Efficient Algorithm for Minimum Feedback Vertex Set Problem on Trapezoid Graphs

Hirotoshi HONMA[1] Yutaro KITAMURA[2]

Abstract: In an undirected graph, the feedback vertex set (FVS for short) problem is to find a set of vertices of minimum cardinality whose removal makes the graph acyclic. The FVS has applications to several areas such that combinatorial circuit design, synchronous systems, computer systems, VLSI circuits and so on. The FVS problem is known to be NP-hard on general graphs but interesting polynomial solutions have been found for some special classes of graphs. In this paper, we present an $O(n^{2.68}+\gamma n)$ time algorithm for solving the FVS problem on trapezoid graphs, where γ is the total number of factors included in all maximal cliques.

Key words: Design and analysis of algorithms, Feedback vertex set, Trapezoid graphs, NP-hard

1 Introduction

Let $G=(V,E)$ be a simple graph where V is the set of vertices and E is the set of edges of G with $|V|=n$ and $|E|=m$. Suppose that V' is a nonempty subset of V. The subgraph of G, whose vertex set is V' and whose edge set is the set of those edges of G that have both vertices in V', is called the subgraph of G induced by V' and is denoted by $G[V']$. A cycle with no repeated vertices is a simple cycle. In this paper, a term "cycle" denotes "simple cycle". The *feedback vertex set* (FVS for short) consists of a subset $F \subseteq V$ such that each cycle in G contains at least one vertex in F. In other words, a subset $F \subset V$ is an FVS of G if the subgraph induced by $G[V-F]$ is acyclic. The FVS problem is to find an FVS of minimum cardinality in G. The FVS problem has applications in several areas such as deadlock prevention in operating systems [19], combinatorial circuit design [12], VLSI circuits [11], and information security [10].

The FVS problem is known to be NP-hard on general graphs [9] and bipartite graphs [21]. However, interesting polynomial-time solutions have been found for special classes of graphs, such as interval graphs [16], permutation graphs [15], butterfly networks [17], hypercubes [8], star graphs [20], diamond graphs [4], and rotator graphs [14].

Trapezoid graphs were first introduced by Dagan et al. [6]. They showed that trapezoid graphs can be used to model a channel routing problem in a single-layer-per-net model and proposed an $O(n^2)$ algorithm for the chromatic number and a less efficient algorithm for the maximum clique on trapezoid graphs. In recent years, many studies have focused on the trapezoid graphs [6, 13, 3, 18]. Thus, trapezoid graphs have been studied extensively from both the theoretical and algorithmic point of view.

It has been shown in [5] that the class of trapezoid graphs is properly contains both the class of interval graphs and that of permutation graphs. Both algorithm for the FVS problem on interval graphs [16] and that on permutation graphs [15] employed a dynamic programming scheme. In contrast, our algorithm finds the minimum FVS by using all maximal cliques and chordless cycles of length 4 in trapezoid graphs. It takes $O(n^{2.68}+\gamma n)$ time to solve the FVS problem associated with trapezoid graphs. Here, γ is the total number of factors included in all maximal cliques.

The remainder of this paper is organized as follows. We first describe, in Section 2, some definitions and notations used throughout this paper. Section 3 deals with several lemmas upon which our algorithm is based. We show an algorithm for the FVS problem and analyze its complexity in Section 4. Finally, Section 5 concludes this paper.

2 Definitions and Notations

There are two horizontal lines, denoted by L_1 and L_2. A *trapezoid model* M consists of some trapezoids with two corner points $a_i < b_i$ lying on L_1 and the other corner points $c_i < d_i$ lying on L_2. A graph $G=(V,E)$ is called a *trapezoid graph* if it can be represented by M such that each trapezoid T_i corresponds to a vertex in V and $(i,j) \in E$ if and only if T_i and T_j intersect in M [6]. Figure 1 shows a trapezoid model M consisting of 17 trapezoids. Figure 2 shows the trapezoid graph G corresponding to M shown in Fig. 1. In this paper,

