Counting Independent Sets and its Variation in Cocomparability Graphs

Sheng-Huang Su and Min-Sheng Lin

Department of Electrical Engineering
National Taipei University of Technology
Taipei 106, Taiwan, R.O.C.

Abstract

This paper addresses the problems of counting the independent sets, the maximal independent sets, the independent dominating sets, and the independent perfect dominating sets in cocomparability graphs.

Keywords: Cocomparability graph, Independent sets, Maximal independent sets, Independent dominating sets, Independent perfect dominating sets

1 Introduction

A directed graph that is obtained from an undirected graph $G$ by assigning a direction to each edge is called an orientation of $G$. A directed graph $G=(V,E)$ is transitive if $(u,v)\in E$ and $(v,w)\in E$ imply $(u,w)\in E$. An undirected graph is a comparability graph if it has a transitive orientation. A cocomparability graph is the complement of a comparability graph. Cocomparability graphs are usefully characterized by the cocomparability ordering. The cocomparability ordering is a numbering $1, 2, ..., n$ of $V$ such that if $u< v< w$ and $(u,w)\in E$, then either $(u,v)\in E$ or $(v,w)\in E$, or, equivalently, if $u< v< w$ and $(u,v), (v,w)\not\in E$ then $(u,w)\not\in E$. A graph $G$ is a cocomparability graph if and only if $G$ has a cocomparability ordering. Throughout this paper, vertices of a cocomparability graph are assumed to be numbered in a cocomparability ordering by $1, 2, ..., n$. 
An independent set (IS) is a subset of vertices of a graph such that there are no two vertices which are adjacent. Equivalently, each edge of the graph has at most one endpoint that lies in the IS. The maximal independent set (MIS) of a graph is an IS that is not a subset of any other IS in the graph.

A dominating set (DS) in a graph $G=(V,E)$ is a subset $D$ of $V$ such that every vertex that is not in $D$ is adjacent to at least one vertex in $D$. An independent dominating set (IDS) of a graph $G$ is a set of vertices of $G$ that is both independent and dominating in $G$. An IDS $D$ is an independent perfect dominating set (IPDS) if every vertex not in $D$ is adjacent to exactly one vertex in $D$.

Let $G=(V,E)$ be a graph. A subset $C \subseteq V$ is called a vertex cover (VC) of $G$ if and only if every edge in $E$ has at least one endpoint in $C$. In graph theory, VC and IS are dual to each other. Since every edge in $E$ is incident on a vertex in a VC $C$, no edge has neither endpoint in $C$. Thus, $V-C$ must be an IS.

This study investigates problems associated with the number of ISs. The number of ISs in a graph is important, particularly because it generally arises in problems that are related to network reliability [3,5] and art-gallery problem [6]. However, computing the number of ISs (VCs) is a #P-complete problem [5]. Valiant [7] defined the class of #P-complete problems. The class of #P problems includes problems that involve counting access computations for problems in NP, while the class of #P-complete problems includes the hardest problems in #P. As is widely known, all exact algorithms for solving these problems have exponential time complexity, making efficient algorithms for this class of problems unlikely to be developed. However, this complexity can be reduced by considering only a restricted subclass of #P-complete problems.

Lin and Chen [4] presented $O(n^2)$ time algorithms for counting VCs (or ISs), minimal VCs (or MISs) in a trapezoid graph with $n$ vertices. This paper extends these results to cocomparability graphs, which is a superclass of trapezoid graphs. The problem is also extended to the counting of IDSs and IPDSs. Table 1 presents the results obtained in this paper.

<table>
<thead>
<tr>
<th>Counting problems in a cocomparability graph</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td># of ISs</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td># of MISs</td>
<td>$O(n^{\omega})$</td>
</tr>
<tr>
<td># of IDSs</td>
<td>$O(n^{\omega})$</td>
</tr>
<tr>
<td># of IPDSs</td>
<td>$O(n^{\omega})$</td>
</tr>
</tbody>
</table>

### Table 1. Summary of results;

$O(n^{\omega})$ is the complexity of matrix multiplication, $\omega<2.3727$ [2,8].

