Counting the Number of Minimum / Maximum Weighted Minimal Vertex Covers in a Weighted Trapezoid Graph

Yung-Jui Chen
National Taipei University of Technology
s2319005@ ntut.edu.tw

Min-Sheng Lin
National Taipei University of Technology
mslin@ee.ntut.edu.tw

***Abstract* — This study presents an $O(n^2)$-time algorithm for counting the number of minimum weighted minimal vertex covers and maximum weighted minimal vertex covers in a weighted trapezoid graph simultaneously.**

***Index Terms* — Minimum / Maximum Weighted Minimal Vertex Covers, Weighted Trapezoid Graph, Counting Problem.**

I. INTRODUCTION

Let $G = (V, E)$ be a simple graph, where V and E are the vertex set and edge set of G, respectively. A subset $C \subseteq V$ is called a *vertex cover* of G if and only if every edge in E has at least one endpoint in C. A vertex cover C is called a *minimal vertex cover* if and only if no proper subset of C is a vertex cover. A simple graph $G = (V, E)$ is *weighted* if each $v \in V$ is associated with a weight $w(v)$. The weight of a vertex cover C, denoted $w(C)$, is the sum of the weights of vertices that it contains. A minimal vertex cover with maximum (minimum) weight is called a *maximum* (*minimum*) *weighted minimal vertex cover*.

Okamoto *et al*. [9] developed $O(n + m)$ time algorithms for counting the numbers of independent sets (vertex covers) and maximum independent sets (minimum vertex covers) in a chordal graph, where n is the number of vertices and m is the number of edges. They found that the problems of counting the numbers of maximal independent sets (minimal vertex covers) and minimum maximal independent sets (maximum minimal vertex covers) are #*P*-complete for a chordal graph. No efficient algorithms have yet been developed for solving #*P*-complete problems [10]. Consequently, many researchers have examined on a restricted sub-class of #*P*-complete problems. Lin *et al*. [6, 7] considered a sub-class of the chordal graphs, interval graphs, and obtained efficient algorithms in $O(n)$ time for counting the number of vertex covers, minimal vertex covers, minimum vertex covers, and maximum minimal vertex covers in an interval graph. In addition, they extended these problems in the super-class of both interval graphs and permutation graphs, trapezoid graphs, and proposed $O(n^2)$ time algorithms for solving them [8]. Table 1.1 lists the comparisons of results of papers [6-9].

Various researchers have independently introduced trapezoid graphs (also called Interval-Interval graphs) [1 - 5, 8]. A graph $G = (V, E)$ is a *trapezoid graph* if there exists a set of trapezoids between a pair of horizontal lines such that for each vertex $i \in V$ there exists a corresponding trapezoid i, and an edge $(i, j) \in E$ if and only if trapezoids i and j intersect each other. Such a family of trapezoids between a pair of horizontal lines is referred to as a *trapezoid diagram* for G. A trapezoid i in the trapezoid diagram is denoted by four corner points $a(i)$, $b(i)$, $c(i)$ and $d(i)$, which represent the upper left, the upper right, the lower left and the lower right corner points of trapezoid i, respectively; notably $a(i) < b(i)$ and $c(i) < d(i)$. Figure 1.1 illustrates a trapezoid graph and its corresponding trapezoid diagram.