[1]Department of Information Engineering, Kushiro National College of Technology

[2]Department of Intelligent Interaction Technologies, University of Tsukuba

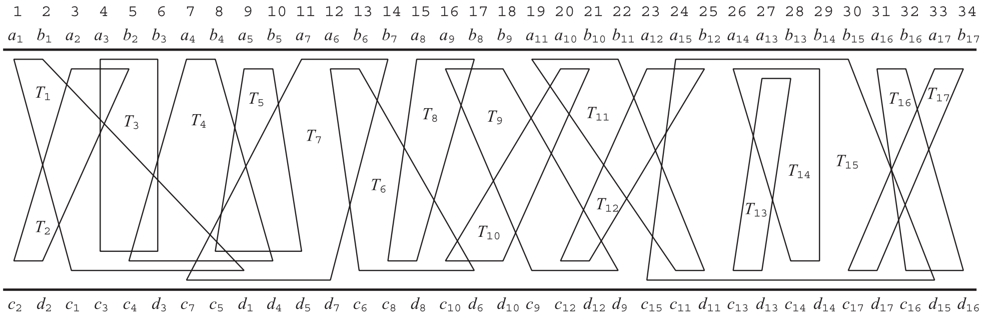

Figure 1: Trapezoid model M.

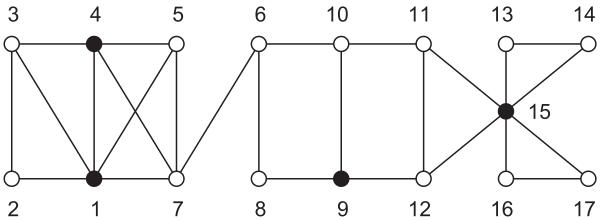

Figure 2: Trapezoid graph G.

we assume that the trapezoid graph is connected, and the corner points a_i, b_i, c_i, d_i, $1 \le i \le n$ sorted by b_i are given. The class of trapezoid graphs includes two well-known classes of intersection graphs: the class of permutation graphs and that of interval graphs. The former is obtained by setting $a_i = b_i$ and $c_i = d_i$ for all i and the latter is obtained by setting $a_i = c_i$ and $b_i = d_i$ for all i.

A *maximal clique* is a clique to which no further vertex of the graph can be added so that it remains a clique. All maximal cliques of G shown in Fig. 2, the vertices of which are arranged in ascending order, are $MC_1 = \{1,2,3\}$, $MC_2 = \{1,3,4\}$, $MC_3 = \{1,4,5,7\}$, $MC_4 = \{6,7\}$, $MC_5 = \{6,8\}$, $MC_6 = \{6,10\}$, $MC_7 = \{8,9\}$, $MC_8 = \{9,10\}$, $MC_9 = \{9,12\}$, $MC_{10} = \{10,11\}$, $MC_{11} = \{11,12,15\}$, $MC_{12} = \{13,14,15\}$, and $MC_{13} = \{15,16,17\}$, where each maximal clique is numbered in lexicographic order. Using Bera et al.'s algorithm [3], all maximal cliques can be generated for a trapezoid graph G. Let N_j be the cardinality of MC_j, for example, $N_1 = 3$, $N_2 = 3$, $N_3 = 4$, $N_4 = 2$, $N_5 = 2$, $N_6 = 2$, $N_7 = 2$, $N_8 = 2$, $N_9 = 2$, $N_{10} = 2$, $N_{11} = 3$, $N_{12} = 3$, and $N_{13} = 3$.

Throughout this paper, we use the term *triangle* to denote a cycle of length 3. A cycle that contains no chord is called a *chordless cycle*. A *square* is defined as a chordless cycle of length 4, and is denoted by S_j. For instance, the trapezoid graph G shown in Fig. 2 has two squares, $S_1 = \{6,8,9,10\}$ and $S_2 = \{9,10,11,12\}$, respectively.