2 Preliminaries

This section presents the preliminaries on which the desired algorithms depend.
Consider a cocomparability graph \( G = (V, E) \) with vertices labeled in a cocomparability ordering, \( V = \{1, 2, 3, \ldots, n\} \). To simplify the implementation of the algorithm, two dummy vertices 0 and \( n+1 \) are added to graph \( G \).

Vertex \( u \) is said to be a predecessor of vertex \( v \), denoted \( u \prec v \), if \( u \prec v \) and \( (u,v) \notin E \). From the cocomparability ordering, if \( u \prec v \) and \( v \prec w \), then \( u \prec w \). Let \( X \) be an \( (n+2) \times (n+2) \) Boolean matrix of the relation \( \prec \) on \( V \cup \{0, n+1\} \), where \( X(u,v)=1 \) if \( u \prec v \) and \( X(u,v)=0 \) otherwise. Given the adjacency matrices and the cocomparability ordering of a cocomparability graph with \( n \) vertices, \( X \) can be obtained in \( O(n^2) \) time.

Vertex \( u \) is said to be an immediate predecessor of vertex \( v \), denoted \( u \ll v \), if \( u \prec v \neq w \prec v \) and no vertex \( w \) exists such that \( u \ll w \prec v \). Let \( Y \) be an \( (n+2) \times (n+2) \) Boolean matrix of the relation \( \ll \) on \( V \cup \{0, n+1\} \), where \( Y(u,v)=1 \) if \( u \ll v \) and \( Y(u,v)=0 \) otherwise. Consider the Boolean matrix product \( X^2 = X \cdot X \) defined as \( X^2(u,v) = \bigvee_{w=0}^{n+1} X(u,w) \land X(w,v) \). Now \( X(Y(u,v) = 1 \) which means that for some \( w \in V \), \( X(u,v) = 1 \) and \( X(w,v) = 1 \); that is, \( u \ll w \) and \( w \ll v \). Therefore, by the definition of the immediate predecessor, \( Y \) can be expressed as \( Y = X^2 \). The bottleneck of the computation of \( Y \) is the pre-calculation of \( X^2 \). The Boolean matrix product \( X^2 \) can be obtained by treating \( X \) as an integer matrix; computing the integer matrix product, and transforming the product matrix into a Boolean matrix. Let \( O(n^w) \) be the current best complexity of the algorithm for multiplying two \( n \times n \) matrices. That \( w < 2.376 \) [2] has long been known and this condition has been recently improved in unpublished work to \( w < 2.3727 \) [8].

### 3 Counting independent sets

The section provides algorithms for counting ISs and MISs in a cocomparability graph. Given a graph \( G \), the set of all ISs and MISs are denoted by \( IS(G) \) and \( MIS(G) \), respectively.

#### 3.1 \( O(n^2) \)-time algorithm for counting ISs

Let \( N_G[v] \) denote the closed neighborhood of vertex \( v \) in \( G \). The following lemma [4] is required for the proof in Theorem 1.

**Lemma 1.** [4] For a graph \( G \),

\[
|IS(G)| = |IS(G \setminus v)| + |IS(G \setminus N_G[v])|
\]

for each \( v \in V \).

Given a cocomparability graph \( G = (V,E) \), define \( x(k) = \{ i \mid i \prec k \} \cup \{ i \mid i < k \} \) and \( (i,k) \notin E \), \( 0 \leq k \leq n+1 \), as the set of all predecessors of vertex \( k \). Whether \( i \in x(k) \) can be easily determined by checking whether \( X(i,k) = 1 \). For \( 1 \leq k \leq n+1 \), let \( G_k \) be the subgraph of \( G \) that is induced by the vertex set \( x(k) \setminus \{0\} \). Accordingly, \( G_k = G[x(k) \setminus \{0\}] \). The following theorem can be used to compute the number of
ISs.

