee \neg z_{v_{i+1},i+1} \vee \ldots \vee \neg z_{v_k,k})$ that contains $\neg z_{v_i,i}$. Resolving all these subclauses with the surjective clause $(z_{1,i} \vee z_{2,i} \vee \ldots \vee z_{n,i})$, we get the clause $(z_{v_{i+1},i} \vee \ldots \vee z_{v_k,i} \vee \neg z_{u_1,j_1} \vee \ldots \vee \neg z_{u_p,j_p})$, where each $z_{u_c,j_c}$ lies in $\{z_{v_{i+1},i+1}, \ldots, z_{v_k,k}\}$. Observe that for each positive literal $z_{v_q,i}, i+1 \leq q \leq k$, in this clause, $(\neg z_{v_q,i} \vee \neg z_{v_{i+1},i+1})$ is either a 1-1 clause or an ordering clause. Resolving with all these clauses finally gives us $(\neg z_{v_{i+1},i+1} \vee \neg z_{u_1,j_1} \vee \ldots \vee \neg z_{u_p,j_p})$, which is the kind of subclause we wanted to derive. This proves the claim.

We can associate each subclause obtained using the iterative procedure with the tuple $(v_i, v_{i+1}, \ldots, v_k)$ for which it was derived, giving a total of $\sum_{i=1}^{k} \binom{n}{i} \leq k(ne/k)^k$ subclauses. Each of these subclauses is used at most once in the proof. Further, the derivation of each such subclause uses at most $n$ new ordering clauses, each of which can be derived in at most $n^2$ steps. Thus, with enough copies to make the refutation tree-like, the size of the proof is $O(n^3 k(ne/k)^k)$, which is $2^{O(k \log(n/k))}$. □

**Corollary 4.** *Let $G \sim \mathcal{G}_{n,m}$ be a random graph, $\Delta \stackrel{\text{def}}{=} m/n$ and $k$ be any integer for which $G$ does not have a $k$-independent set. With probability $1 - o(1)$ in $n$,*

$$DLL(\alpha_{block}(G,k)) = 2^{O((n/\Delta)\ln^2 \Delta)}.$$

*This bound also holds when 1-1 clauses are removed from $\alpha_{block}(G,k)$.*

*Proof.* Let $k'$ be the smallest integer such that $G$ does not have a $k'$-independent set. Clearly, $k' \leq k$. We first prove the bound for the mapping based encoding. Starting with $\alpha_{map}(G,k)$, we will ignore clauses that involve variables $z_{v,i}$ for $i > k'$ and construct a refutation using only the remaining clauses. The clauses that remain, however, are simply the clauses of $\alpha_{map}(G,k')$. From Theorem 2, we can construct a tree-like resolution refutation of these which is of size $2^{O(k' \log(n/k'))}$ Also, this refutation does not use the 1-1 clauses of $\alpha_{map}(G,k)$. Applying Proposition 3, we can almost certainly bound the size by $2^{O(k_{+\epsilon} \ln(n/k_{-\epsilon}))}$, which is $2^{O((n/\Delta)\ln^2 \Delta)}$. This gives us an upper bound on $DLL(\alpha_{map}(G,k))$ without using 1-1 clauses. Finally, we apply Lemma 2 to get the desired bound on $DLL(\alpha_{block}(G,k))$ without 1-1 clauses. □

**Theorem 3.** *Let $G$ be a graph with $n$ vertices and $m$ edges, $\Delta \stackrel{\text{def}}{=} m/n$ and $k$ be any integer for which $G$ does not have a $k$-independent set. Then*

$$DLL(\alpha_{count}(G,k)) = 2^{O(k \log(n/k))}$$

*Proof.* As in the proof of Theorem 2, we construct a refutation by looking at each size $k$ subset of vertices and using as evidence an edge from that subset.