For each vertex $i \in V$, $\sigma(i)$ is the total number of maximal cliques ($N_j \ge 3$) and squares containing i. That is, $\sigma(i) = |\{MC_j \mid i \in MC_j, N_j \ge 3, j = 1,\ldots,p\}| + |\{S_j \mid i \in S_j, j = 1,\ldots,q\}|$ when a trapezoid graph G consists of p maximal cliques and q squares. For the trapezoid graph shown in Fig. 2, $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 2$, $\sigma(4) = 2$, $\sigma(5) = 1$, $\sigma(6) = 1$, $\sigma(7) = 1$, $\sigma(8) = 1$, $\sigma(9) = 2$, $\sigma(10) = 2$, $\sigma(11) = 2$, $\sigma(12) = 2$, $\sigma(13) = 1$, $\sigma(14) = 1$, $\sigma(15) = 3$, $\sigma(16) = 1$, and $\sigma(17) = 1$. For the sake of convenience, we denote the σ value sequence of G by $\sigma = [3,1,2,2,1,1,1,1,2,2,2,2,1,1,3,1,1]$.

3 Our Approach

We show some lemmas that are useful for constructing an efficient algorithm to solve the FVS problem on a trapezoid graph G. The following Lemma 1 provides an important property of a trapezoid graph.

Lemma 1 ([2]) *Let G be a trapezoid graph. Then, the cycles contained in G are only triangles or squares.*

Lemma 2 was established by Bera et al. [3], who presented an algorithm to find all maximal cliques of a trapezoid graph.

Lemma 2 ([3]) *Let G be a trapezoid graph with n vertices. All maximal cliques of G can be computed in $O(n^2 + \gamma n)$ time, where γ is the total number of factors included in all maximal cliques.*

Lemma 3 was established by Alon et al. [1], who developed an efficient algorithm to find all squares on a simple graph.

Lemma 3 ([1]) *Let G be a simple graph with n vertices. All squares contained in G are found in $O(n^{2.68})$ time.*

Lemma 4 *Let $G = (V,E)$ be a trapezoid graph. Moreover, let MC_j and S_j be the maximal cliques and squares on G, respectively. Initially, F is an empty set. After executing the following steps (1) and (2), F is an FVS of G.*

(1) For all MC_j, $N_j \geq 3$, include all vertices except any two of MC_j in F.

(2) For all S_j, include one vertex of S_j in F.

Proof: By Lemma 1, any cycle contained in the trapezoid graph is either a triangle or square. Note that each triangle is a subset of any maximal clique. A graph obtained by deleting all vertices except any two from a maximal clique has no cycle. For example, removing any $m-2$ vertices from a maximal clique of cardinality m makes the graph acyclic. Similarly, if one vertex is removed from a square, the graph has no cycle. Thus, no cycle exists in $G[V-F]$ after executing steps (1) and (2), implying that F is an FVS of G. □

We show an example of the construction of an FVS on the trapezoid graph shown in Fig. 2 by executing steps (1) and (2) of Lemma 4. First, we compute all maximum cliques and squares of G by employing Bera et al. and Alon et al.'s algorithms, respectively. Then, 13 maximal cliques and 2 squares are obtained. Next, step (1) of Lemma 4 is carried out for each maximal clique MC_j. For instance, we choose vertex 2 for $MC_1 = \{1,2,3\}$ and include it into F. In this manner, vertex 3 for MC_2, vertices 4 and 5 for MC_3, vertex 11 for MC_{11}, vertex 13 for MC_{12}, and vertex 17 for MC_{13} are iteratively selected and included into F. In what follows, step (2) of Lemma 4 is executed for each square S_j. For example, we choose vertices 8 and 12 for $S_1 = \{6,8,9,10\}$ and $S_2 = \{9,10,11,12\}$, respectively, and include them into F. After executing these processes on G, we obtain the result $F = \{2,3,4,5,8,11,12,13,17\}$; this is an FVS of G by Lemma 4. Unfortunately, F constructed in this manner is not necessarily the FVS of minimum cardinality. As shown in Fig. 2, a set $\{1,4,9,15\}$ is the minimum FVS on G.