**Theorem 1.** For $1 \leq k \leq n+1$, 

$$|\text{IS}(G_k)| = \sum_{i \in x(k)} |\text{IS}(G_i)|; \quad |\text{IS}(G_0)| = 1.$$  

**Proof.** Let $v = \max x(k)$. Clearly, $v < k$ and $(v, k) \notin E$. By Lemma 1, 

$$|\text{IS}(G_k)| = |\text{IS}(G_{k-v})| + |\text{IS}(G_{k-N_G[v]})|.$$  

The present objective is to show $x(k) - N_G[v] = x(v)$. Since $v$ is the largest vertex in $x(k)$, $x(k) - N_G[v] = \{\hat{i} | i < v, (i, k) \notin E \text{ and } (i, v) \notin E\}$. By cocomparability ordering, when $i < v < k$ and $(v, k) \notin E$, $(i, v) \notin E$ implies $(i, k) \notin E$. Thus, $x(k) - N_G[v]$ can be rewritten as $\{\hat{i} | i < v \}$ and $(i, v) \notin E \} = x(v)$. Hence, 

$$|\text{IS}(G_k)| = |\text{IS}(G_{k-v})| + |\text{IS}(G_v)|.$$  

Repeat the above process for the reduced cocomparability graph $G_{k-v}$ until all vertices in $x(k)$ have been eliminated from $G_k$. Thus, Theorem 1 holds. □

Notably, the number of ISs in $G$ can be obtained as $|\text{IS}(G_{n+1})|$ since $G_{n+1} = G$. Based on Theorem 1, a straightforward $O(n^2)$-time algorithm can be derived to compute the number of ISs in a cocomparability graph.

### 3.2 $O(n^3)$-time algorithm for counting MISs ($\omega<2.3727$)

This section presents an algorithm for counting MISs in a cocomparability graph. Before the new algorithm is proposed, some properties of MISs and a theorem are presented here.

Let $\text{MIS}(G_k, i)$ be the set of all MISs in $G_k$ that satisfy the condition for which $i$ is the largest vertex in the MIS. Accordingly, $\text{MIS}(G_k, i) = \{I | I \in \text{MIS}(G_k) \text{ and } \max I = i\}$. The following properties can be trivially established.

**Property 1.** For $i \neq j$, $\text{MIS}(G_k, i) \cap \text{MIS}(G_k, j) = \emptyset$.

**Property 2.** For $1 \leq k \leq n+1$, $\text{MIS}(G_k) = \bigcup_{i \in x(k)} \text{MIS}(G_k, i)$.

Define $y(k) = \{\hat{i} | i < k\}$, $0 \leq k \leq n+1$, as the set of all immediate predecessors of vertex $k$. Whether $i \in y(k)$ can be easily determined by checking whether $y(i, k) = 1$.

**Property 3.** For $i \in x(k)$ and $i \notin y(k)$, $\text{MIS}(G_k, i) = \emptyset$.

**Proof.** $i \in x(k)$ and $i \notin y(k)$ imply that there exists $j \in x(k)$ such that $i < j < k$. Suppose that there exists an MIS $I \in \text{MIS}(G_k, i)$. Clearly, $I \cup \{j\}$ is also an IS in $G_k$, which contradicts the fact that $I$ is an MIS in $G_k$. □

**Property 4.** For $i \in y(k)$, $\text{MIS}(G_k, i) = \{I \cup \{i\} | I \in \text{MIS}(G_k)\}$.

**Proof.** ($\subseteq$) Let $I'$ be an MIS in $G_k$. Since $i \in y(k)$, there exists no $j$ such that $i \sim j \sim k$. Therefore, $I' \cup \{i\}$ is an MIS in $G_k$. 