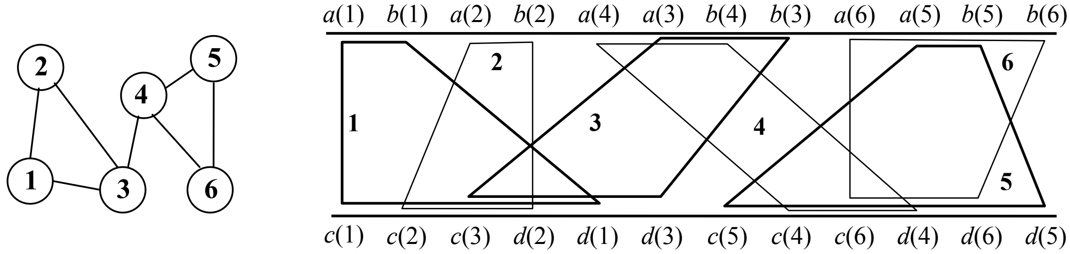

Figure 1.1: Example of a trapezoid graph and its corresponding trapezoid diagram

Table 1.1: Summary of the results

Counting problems \ Graphs	Chordal	Interval	Trapezoid
# of vertex covers	$O(n + m)$ [9]	$O(n)$ [6]	$O(n^2)$ [8]
# of minimal vertex covers	#P-Complete [9]	$O(n)$ [6]	$O(n^2)$ [8]
# of minimum vertex covers	$O(n + m)$ [9]	$O(n)$ [7]	$O(n^2)$ [8]
# of maximum minimal vertex covers	#P-Complete [9]	$O(n)$ [7]	$O(n^2)$ [8]

The rest of this paper is organized as follows. The next section introduces preliminaries on which the rest of the paper depends. Section III presents an $O(n^2)$ time algorithm that simultaneously counts the number of minimum weighted minimal vertex covers and maximum weighted minimal vertex covers.

II. PRELIMINARIES

This section presents the preliminaries on which the desired algorithms depend. Trapezoid graphs generalize both interval graphs and permutation graphs. A graph $G = (V, E)$ is an *interval graph* if there exists a set of intervals on the real line such that for each vertex $i \in V$, a corresponding interval i exists, and an edge $(i, j) \in E$ if and only if intervals i and j intersect each other. A graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$ is a *permutation graph* if there exists a permutation π over $\{1, 2, \ldots, n\}$ such that an edge $(i, j) \in E$ if and only if $(i - j)(\pi^{-1}(i) - \pi^{-1}(j)) < 0$. A permutation graph $G = (V, E)$ can be represented by a permutation diagram consisting of two horizontal lines. The points of the top line are numbered from 1 to n and the points of the bottom line are numbered by a permutation π over $\{1, 2, \ldots, n\}$. Each vertex i in V corresponds to a line i with points i and $\pi^{-1}(i)$ in the diagram, and an edge $(i, j) \in E$ if and only if lines i and j intersect each other in the diagram. A trapezoid graph can be reduced to an interval graph if $a(i) = c(i)$ and $b(i) = d(i)$ for each trapezoid i in its trapezoid diagram. Additionally, a trapezoid graph can be reduced to a permutation graph if $a(i) = b(i)$ and $c(i) = d(i)$ for each trapezoid i in its trapezoid diagram. Thus, both permutation graphs and interval graphs are sub-classes of trapezoid graphs.

Let $T = \{1, 2, \ldots, n\}$ denote the trapezoid diagram of trapezoid graph $G = (V, E)$ with $|V| = n$. Additionally, for each $v \in V$ is associated with a weight $w(v)$. For simplicity, the trapezoid in T that corresponds to vertex i in V is called trapezoid i. The terms trapezoid and vertex are used interchangeably whenever the context is unambiguous. Without loss of generality, the following assumptions are made. First, no two trapezoids share a common endpoint. The points on each horizontal line in T are labeled with distinct integers between 1 and $2n$. Second, two dummy vertices (trapezoids) 0 and n+1 are added to T, where $a(0) = b(0) = c(0) = d(0) = 0$ for vertex 0 and

$a(n+1) = b(n+1) = c(n+1) = d(n+1) = 2n+1$ for vertex $n+1$. Therefore, $T = \{0, 1, 2, \ldots, n, n+1\}$ now. In addition, the weights of these dummy vertices are both 0; i.e., $w(0) = 0$ and $w(n+1) = 0$.

