Counting Independent Sets in Cocomparability Graphs [*]

Martin Dyer Haiko Müller

School of Computing, University of Leeds, Leeds LS2 9JT, UK
{M.E.Dyer|H.Muller}@leeds.ac.uk

Abstract

We show that the number of independent sets in cocomparability graphs can be counted in linear time, as can counting cliques in comparability graphs. By contrast, counting cliques in cocomparability graphs and counting independent sets in comparability graphs are #P-complete. We extend these results to counting maximal cliques and independent sets. We also consider the fixed-parameter versions of counting cliques and independent sets of given size k. Finally, we combine the results to show that both counting cliques and independent sets in permutation graphs are in linear time.

Keywords: comparability graph, cocomparability graph, permutation graph, independent set, counting, linear time algorithm, computational complexity

1 Introduction

Counting independent sets in graphs is known to be #P-complete in general [14], even for several restricted cases [15]. Even approximate counting is NP-hard, and unlikely to be in polynomial time for bipartite graphs [5]. However, smaller graph classes may permit polynomial time algorithms. For example, [11] gives a linear time algorithm for counting independent sets in bipartite permutation graphs. Here "linear time" means computable with a number of arithmetic operations and comparisons linear in the size of the graph. Similar results are known for *chordal graphs* [13] and *tree-convex graphs* [10]. For information on these and other graph classes, see [2].

Counting cliques in a graph is equivalent to counting independent sets in the complementary graph. Thus, for general graphs, these two problems have equivalent complexity. However, for graph classes the two problems may have very different complexity.

Here we consider the problems of counting independent sets and cliques in three classes of graphs: *cocomparability graphs*, *comparability graphs* and *permutation graphs*. See section 2 below for definitions. In each case we show that either the counting problem can be accomplished in linear time, or else counting is #P-complete.

In section 3 we present a simple linear time algorithm for counting independent sets in cocomparability graphs, and show that counting cliques is #P-complete. We extend this to counting maximal independent sets, and independent sets of a given size, and maximal independent sets of a given size. In section 4, we modify this to give linear time algorithms for counting cliques in comparability graphs, and show that counting independent sets is #P-complete.

[*] The work reported in this paper was partially supported by EPSRC research grant EP/S016562/1 "Sampling in hereditary classes".

Together, these results imply simple linear time algorithms for counting both cliques and independent sets in permutation graphs, as we discuss in section 5. Thus our results give strong generalisations of the algorithm of [11]. Table 1 summarises the results of this note.

2 Preliminaries and notation

For integers n and n' with $n \leq n'$ the set $\{n, n+1, \ldots, n'\}$ of consecutive integers will be abbreviated as $[n, n']$.

Let $P = (V, \prec)$ be a finite partial order, with $|V| = n$. The *cover relation* $\prec\!\!\cdot$ of $\prec$ is its transitive reduction, defined by $u \prec\!\!\cdot\, v$ if $u \prec v$ and there is no $w \in V$ with $u \prec w$ and $w \prec v$. By $u \preccurlyeq v$ we mean $u \prec v$ or $u = v$.

A *chain* in $(V, \prec)$ is a set $S \subseteq V$ of pairwise comparable elements. That is, the restriction of $\prec$ to S is a linear order. A chain $S = \{v_i \mid i \in [1, k]\}$ with $v_{i-1} \prec v_i$ for all $i \in [2, k]$ is *tight* if $v_{i-1} \prec\!\!\cdot\, v_i$ holds for all $i \in [2, k]$.

A *linear extension* of $(V, \prec)$ is a permutation $v_1, v_2, \ldots, v_n$ of V such that $v_i \prec v_j$ implies $i < j$.

Let $G = (V, E)$ be a graph with vertex set V, with $|V| = n$ and $|E| = m$. Here all graphs are finite, undirected and simple, unless stated otherwise. We will denote the complementary graph by $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{\{u, v\} \mid u, v \in V, \{u, v\} \notin E, u \neq v\}$. Let $\bar{m} = |\bar{E}|$, that is $\bar{m} = \binom{n}{2} - m$, and $m^* = \min\{m, \bar{m}\}$.