Huang Sheng-Huang Su and Min-Sheng Lin
(⇒) Let \( I \in \text{MIS}(G_k, i) \) and \( I' = I \setminus \{i\} \). Since \( i \in y(k) \), all \( v \in I' \) and \( I' \) is an IS in \( G_i \). Suppose there exists \( j \neq i \) such that \( I' \cup \{j\} \) is an IS in \( G_i \). Then, \( I' \cup \{j\} \cup \{i\} \) is an IS in \( G_k \), i.e. \( I' \cup \{j\} \) is also an IS in \( G_k \). This contradicts that \( I \) is an MIS in \( G_k \). □

By properties 1, 2, 3, and 4, the following theorem holds.

Theorem 2. For \( 1 \leq k \leq n+1 \),

\[
|\text{MIS}(G_k)| = \sum_{i \in y(k)} |\text{MIS}(G_i)|; \quad |\text{MIS}(G_0)| = 1.
\]

From Theorem 2, a straightforward algorithm can be derived to compute the number of MISs in a cocomparability graph. As mentioned in the Preliminaries section, the Boolean matrix of the relation \( \ll \), \( Y \), can be computed in \( O(n^{2.3727}) \) time. Thus, the algorithm also takes \( O(n^{2.3727}) \) time.

4 Counting independent dominating sets

The section proposes algorithms for counting IDSs and IPDSs in a cocomparability graph.

4.1 \( O(n^9) \)-time algorithm for counting IDSs (\( \omega < 2.3727 \))

Since an IS is an IDS if and only if it is maximal, the number of IDSs in a cocomparability graph can be computed directly using Theorem 2 in \( O(n^{2.3727}) \) time.

4.2 \( O(n^9) \)-time algorithm for counting IPDSs (\( \omega < 2.3727 \))

This section presents a polynomial-time algorithm for counting IPDSs in a cocomparability graph. Define \( N^+[v] = \{u | u \in N[v] \text{ and } u \geq v\} \), \( N^-[v] = \{u | u \in N[v] \text{ and } u \leq v\} \), \( \text{high}(v) = \max N[v] \) and \( \text{low}(v) = \min N[v] \). Notably, \( |N^+[v]| \), \( |N^-[v]| \), \( \text{high}(v) \), and \( \text{low}(v) \), for \( 0 \leq v \leq n+1 \), can be obtained in \( O(n^2) \) time. The following lemma has been proven elsewhere by Chang, Rangan, and Coorg [1].

Lemma 2. [1] \( D = \{0 = v_0 \leq v_1 \leq \ldots \leq v_r \leq v_{r+1} = n+1\} \) is an IPDS of a cocomparability graph if and only if the following three conditions hold for all \( 1 \leq i \leq n+1 \).

1. \( \text{high}(v_{i-1}) < v_i \),
2. \( v_{i-1} < \text{low}(v_i) \), and
3. \( \{u \mid v_{i-1} \leq u \leq v_i\} \) is the disjoint union of \( N^+[v_{i-1}] \) and \( N^-[v_i] \).

Lemma 2 can be rewritten in the following simple form, using only condition (3).
Lemma 3. \( D = \{0 \equiv v_0 < v_1 < v_2 < \ldots < v_r = v_{r+1} \equiv n+1\} \) is an IPDS of a cocomparability graph if and only if \( \{u|v_{i-1} \leq u \leq v_i\} \), for \( 1 \leq i \leq r \), is the disjoint union of \( N^+[v_{i-1}] \) and \( N^-[v_i] \).