Let i, j be two trapezoids in the trapezoid diagram T. Trapezoid i lies entirely to the left of trapezoid j, denoted by $i \ll j$, if $b(i) < a(j)$ and $d(i) < c(j)$. Clearly, if $i \ll j$ and $j \ll k$, then $i \ll k$. Accordingly, the relation $\ll$ is a partial order over the trapezoid diagram T and $(T, \ll)$ is a strictly partially ordered set. Clearly, two trapezoids i, j intersect each other, expressed by $i \sim j$, if and only if neither $i \ll j$ nor $j \ll i$. Restated, $i \sim j$ if and only if vertices i, j are adjacent to each other. Notice that $\{i : i \ll 0\} = \varnothing$ and $\{i : i \ll n+1\} = \{0, 1, 2, \ldots, n\}$.

A subgraph induced by the set of trapezoids satisfying $\{x : x \ll k\} - \{0\}$, $1 \le k \le n+1$, denoted by G_k. One can easily see that G_k, $1 \le k \le n+1$, is also a trapezoid graph and $G_{n+1} = G$ when $k = n+1$. In the following context, the trapezoids are reordered in topological order according to the strictly partially ordered set $(T, \ll)$. Therefore, for trapezoids i and j, if $i \ll j$ then $i < j$. The work for the topological sorting can be done in $O(n^2)$ time. A trapezoid i is called the rightmost trapezoid of G_k if and only if $i \in \{x : x \ll k\}$ and no trapezoid lies entirely to the right of trapezoid i in the trapezoid diagram of G_k; that is, there exists no trapezoid $j \in \{x : x \ll k\}$ such that $i \ll j \ll k$. Let $y(k)$ denote the set of all rightmost trapezoids of G_k. According to the definition of $y(k)$, $y(k)$ form a clique. Let $MVC(G_k)$ be the set of all minimal vertex covers in G_k. Lin *et al.*[8] designed an $O(n^2)$ algorithm to obtain $y(k)$ and showed that

$$MVC(G_k) = \bigcup_{i \in y(k)} \{C' \cup N_k(i) : C' \in MVC(G_i)\} \qquad (1)$$

, where $N_k(i)$ is the set of vertices adjacent to vertex i in G_k.

Furthermore, they established the following theorem for counting the number of minimal vertex covers in a trapezoid graph.

Theorem 2.1 [8] For $1 \le k \le n+1$,

$$|MVC(G_k)| = \sum_{i \in y(k)} |MVC(G_i)|; \; |MVC(G_0)| = 1.$$

It is obvious that an $O(n^2)$ algorithm can be applied to compute $|MVC(G_k)|$ for $1 \le k \le n+1$. Next section, we deal with problems for counting the number of minimum weighted minimal vertex covers and maximum weighted minimal vertex covers in a weighted trapezoid graph.

III. $O(n^2)$-TIME ALGORITHM TO COUNT THE NUMBER OF MINIMUM / MAXIMUM WEIGHTED MINIMAL VERTEX COVERS

This section presents an $O(n^2)$ algorithm for simultaneously counting the number of minimum vertex covers and maximum minimal vertex covers. Let $W(k)$ be the weight of the set of the trapezoids lying entirely to the left of trapezoid k, i.e., $W(k) = \sum_{i << k} w(i)$ for $1 \le k \le n+1$. Hence, it is easily to get $W(k)$, $1 \le k \le n+1$, in $O(n^2)$ time.

Let $\alpha(k)$ and $\beta(k)$ be the minimum weight of minimal vertex covers in G_k and the maximum weight of those in G_k, respectively.