A *clique* in a graph is a set of pairwise adjacent vertices. A set of pairwise non-adjacent vertices is *independent.* Therefore a clique of G is an independent set in the complement of G and vice versa. A *maximal* clique or independent set is maximal with respect to set inclusion. By contrast, *maximum* refers to the largest cardinality.

A partial order $P = (V, \prec)$ can be considered as digraph (V, A) with vertex set V and arcs $(x, y) \in A$ for all pairs such that $x \prec y$. The comparability graph for P is the underlying undirected graph (V, E) of (V, A). That is, $E = \{\{x, y\} \mid x \prec y\}$. Then a graph $G = (V, E)$ is a *comparability graph* if there is a partial order P on the set V such G is the comparability graph of P. Given a comparability graph G, a corresponding partial order P and a linear extension can be computed in $O(m + n)$ (linear) time [12]. Note, however, that this algorithm does not certify that G is a comparability graph. Recognition is currently only known in $O(n + m \log n)$ time. Moreover, in 3.3 and 3.4 we assume that access to the cover relation $\prec\!\!\cdot$ is possible in constant time. Therefore we have to compute the transitive reduction of $P = (V, \prec)$. In general this is hard as matrix multiplication, see Theorem 3 in [1]. We will sidestep these issues by assuming the input comparability graph is given with a transitive orientation P, its transitive reduction $(V, \prec\!\!\cdot)$, and a linear extension of P.

The complements of comparability graphs are called *cocomparability graphs*, so these are the incomparability graphs of partially ordered sets. In the incomparability graph of $P = (V, \prec)$ two vertices in V are adjacent if they are $\prec$-incomparable. Again, recognition is not known to be possible in linear time, but a linear time algorithm can compute P and a linear extension for

	all	maximal	size k	size k maximal
independent sets	$O(n + m^*)$ Theorem 5	$O(n + m^*)$ Theorem 9	$O(k(n + m^*))$ Theorem 13	$O(k(n + m^*))$ Theorem 17
cliques	#P-complete Theorem 6	#P-complete Theorem 10	#W[1]-complete Theorem 14	open

Table 1: Complexity of counting independent sets and cliques in cocomparability graphs.

the complementary comparability graph [7, 12].

A graph is a *permutation graph* if there is a permutation $v_1, v_2, \dots, v_n$ of $V = [1, n]$ such that $\{v_i, v_j\} \in E$ if and only if $i < j$ implies $v_i > v_j$. It is not difficult to show that G is a permutation graph if and only if it is both a comparability graph and a cocomparability graph. Permutation graphs can be recognised, and a permutation ordering obtained, in linear time [8], so we need no assumptions on the input graph.

These are well-established classes of graphs, and are all subclasses of *perfect graphs.* Important subclasses of comparability graphs are *bipartite graphs* and *cographs. Interval graphs* are an important subclass of cocomparability graphs (they are graphs that are both chordal and cocomparability). Permutation graphs are an important subclass of both comparability and cocomparability graphs.

3 Cocomparability graphs

Let $G = (V, E)$ be a cocomparability graph and let $\prec$ be a partial order on V, with linear extension $v_1, v_2, \dots, v_n$. We will extend the poset $(V, \prec)$ by a unique minimal element $\bot \notin V$ and a unique maximal element $\top \notin V \cup \{\bot\}$. That is, we define $\bot \prec v$ and $v \prec \top$ for all $v \in V$, and (if not enforced by transitivity) $\bot \prec \top$. Let $S^+ = S \cup \{\bot, \top\}$, for any $S \subseteq V$. Then we denote the extended partial order by $(V^+, \prec)$, If $\bot, v_1, v_2, \dots, v_n, \top$ is linear extension of $(V^+, \prec)$, and $S \subseteq V$, we write $\max(S)$ for v_i such that $i = \max\{j : v_j \in S\}$,

Lemma 1. *Let $G = (V, E)$ be a cocomparability graph which is the incomparability graph of a poset $(V, \prec)$, and let $(V^+, \prec)$ be the above extension. Then*

(1) a set $S \subseteq V$ is independent in G if and only if S^+ is a chain of $(V^+, \prec)$;

(2) a set $S \subseteq V$ is maximally independent in G if and only if S^+ is a tight chain of $(V^+, \prec)$.