Proof. The present objective is to show that condition (3) implies conditions (1) and (2) in Lemma 2. The proof is by contradiction. Suppose \( high(v_{i-1}) \rangle v_i \). Then there exists a vertex \( x \) such that \( x \in N^+[v_{i-1}] \) and \( x \not\in \{u|v_{i-1} \leq u \leq v_i\} \). This contradicts condition (3). Similarly, suppose \( v_{i-1} \rangle \text{low}(v_i) \). There exists a vertex \( x \) such that \( x \in N^-[v_i] \) and \( x \not\in \{u|v_{i-1} \leq u \leq v_i\} \). This also contradicts condition (3). \( \square \)

Since an IPDS is also an MIS, this property provides an easy method of checking where \( \{u|v_{i-1} \leq u \leq v_i\} \) is disjoint union of \( N^+[v_{i-1}] \) and \( N^-[v_i] \) in Lemma 3, as described in the following theorem.

Theorem 3. \( D = \{0 \equiv v_0 < v_1 < v_2 < \ldots < v_r = v_{r+1} \equiv n+1\} \) is an IPDS if and only if \( D \) is an MIS and \( |N^+[v_{i-1}]| + |N^-[v_i]| = v_{i-1} - v_i + 1 \), for \( 1 \leq i \leq r \).

Proof. \((\Rightarrow)\) Let \( D \) be an IPDS. Clearly, \( D \) is an MIS and by Lemma 3, \( \{u|v_{i-1} \leq u \leq v_i\} \), for \( 1 \leq i \leq r \), is the disjoint union of \( N^+[v_{i-1}] \) and \( N^-[v_i] \). Therefore, \( |N^+[v_{i-1}]| + |N^-[v_i]| = v_{i-1} - v_i + 1 \), for \( 1 \leq i \leq r \).

\((\Leftarrow)\) Let \( D \) be an MIS and \( |N^+[v_{i-1}]| + |N^-[v_i]| = v_{i-1} - v_i + 1 \), for \( 1 \leq i \leq r \). Since \( D \) is an MIS, every vertex \( u \) not in \( D \) is adjacent to at least one of the vertices in \( D \). Assume \( v_{i-1} \leq u \leq v_i \) and \( u \) is adjacent to \( v_j \) if \( j > i \) and \( u \) is adjacent to \( v_{i-1} \) if \( j < i \). Thus, \( u \) is adjacent to at least one of \( v_i \) and \( v_{i-1} \). Hence, \( \{u|v_{i-1} \leq u \leq v_i\} \subseteq N^+[v_{i-1}] \cup N^-[v_i] \). Since \( v_{i-1} - v_i + 1 = |N^+[v_{i-1}]| + |N^-[v_i]| \), \( \{u|v_{i-1} \leq u \leq v_i\} \) is the disjoint union of \( N^+[v_{i-1}] \) and \( N^-[v_i] \). By Lemma 3, \( D \) is an IPDS. \( \square \)

Let \( IPDS(G) \) denote the set of all IPDSs in a graph \( G \). Applying Theorem 2 and Theorem 3 yields the following algorithm that takes \( O(n^2 \cdot 3^{\frac{n}{2}}) \) time to compute the number of IPDSs in a cocomparability graph \( G \).

**Algorithm**

Compute the number of IPDSs in a cocomparability graph.

- Compute \( Y_n \); /* the Boolean matrix of the relation \( \ll \)*/
- \( |IPDS(G_0)| \leftarrow 1 \);
- for \( k \leftarrow 1 \) to \( n+1 \) do
  - \( |IPDS(G_k)| \leftarrow 0 \);
  - for \( i \leftarrow 0 \) to \( k-1 \) do
    - if \( (Y(i,k)=1 \text{ and } |N^+[i]|+|N^-[k]|=k-1+1) \) then
      - \( |IPDS(G_k)| \leftarrow |IPDS(G_k)| + |IPDS(G_0)| \);
  - end for
- end for
- output("The number of IPDS is ", \( |IPDS(G_{n+1})| \));

end Algorithm
Acknowledgements. The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 102-2221-E-027-041.

References


[8] V.V. Williams, Breaking the Coppersmith-Winograd barrier. 
http://www.cs.berkeley.edu/virgi/matrixmult.pdf

Received: March 21, 2014