Lemma 3.1 For $1 \le k \le n+1$,

$$\alpha(k) = \min_{i \in y(k)} \{\alpha(i) + W(k) - W(i) - w(i)\};$$

$$\alpha(0) = 0.$$

Proof. According to the definition of $\alpha(k)$, $\alpha(k) = \min\{\sum_{v \in C} w(v) : C \in MVC(G_k)\}$. Recall that $MVC(G_k) = \bigcup_{i \in y(k)} \{C' \cup N_k(i) : C' \in MVC(G_i)\}$. Hence,

$$\alpha(k) = \min_{i \in y(k)} \left\{ \min\{\sum_{j \in C'} w(j) : C' \in MVC(G_i)\} + \sum_{j \in N_k(i)} w(j) \right\}.$$

Since $y(k)$ is a clique, by the definition of $y(k)$, $N_k(i) = \{x : x \ll k\} - \{x : x \ll i\} - \{i\}$ for $i \in y(k)$. Because $\min\left\{\sum_{j \in C'} w(j) : C' \in MVC(G_i)\right\} = \alpha(i)$ and $\sum_{j \in N_k(i)} w(j) = W(k) - W(i) - w(i)$ for $i \in y(k)$, the lemma follows. □

A similar lemma applies for $\beta(k)$.

Lemma 3.2 For $1 \le k \le n+1$,

$$\beta(k) = \max_{i \in y(k)} \{\beta(i) + W(k) - W(i) - w(i)\};$$

$\beta(0) = 0$.

Based on Lemmas 3.1 and 3.2, the following algorithm can be used to compute $\alpha(k)$ and $\beta(k)$, for $1 \le k \le n+1$, in $O(n^2)$ time.

```
Algorithm Compute_αk_and_βk
  α(0) ← 0;        //initial condition //
  β(0) ← 0;        //initial condition //
  for k ← 1 to n+1 do
    α(k) ← W(k);   β(k) ← 0;
    for each i ∈ y(k) do
      if α(i) + W(k) − W(i) − w(i) < α(k) then
        α(k) ← α(i) + W(k) − W(i) − w(i);
      if β(i) + W(k) − W(i) − w(i) > β(k) then
        β(k) ← β(i) + W(k) − W(i) − w(i);
    end-for
  end-for
end-Algorithm
```

Let $\#\alpha(k)$ and $\#\beta(k)$ be the number of minimum weighted minimal vertex covers in G_k and the number of maximum weighted minimal vertex covers in G_k, respectively.

Theorem 3.1 For $1 \le k \le n+1$,

$$\#\alpha(k) = \sum_{i \in y(k)} \#\alpha(i) \cdot t(k,i), \text{ where}$$

$$t(k,i) = \begin{cases} 1 & \text{if } \alpha(i) + W(k) - W(i) - w(i) = \alpha(k) \\ 0 & \text{otherwise} \end{cases};$$

$\#\alpha(0) = 1$.

Proof. Let $MVC(G_k, i)$, $i \in \{x : x \ll k\}$, be the set of all minimal vertex covers C in G_k such that vertex i is the largest vertex that is not in C; that is, $MVC(G_k, i) = \{C \in MVC(G_k) : i \notin C \text{ and } \{j \in \{x : x \ll k\} : i < j\} \subseteq C\}$. Let $C \in MVC(G_k, i)$ be a minimal vertex cover in G_k and $i \in y(k)$. By the definition of $MVC(G_k, i)$, vertex $i \notin C$, implying $N_k(i) \subseteq C$. Since C is a minimal vertex cover in G_k, $C' = C - N_k(i)$ must be a minimal vertex cover in $G_k - i - N_k(i)$. When $i \in y(k)$, $G_k - i - N_k(i) = G_i$. Therefore, $C = C' \cup N_k(i)$ and $C' \in MVC(G_i)$. Accordingly, for $i \in y(k)$, a one-to-one correspondence clearly exists between $MVC(G_k, i)$ and $MVC(G_i)$; that is $C = C' \cup N_k(i)$ for $C \in MVC(G_k, i)$ and $C' \in MVC(G_i)$. Therefore, the number of all minimal vertex covers with the minimum size $\alpha(k)$ in $MVC(G_k, i)$ equals that with the minimum size $\alpha(i)$ in $MVC(G_i)$. Thus, $\#\alpha(k)$ can be accumulated through $\#\alpha(i)$ by checking whether $\alpha(k)$ equals $\alpha(i) + W(k) - W(i) - w(i)$ for each $i \in y(k)$. □

A similar theorem is valid for $\#\beta(k)$.