Proof. Two vertices in V are $\prec$-comparable if and only if they are non-adjacent in G. The extra elements $\bot$ and $\top$ are comparable to all vertices in V. Together these imply property (1).

To see (2) we first consider an independent set S of G that is not maximal independent. Hence there is a vertex $v \in V$ such that $S \cup \{v\}$ is still independent. By property (1) the set $S^+ \cup \{v\}$ is a chain, and hence S^+ is also a chain, but not a tight one.

Now let S be a maximal independent set of G. So by property (1) the set S^+ is a chain of $(V^+, \prec)$. Since S is a maximal independent set of G, for all $v \in V$ the set $S \cup \{v\}$ is not independent in G. By property (1) the set S^+ is not a chain, and therefore S^+ is a tight chain of $(V^+, \prec)$. □

For every vertex $v \in V^+$ let $G_v = G[\{u \in V \mid u \preccurlyeq v\}]$. Especially, $G_\bot = (\varnothing, \varnothing)$ and $G_\top = G$. Let $V_v = \{u \in V^+ \mid u \preccurlyeq v\}$. For every $v \in V^+$, the relation $\prec$ restricted to V_v is a partial order with unique minimal element $\bot$ and unique maximal element v.

3.1 Independent sets

For every vertex $v \in V^+$ let $A(v)$ be the set of independent sets of S of G_v with $v \in S$, and $a(v) = |A(v)|$. That is, $a(\top)$ is the number of independent sets of G. We can evaluate A and a recursively as follows:

$$
\begin{aligned}
A(\bot) &= \{\varnothing\} & a(\bot) &= 1 \\
A(v_i) &= \bigcup_{u \prec v_i} \{S \cup \{v_i\} \mid S \in A(u)\} & a(v_i) &= \sum_{u \prec v_i} a(u) & (i \in [1,n]) \\
A(\top) &= \bigcup_{u \prec \top} A(u) & a(\top) &= \sum_{u \prec \top} a(u)
\end{aligned}
$$

The sets $A(v)$ can be exponential in size (for example if $E = \varnothing$), but the recurrence for a can be evaluated efficiently. We use the linear extension of $\prec$ to evaluate the equations above in the order $\bot, v_1, v_2, \ldots, v_n, \top$, as shown. At the end $a(\top)$ is the number of independent sets in G.

We show that the recurrence above is correct by proving a sequence of lemmas, which themselves are proven directly or by induction on $\prec$.

Lemma 2. *For all $v \in V^+$ every set $S \in A(v)$ is independent in G_v and $v \in V$ implies $v \in S$.*

Proof. The base of the induction is for $v = \bot$. Clearly $\varnothing$ is the unique independent set of $G_\bot$.

Now let $v \in V$. Every set $S \in A(v)$ contains the vertex v and vertices in $A(u)$ for $u \prec v$. By induction hypothesis $S \setminus \{v\}$ is an independent set of some G_u. Therefore S only contains vertices of G_v. Moreover $S \setminus \{v\}$ is a chain in $(V, \prec)$. By transitivity, $S \setminus \{v\} \in A(u)$ and $u \prec v$ imply that S is also a chain in $(V, \prec)$ and therefore an independent set of G_v.

The latter argument also applies to $v = \top$. □

Lemma 3. *Every nonempty independent set S of G is contained in $A(\max(S))$.*

Proof. Let S be an independent set of G. This implies that S is a chain of $(V, \prec)$ and therefore S has a unique $\prec$-maximal element v. We have $\varnothing \in A(\bot)$ and, by induction hypothesis, $S \setminus \{v\} \in A(u)$ for some $u \prec v$. Consequently we have $S \in A(v)$, and clearly $v = \max(S)$. □

Lemma 4. *For different $u, v \in V \cup \{\bot\}$ we have $A(u) \cap A(v) = \varnothing$.*

Proof. If u and v are $\prec$-incomparable then these are adjacent vertices of G. Hence no independent set of G can contain both u and v. By Lemma 2 for all $S \in A(u)$ and all $T \in A(v)$ we have $u \in S \setminus T$ and $v \in T \setminus S$, that is, $S \neq T$. Consequently $A(u) \cap A(v) = \varnothing$ holds.