Now, we define Q_j in order to compute the minimum FVS for a trapezoid graph containing p maximal cliques and q squares. Q_j is numbered in the ascending lexicographic order for all maximal cliques MC_j, $N_j \geq 3$, $j = 1,2,\ldots,p$, and squares S_j, $j = 1,2,\ldots,q$, of G. For a trapezoid graph of Fig. 2, we have $Q_1 = MC_1 = \{1,2,3\}$, $Q_2 = MC_2 = \{1,3,4\}$, $Q_3 = MC_3 = \{1,4,5,7\}$, $Q_4 = S_1 = \{6,8,9,10\}$, $Q_5 = S_2 = \{9,10,11,12\}$, $Q_6 = MC_{11} = \{11,12,15\}$, $Q_7 = MC_{12} = \{13,14,15\}$, and $Q_8 = MC_{13} = \{15,16,17\}$. Note that Q_1, Q_2, Q_3, Q_6, Q_7, and Q_8 are maximal cliques, and Q_4 and Q_5 are squares. Moreover, we can easily see that $\sigma(i) = |\{Q_j \mid i \in Q_j \text{ for all } Q_j\}|$ for $i \in V$.

The following lemma is given for obtaining the minimum FVS on trapezoid graph G. It is based on our algorithm.

Lemma 5 *Let $G = (V,E)$ be a trapezoid graph that contains Q_j, where $j = 1,2,\ldots,r$. Then, F obtained after executing the following Process A is the minimum FVS of G. Initially, F is an empty set.*

// Process A //
For all Q_j, $j = 1,2,\ldots,r$,

(a) If $|Q_j - F| \geq 3$ and Q_j is a "maximal clique,"

(a.1) If there exist two or more xs that minimize $\sigma(x)$ for $x \in Q_j - F$, then let x_1 and x_2 be any two of them. Then, include all vertices of $Q_j - F$ except x_1 and x_2 in F.

(a.2) If there exists exactly one x that minimizes $\sigma(x)$ for $x \in Q_j - F$, then let it be x_1 and let x_2 be any x that attains the second-minimum value of $\sigma(x)$. Then, include all vertices of $Q_j - F$ except x_1 and x_2 in F.

(b) If $|Q_j - F| = 4$ and Q_j is a "square," include any vertex x of Q_j in F, such that $\sigma(x)$ is the maximum of $x \in Q_j$.

(c) $\sigma(i) := \sigma(i) - 1$ for all vertices $i \in Q_j$.

Proof: As mentioned in Lemma 4, F is an FVS since $G[V-F]$ contains no cycle after executing Process A. The following arguments show that F obtained by Process A is a minimum cardinality FVS. Suppose that we have Q_j, where $j = 1,2,\ldots,r$. We denote F obtained after executing the jth iteration of Process A as F_j. For each vertex $i \in V$, $\sigma(i)$ is the total number of maximal cliques and squares containing i.

After executing the 1st iteration, $G[Q_1 - F_1]$ has no cycle and F_1 is clearly the minimum FVS for the subgraph $G[Q_1]$. In the 2nd iteration, two cases must be considered.

Case 1: Q_2 is a maximal clique. In this case, step (a) is executed. If $|Q_2 - F_1| < 3$, no vertex is included in F_2. This implies that the elimination of the vertices in F_1 obtained in the previous iteration breaks all triangles of Q_2. If $|Q_2 - F_1| \geq 3$ and there exist two or more xs that minimize $\sigma(x)$ for $x \in Q_2 - F_1$ (step (a.1)), include all vertices of $Q_2 - F_1$ except two vertices x_1 and x_2 in F_2 where $\sigma(x_1)$ and $\sigma(x_2)$ are the two minima of $x \in Q_2 - F_1$. Moreover, if $|Q_2 - F_1| \geq 3$ and there exists exactly one x that minimizes $\sigma(x)$ for $x \in Q_2 - F_1$ (step (a.2)), let it be x_1 and let x_2 be any x that attains the second-minimum value of $\sigma(x)$. Then, include all vertices of $Q_2 - F_1$ except x_1 and x_2 in F_2. It is obvious that the cardinality of F can be reduced by including those vertices that appear in many maximal cliques or squares in F. After executing step (a), $G[Q_1 \cup Q_2 - F_2]$ has no cycle and F_2 is the minimum FVS of the subgraph $G[Q_1 \cup Q_2]$.