Theorem 3.2 For $1 \le k \le n+1$,

$$\#\beta(k) = \sum_{i \in y(k)} \#\beta(i) \cdot t(k,i), \text{ where}$$

$$t(k,i) = \begin{cases} 1 & \text{if } \beta(i) + W(k) - W(i) - w(i) = \beta(k) \\ 0 & \text{otherwise} \end{cases};$$

$\#\beta(0) = 1$.

Based on Theorems 3.1 and 3.2, the following algorithm can be applied to compute $\#\alpha(k)$ and $\#\beta(k)$, for $1 \le k \le n+1$, in $O(n^2)$ time.

```
Algorithm Compute_#αk_and_#βk
  #α(0) ← 1;       //initial condition //
  #β (0) ← 1;      //initial condition //
  for k ← 1 to n+1 do
    for each i ∈ y(k) do
      #α(k) ← 0;
      #β (k) ← 0;
      if α(i) + W(k) − W(i) − w(i) = α(k) then
        #α(k) ← #α(k) + #α(i);
      if β(i) + W(k) − W(i) − w(i) = β(k) then
        #β(k) ← #β(k) + #β(i);
    end-for
  end-for
  output("The number of minimum weighted
          minimal vertex covers is ", #α(n+1));
  output("The number of maximum weighted
          minimal vertex covers is ", #β(n+1));
end-Algorithm
```

IV. ACKNOWLEDGEMENTS

The authors would like to thank to the National Science Council of Taiwan for financial support of this work (NSC-98-2221-E-027-059).

REFERENCE

[1] D. Bera, M. Pal and T.K. Pal, "An efficient algorithm to generate all maximal cliques on trapezoid graphs", Int. J. Comput. Math., Vol. 79, No. 10, pp. 1057-1065, 2002.

[2] D. G. Corneil, P. A. Kamula, "Extensions of permutation and interval graphs", Proceedings of 18th Southeastern Conference on Combinatorics, Graph theory, and Computing, pp. 267-275, 1987.

[3] I. Dagan , M. C. Golumbic , R. Y. Pinter, "Trapezoid graphs and their coloring", Discrete Applied Mathematics, Vol. 21 No.1 , pp.35-46, 1988.

[4] M. Hota, M. Pal, T. K. Pal, "Optimal sequential and parallel algorithms to compute all cut vertices on trapezoid graphs", Computational Optimization and Applications, Vol. 27, No. 1, p.95-113, 2004.

[5] Y. D. Liang, C. L. Lu, C. Y. Tang, "Efficient Domination on Permutation Graphs and Trapezoid Graphs", Proceedings of the Third Annual International Conference on Computing and Combinatorics, August 20-22, pp.232-241, 1997.

[6] M. S. Lin, "Fast and Simple Algorithms to Count the Number of Vertex Covers in an Interval Graph", Information Processing Letters, Vol. 102, No. 4, pp. 143-146, 2007.

[7] M. S. Lin, Y. J. Chen, "Linear time algorithms for counting the number of minimal vertex covers with minimum/maximum size in an interval graph", Information Processing Letters, Vol. 107, No. 6, pp. 257-264, 2008.

[8] M. S. Lin, Y. J. Chen, "Counting the Number of Vertex Covers in a Trapezoid Graph", Information Processing Letters, Vol. 109, Issues 21-22, pp. 1187-1192, 2009.

[9] Y. Okamoto, T. Uno, R. Uehara, "Counting the number of independent sets in chordal graphs", J. Discrete Algorithms, Vol. 6, No. 2, pp. 229-242, 2005.

[10] L. G. Valiant, "The complexity of enumeration and reliability problems", SIAM J. Computing, Vol. 8, pp. 410-421, 1979.