Now let u and v be $\prec$-comparable. By symmetry we may assume $u \prec v$, especially $v \in V$. Again, Lemma 3 implies, for all $S \in A(u)$ and all $T \in A(v)$, that $v \in T \setminus S$ holds. As before this means $S \neq T$, and therefore $A(u) \cap A(v) = \varnothing$. □

We can implement the above recurrences for a directly, but it is clear that the time complexity is $O(n+\bar{m})$, not $O(n+m)$, since the summations are over the edges of $\bar{E}$. If $\bar{m} < m$, this clearly implies $O(n+m)$ time, but otherwise we correct this as follows.

For all $v \in V^+$ we define $t(v)$ by

$$
\begin{aligned}
t(\bot) &= 0 \\
t(v_i) &= a(\bot) + \sum_{j=1}^{i-1} a(v_j) & (i \in [1,n]) \\
t(\top) &= a(\bot) + \sum_{j=1}^{n} a(v_j)
\end{aligned}
$$

The values of t and a are mutually recursive:

$$\begin{aligned}
t(\bot) &= 0 & a(\bot) &= 1 & \\
t(v_1) &= 1 & a(v_1) &= 2 & \\
t(v_i) &= t(v_{i-1}) + a(v_{i-1}) & a(v_i) &= t(v_i) - \sum_{j<i,\, v_j \not\prec v_i} a(v_j) & (i \in [2, n]) \\
t(\top) &= t(v_n) + a(v_n) & a(\top) &= t(\top) &
\end{aligned}$$

The recurrence for $t(v_i)$ is obvious, and for $a(v_i)$ we have

$$t(v_i) - \sum_{j<i,\, v_j \not\prec v_i} a(v_j) = a(\bot) + \sum_{j=1}^{i-1} a(v_j) - \sum_{j<i,\, v_j \not\prec v_i} a(v_j) = \sum_{u \prec v_i} a(u) = a(v_i)\,,$$

as required. Now the summations are over the edges of E, and the algorithm is $O(n+m)$ time. Thus we have

Theorem 5. *Given a cocomparability graph on n vertices, we can compute the number of its independent sets in time $O(n+m^*)$.*

Proof. Lemmas 2 and 3 imply that, for all $v \in V^+$, the sets $A(v)$ defined by the recurrence contains all independent sets of G_v, especially $A(\top)$ is the set of all independent sets of G. It remains to show that $a(v) = |A(v)|$. To see this we observe that the unions over all $u \prec v$ are always disjoint by Lemma 4.

Our algorithm just recurrences for $a(v)$ for all $v \in V^+$ in the order of linear extension of $\prec$. Then $a(\top)$ is the number of independent sets of G. Assuming additions can be performed in constant time, this takes $O(n+m^*)$ time, since each edge or non-edge of G is used exactly once in the recurrences. □

The alternative evaluation of the recurrences which leads to $O(n+m^*)$ can be used in all the algorithms below, and we will not elaborate further on this.

On the other hand, we have the following.

Theorem 6. *It is #P-complete to count cliques in a cocomparability graph.*

Proof. Counting cliques in cocomparability graphs is equivalent to counting independent sets in comparability graphs. Bipartite graphs are a subclass of comparability graphs. It is #P-complete to count independent sets in bipartite graphs [14, 15]. □

In fact, even approximately counting cliques in cocomparability graphs appears to be hard, by the same argument, being equivalent to the canonical problem #BIS [5].

Finally, we note that counting independent sets in cocomparability graphs is equivalent to counting chains in partially ordered sets, and counting cliques is equivalent to counting antichains. See [14] and [5], where antichains are called downsets. Thus the results of this paper could be recast in the language of partial orders.