For every $i, v$ such that $0 \leq i \leq v < n$, we first derive a new counting clause $(\neg y_{v+1,i+1} \vee y_{v,i} \vee y_{v-1,i} \vee \ldots \vee y_{i,i})$ by resolving original counting clauses $(\neg y_{u+1,i+1} \vee y_{u,i+1} \vee y_{u,i})$ for $u = v, v-1, \ldots, i+1$ together, and resolving the result with the counting clause $(\neg y_{i+1,i+1} \vee y_{i,i})$. Next, for any edge $(i,j)$, $i > j$, we resolve the edge clause $(\neg x_i \vee \neg x_j)$ with the counting clauses $(\neg y_{i,i} \vee x_i)$ and $(\neg y_{j,j} \vee x_j)$ to get the clause $(\neg y_{i,i} \vee \neg y_{j,j})$. We call this clause $E_{i,j}$. We will now construct a tree-like refutation using the initial clauses, these new counting clauses and the new $E_{i,j}$ clauses.

We claim that for any $i \in \{1,2,\ldots,k\}$ and for any $1 \leq v_i < v_{i+1} < \ldots < v_k \leq n$ with $v_j \geq j$ for $i \leq j \leq k$, we can derive a subclause of $(\neg y_{v_i,i} \vee y_{v_i-1,i} \vee \neg y_{v_{i+1},i+1} \vee y_{v_{i+1}-1,i+1} \vee \ldots \vee \neg y_{v_k,k} \vee y_{v_k-1,k})$ such that if $y_{v_{j-1},j}$ occurs in the subclause for some $j$, then so does $\neg y_{v_j,j}$. Note that for $v_j = j$, the variable $y_{v_j-1,j}$ does not even exist and will certainly not appear in the subclause. Given this claim, we can derive for $i = k$ a subclause $B_j$ of $(\neg y_{j,k} \vee y_{j-1,k})$ for each $j \in \{k+1, \ldots, n\}$ and a subclause $B_k$ of $\neg y_{k,k}$. If any of these $B_j$'s is the empty clause, we are done. Otherwise every $B_j$ contains $\neg y_{j,k}$. Let $j'$ be the largest index such that $B_{j'}$ does not contain $y_{j'-1,k}$. Since $B_k$ has to be the clause $\neg y_{k,k}$, such a $j'$ must exist. Resolving all $B_j$'s for $j \in \{j', \ldots, k\}$ with each other gives us the clause $y_{n,k}$. Resolving this with the size-$k$ clause $y_{n,k}$ gives the empty clause.

We now prove the claim by induction on $i$. For the base case $i = 1$, fix $1 \leq v_1 < v_2 < \ldots < v_k \leq n$. Pick

an edge $(v_p, v_q)$ that witnesses the fact that these $v_i$'s do not form an independent set. We then have the edge clause $(\neg x_{v_p} \vee \neg x_{v_q})$. Resolving this with the counting clauses $(\neg y_{v_p,p} \vee y_{v_p-1,p} \vee x_p)$ and $(\neg y_{v_q,q} \vee y_{v_q-1,q} \vee x_q)$, we get $(\neg y_{v_p,p} \vee y_{v_p-1,p} \vee \neg y_{v_q,q} \vee y_{v_q-1,q})$, which is a subclause of the desired form.

For the inductive step, fix $v_{i+1} < v_{i+2} < \ldots < v_k$. By induction, we can derive a subclause $C_j$ of $(\neg y_{j,i} \vee y_{j-1,i} \vee \neg y_{v_{i+1},i+1} \vee y_{v_{i+1}-1,i+1} \vee \ldots \vee \neg y_{v_k,k} \vee y_{v_k-1,k})$ for any j in $\{i, i+1, \ldots, v_{i+1}-1\}$. If for some such $j$, neither $\neg y_{j,i}$ nor $y_{j-1,i}$ appears in $C_j$, then this subclause also works here for the inductive step and we are done. Otherwise for every $j$, $C_j$ definitely contains $\neg y_{j,i}$, possibly $y_{j-1,i}$ and other positive or negative occurrences of variables of the form $y_{v',i'}$ where $i' > i$. We now use these $C_j$'s to derive clauses $C'_j$'s such that $C'_j$ contains $\neg y_{j,i}$ but not $y_{j-1,i}$. The other variables appearing in $C'_j$ will all be of the form $y_{v',i'}$ for $i' > i$.