3.2 Maximal independent sets

Similarly we can compute the number of maximal independent sets. For every vertex $v \in V^+$ let $B(v)$ denote the set of maximal independent sets S of G_v with $v \in S$, and let $b(v) = |B(v)|$. We can compute B and b as follows:

$$
\begin{aligned}
B(\bot) &= \{\varnothing\} & b(\bot) &= 1 & \\
B(v_i) &= \bigcup_{u \prec\!\cdot\, v_i} \{S \cup \{v\} \mid S \in B(u)\} & b(v_i) &= \sum_{u \prec\!\cdot\, v_i} b(u) & (i \in [1,n]) \\
B(\top) &= \bigcup_{u \prec\!\cdot\, \top} B(u) & b(\top) &= \sum_{u \prec\!\cdot\, \top} b(u) &
\end{aligned}
$$

The only difference to A and a is that the partial order $\prec$ is replaced by its cover relation $\prec\!\cdot$, as anticipated in Lemma 1. For all $u \prec v$ and all maximal independent sets S' of G_u the set $S' \cup \{v\}$ is maximal independent if and only if there is no $w \in V$ that is $\prec$-between u and v, hence if and only if $u \prec\!\cdot\, v$.

The recurrence for B and b can be shown to be correct by arguments similar to the ones given above to justify the recurrence for A and a.

Lemma 7. *For all $v \in V^+$ every set $S \in B(v)$ is a maximal independent set of G_v and $v \in V$ implies $v \in S$.*

Proof. The base of the induction is for $v = \bot$. Clearly $\varnothing$ is the unique independent set of $G_\bot$.

Now let $v \in V$. Every set $S \in B(v)$ contains the vertex v and vertices in $B(u)$ for $u \prec\!\cdot\, v$. By induction hypothesis $S \setminus \{v\}$ is a maximal independent set of some G_u. Therefore S only contains vertices of G_v. Moreover $S \setminus \{v\}$ is a tight chain in $(V_u, \prec)$. Since $u \prec\!\cdot\, v$ implies $u \prec v$ and because $\prec$ is transitive, $S \setminus \{v\} \in B(u)$ and $u \prec\!\cdot\, v$ imply that S is also a tight chain in $(V_v, \prec)$ and therefore a maximal independent set of G_v.

A similar argument proves the assertion for $v = \top$. □

Lemma 8. *For every $v \in V^+$ every maximal independent set S of G_v satisfies $S \in B(v)$ and $v \in V$ implies $v \in S$.*

Proof. Let S be a maximal independent set of G_v. By Lemma 1, $S \cup \{\bot\}$ is a tight chain of $(V_v, \prec)$ with minimal element $\bot$ and maximal element v. Therefore $v \in V$ implies $v \in S$. We show $S \in B(v)$ by induction. For $v = \bot$ we have $\varnothing \in B(\bot)$. Otherwise, by induction hypothesis and the tightness stated above, $S \setminus \{v\} \in B(u)$ for some $u \prec\!\cdot\, v$. Consequently we have $S \in B(v)$. □

Theorem 9. *Given a cocomparability graph on n vertices, we can compute the number of its maximal independent sets in $O(n + m^*)$ time.*

Proof. Lemmas 7 and 8 imply that, for all $v \in V^+$, the sets $B(v)$ defined by the recurrence contains indeed all maximal independent sets of G_v, especially $B(\top)$ is the set of all maximal independent sets of G.

It remains to show that $b(v) = |B(v)|$. Again, the unions over all $u \prec\!\cdot\, v$ are always disjoint because $B(u) \subseteq A(u)$ holds for all u, and by Lemma 4.

The time analysis is essentially the same as in the proof of Theorem 5. □

Corresponding to Theorem 6, we also have the following

Theorem 10. *It is #P-complete to count maximal cliques in a cocomparability graph.*

Proof. By the same argument as Theorem 6, using that it is #P-complete to count maximal independent sets in bipartite graphs [9]. □

Approximately counting maximal independent sets in bipartite graphs, a subclass of comparability graphs, is #SAT-hard [6].

3.3 Independent sets of size k

Next we consider independent sets of size exactly k for some fixed value of $k \in [0, n]$. For every vertex $v \in V^+$ and every integer $i \in [0, k]$, the set $C(v, i)$ of independent sets S of G_v with $v \in S$ and $|S| = i$, and size $c(v, i)$, satisfy the following recurrences:

$$\begin{array}{llll}
C(\bot, 0) = \{\varnothing\} & c(\bot, 0) = 1 & \\
C(\bot, i) = \varnothing & c(\bot, i) = 0 & (i \in [1, k]) \\
C(v_j, 0) = \varnothing & c(v_j, 0) = 0 & (j \in [1, n]) \\
C(v_j, i) = \displaystyle\bigcup_{u \prec v_j} \{S \cup \{v_j\} \mid S \in C(u, i-1)\} & c(v_j, i) = \displaystyle\sum_{u \prec v_j} c(u, i-1) & (i \in [1, k],\ j \in [1, n]) \\
C(\top, i) = \displaystyle\bigcup_{u \prec \top} C(u, i) & c(\top, i) = \displaystyle\sum_{u \prec \top} c(u, i) & (i \in [0, k])
\end{array}$$

The correctness of these recurrences is again based on the fact that, for every $v \in V$ and every $i \in [1, k]$, every independent set S of G_v with $v \in S$ and $|S| = i$ there is an independent set S' of G_u of size $i - 1$ for some $u \prec v$, where $u = \bot$ if $S' = \varnothing$ and otherwise u is the $\prec$-maximal vertex in S'. For $v = \top$ we have $|S| = |S'|$ because $\top \notin V$. Then we have the following.

Lemma 11. *For all $v \in V$ and all $i \in [1, k]$ every set $S \in C(v, i)$ is a independent set of G_v with $|S| = i$ and $v \in S$.*

Lemma 12. *For every $i \in [1, k]$ every independent set S of size i is contained in $C(\max(S), i)$.*

Theorem 13. *Given a cocomparability graph on n vertices and a number $k \in [0, n]$, we can compute the number of its independent sets of size exactly k in time $O(k(n + m^*))$. In time $O(n^2 + nm^*)$ we can do this for all $k \in [0, n]$.*

Proof. Lemmas 2 and 3 imply that, for all $v \in V^+$, the sets $A(v)$ defined by the recurrence contains indeed all independent sets of G_v, especially $A(\top)$ is the set of all independent sets of G. It remains to show that $a(v) = |A(v)|$. To see this we observe that the unions over all $u \prec v$ are always disjoint by Lemma 4.

Our algorithm just evaluates the sums for $a(v)$ for all $v \in V^+$ in an order that is a linear extension of $\prec$. Then $a(\top)$ is the number of independent sets of G. Assuming additions can be performed in constant time, this takes $O(n^2)$ time. □

As a by-product the algorithm of Theorem 13 computes the size $\alpha(G)$ of a maximum independent set of G, which is $\max\{i \mid c(\top, i) > 0\}$, and the number of maximum independent set in G, which is $c(\top, \alpha(G))$. Since every maximum independent set is also maximal, the algorithm from Theorem 17 can be used as well and should be faster on average.

Once the number $c(\top, i)$ of independent sets of size exactly i has been determined for all $i \in [0, \alpha(G)]$ we can evaluate the independent set polynomial $\sum_{i=0}^{\alpha(G)} c(\top, i)x^i$ for all values of x.

We also have the following fixed-parameter hardness result for counting k-cliques in cocomparability graphs.

Theorem 14. *It is #W[1]-complete to count k-cliques in a cocomparability graph.*

Proof. By the same argument as Theorem 6, using that it is #W[1]-complete to count independent sets of size k in bipartite graphs [4, Theorem 4]. □

3.4 Maximal independent sets of size k

For every vertex $v \in V^+$ and every integer $i \in [0, k]$, let $D(v, i)$ be the set of maximal independent sets S of G_v with $v \in S$ and $|S| = i$, and $d(v, i) = |D(v, i)|$. We have

$$\begin{alignedat}{3}
D(\bot, 0) &= \{\varnothing\} & \qquad d(\bot, 0) &= 1 & & \\
D(\bot, i) &= \varnothing & d(\bot, i) &= 0 & \qquad & (i \in [1, k]) \\
D(v_j, 0) &= \varnothing & d(v_j, 0) &= 0 & & (j \in [1, n]) \\
D(v, i) &= \bigcup_{u \lessdot v} \bigl\{ S \cup \{v\} \mid S \in D(u, i-1) \bigr\} & d(v, i) &= \sum_{u \lessdot v} d(u, i-1) & & (i \in [1, k], j \in [1, n]) \\
D(\top, i) &= \bigcup_{u \lessdot \top} D(u, i) & d(\top, i) &= \sum_{u \lessdot \top} d(u, i) & & (i \in [0, k])
\end{alignedat}$$

The following are proved similarly to the corresponding results above.