If $\{v_{i+1}, \ldots, v_k\}$ is not an independent set, then there is an edge $(v_p, v_q)$ witnessing this. In this case, we simply use $E_{p,q}$ as the desired subclause and the inductive step is over. Otherwise there must be an edge $(i, v_q)$ from vertex $i$ touching this set. We let $C'_i$ be the clause $E_{i,v_q}$. For $j$ going from $i+1$ to $k$, we do the following iteratively. If $y_{j-1,i}$ does not appear in $C_j$, then we set $C'_j = C_j$. Otherwise we set $C'_j$ to be the clause obtained by resolving $C_j$ with $C'_{j-1}$. If $C'_{j-1}$ does not contain $\neg y_{j,i}$, then it can be used as the desired subclause for this inductive step and we stop the iteration here, otherwise we continue onto the next value of $j$. If we do not derive a desired subclause somewhere along this iterative process, then we end up with all $C'_j$'s containing $\neg y_{j,i}$ but not $y_{j-1,i}$. Resolving all these with the new counting clause $(\neg y_{v_{i+1},i+1} \vee y_{v_{i+1}-1,i} \vee y_{v_{i+1}-2,i} \vee \ldots \vee y_{i,i})$ finally gives us a subclause of the desired form. This proves the claim.

We can associate each subclause obtained using the iterative procedure with the tuple $(v_i, v_{i+1}, \ldots, v_k)$ for which it was derived, giving a total of $\sum_{i=1}^{k} \binom{n}{i} \leq k(ne/k)^k$ subclauses. Each of these subclauses is used at most once in the proof. Further, the derivation of each such subclause uses one new counting clause and one new clause $E_{i,j}$, each of which can be derived in at most $n$ steps. Thus, with enough copies to make the refutation tree-like, the size of the proof is $O(nk(ne/k)^k)$, which is $2^{O(k \log(n/k))}$. □

## 9. Directions for Further Work

A natural open problem at this point is to see if one can get similar lower bounds on the complexity of independent sets in random graphs for more powerful proof systems such as Cutting Planes [14, 7] and Frege systems [11]. This is interesting in its own right and also because some of the very simple algorithms for finding large independent sets do not seem to be captured by resolution. For instance, the algorithm of Robson [18] uses the following idea: if a vertex $v$ is not included in a maximum independent set then w.l.o.g. we might as well assume that at least two of its neighbors are included. In other words, if only one neighbor of $v$ is chosen, we don't lose anything by choosing $v$ instead. It is, however, not clear how one could translate this simple "without loss of generality" argument into a resolution proof. In Robson's paper, this is only applied in a limited way when the neighborhood set is of constant size and only one such set of neighbors is remembered at a time. This makes it possible to translate the reasoning into a small resolution proof. However, in a more general search, this w.l.o.g. idea does not seem to be captured by resolution arguments.

On another front, we now have exponential resolution lower bounds for random $k$-CNF formulas [9, 3], $k$-coloring of random graphs [2] and independent sets in random graphs. It would be interesting to understand more fully which coNP-complete problems require large resolution proofs for random instances.

## References


[1] D. Angluin and L. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155–193, 1979.

[2] P. Beame, J. Cullberson, and M. D. The resolution complexity of random graph $k$-colorability. In preparation, 2000.

[3] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability of random $k$-CNF formulas. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 561–571, Dallas, TX, May 1998.

[4] P. W. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings 37th Annual Symposium on Foundations of Computer Science*, pages 274–282, Burlington, VT, Oct. 1996. IEEE.

[5] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 517–526, Atlanta, GA, May 1999.

[6] B. Bollobás. *Random Graphs*. Academic Press, London, 1985.

[7] M. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, Sept. 1997.

[8] V. Chvátal. Determining the stability number of a graph. *SIAM Journal on Computing*, 6(4):643–662, 1977.

[9] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.