Lemma 15. *For all $v \in V$ and all $i \in [1, k]$ every set $S \in D(v, i)$ is a maximal independent set of G_v with $|S| = i$ and $v \in S$.*

Lemma 16. *For every $v \in V^+$ and $i \in [1, k]$ every maximal independent set S of size i in G_v is contained in $D(v, i)$, and $v \in V$ implies $v \in S$.*

Theorem 17. *Given a cocomparability graph on n vertices and a number $k \in [0, n]$, we can compute the number of its maximal independent sets of size exactly k in time $O(k(n + m^*))$. In time $O(n^2 + nm^*)$ we can do this for all $k \in [0, n]$.*

We have no corresponding hardness result in this case, since the complexity of counting *maximal* k-independent sets in bipartite graphs appears to be open.

4 Comparability graphs

Counting independent sets in a cocomparability graph G is equivalent to counting cliques in its complement $\bar{G}$. Since our algorithms are symmetrical between G and $\bar{G}$, all the results of section 3 remain true by interchanging the words "comparability" and "cocomparability", and the words "clique" and "independent set". Therefore, we will not detail the modified results.

5 Permutation graphs

Permutation graphs are both comparability and cocomparability graphs, so the algorithms of section 3 are valid both for counting independent sets in permutation graphs and (with obvious modifications) for counting cliques in permutation graphs. The result of [11] for counting independent sets in bipartite permutation graphs is a special case.

References

[1] A. V. Aho, M. R. Garey and J. D. Ullman, The transitive reduction of a directed graph, *SIAM Journal on Computing* **1** (1972) 131–137.

[2] A. Brandstädt, V. B. Le and J. P. Spinrad, Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Application, Philadelphia (1999).

[3] D. G. Corneil, S. Olariu and L. Stewart, Asteroidal triple-free graphs, *SIAM Journal on Discrete Mathematics* **10** (1997) 399-430.

[4] R. Curticapean, H. Dell, F. V. Fomin, L. A. Goldberg and J. Lapinskas, A fixed-parameter perspective on #BIS, in *Proc. 12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, Leibniz International Proceedings in Informatics (LIPIcs) **89** (2018) 13:1–13:13.

[5] M. Dyer, L. A. Goldberg, C. Greenhill and Mark Jerrum, On the relative complexity of approximate counting problems, *Algorithmica* **38** (2003) 471–500.

[6] L. A. Goldberg, R. Gysel and J. Lapinskas, Approximately counting locally-optimal structures, *Journal of Computer and System Sciences* **82** (2016) 1144–1160.

[7] E. Köhler and L. Mouatadid, Linear time LexDFS on cocomparability graphs, in *Algorithm Theory (SWAT 2014)*, Springer LNCS 8503 (2014) 319–330.

[8] D. Kratsch, R. McConnell, K. Mehlhorn and J. P. Spinrad, Certifying algorithms for recognizing interval graphs and permutation graphs, *SIAM Journal on Computing* **36** (2003) 326–353.

[9] M.-S. Lin, Counting independent sets and maximal independent sets in some subclasses of bipartite graphs, *Discrete Applied Mathematics* **251** (2018) 236–244.

[10] M.-S. Lin and C.-M. Chen, Counting independent sets in tree convex bipartite graphs. *Discrete Applied Mathematics* **218** (2017) 113–122.

[11] M.-S. Lin and C.-M. Chen, Linear-time algorithms for counting independent sets in bipartite permutation graphs. *Information Processing Letters* **122** (2017) 1–7.

[12] R. M. McConnell and J. P. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, in *Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms* (1994) 536–545.

[13] Y. Okamoto, T. Uno and R. Uehara, Counting the number of independent sets in chordal graphs. *Journal of Discrete Algorithms* **6** (2008) 229–242.

[14] J. S. Provan and M. O. Ball, The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing* **12** (1983) 777–788.

[15] S. Vadhan, The complexity of counting in sparse, regular, and planar graphs, *SIAM Journal on Computing* **31** (2001) 398–427.