Case 2: Q_2 is a square. In this case, step (b) is executed. If $|Q_2 - F_1| \neq 4$, no vertex is included in

F_2. This implies that the elimination of the vertices in F_1 breaks the square of Q_2. If $|Q_2-F_1|=4$, include a vertex x of Q_2 in F_2 such that $\sigma(x)$ is the maximum of $x \in Q_2$. After executing step (b), $G[Q_1 \cup Q_2 - F_2]$ has no cycle and F_2 is the minimum FVS of the subgraph $G[Q_1 \cup Q_2]$.

Hence, after executing the 2nd iteration, F_2 is the minimum FVS of the subgraph $G[Q_1 \cup Q_2]$. Similarly, in the 3rd iteration, $G[Q_1 \cup Q_2 \cup Q_3 - F_3]$ contains no cycle and F_3 is the minimum FVS of the subgraph $G[Q_1 \cup Q_2 \cup Q_3]$. Using a similar argument, after applying Lemma 5, $G[Q_1 \cup Q_2 \cup \cdots \cup Q_r - F_r]$ has no cycle and F_r is the minimum FVS of the subgraph $G[Q_1 \cup Q_2 \cup \cdots \cup Q_r]$. Hence, the Process A constructs the minimum FVS F of the trapezoid graph G. □

In step (a) of Process A, if there exist three or more vertices corresponding to x_1 and x_2, we can choose any two of these vertices. Furthermore, in step (b) of Process A, if there exist two or more vertices corresponding to x, we can choose any one of these vertices. It is clear that the FVS constructed according to Process A depends on the chosen vertices. However, the cardinality of each FVS is minimum by Process A. In general, it is not necessary that graph G has only one minimum FVS. For example, $\{1,5,9,15\}$, $\{1,4,10,15\}$, and $\{1,7,10,15\}$ are all minimum FVSs of G, as shown in Fig. 2.

4 Algorithm and its Complexity

We present an algorithm FVS for constructing the minimum FVS on a trapezoid graph. The algorithm FVS is based on the result of Lemma 5 presented in the previous section. We use the trapezoid graph G shown in Fig. 2 as an example to illustrate this algorithm. The major steps of the proposed algorithm FVS to construct a minimum FVS of G are given below.

First, we compute all maximal cliques MC_j for the trapezoid graph G and its corresponding trapezoid model M and then arrange the vertices of MC_j in ascending order in Step 1. As a result, $MC_1=\{1,2,3\}$, $MC_2=\{1,3,4\}$, $MC_3=\{1,4,5,7\}$, $MC_4=\{6,7\}$, $MC_5=\{6,8\}$, $MC_6=\{6,10\}$, $MC_7=\{8,9\}$, $MC_8=\{9,10\}$, $MC_9=\{9,12\}$, $MC_{10}=\{10,11\}$, $MC_{11}=\{11,12,15\}$, $MC_{12}=\{13,14,15\}$, and $MC_{13}=\{15,16,17\}$ are constructed. In Step 2, all squares S_j for G are computed. Then, two squares, $S_1=\{6,8,9,10\}$ and $S_2=\{9,10,11,12\}$, are obtained in this step. In Step 3, all Q_j, $j=1,2,\ldots,r$, are obtained. We have $Q_1=MC_1=\{1,2,3\}$, $Q_2=MC_2=\{1,3,4\}$, $Q_3=MC_3=\{1,4,5,7\}$, $Q_4=S_1=\{6,8,9,10\}$, $Q_5=S_2=\{9,10,11,12\}$, $Q_6=MC_{11}=\{11,12,15\}$, $Q_7=MC_{12}=\{13,14,15\}$, and $Q_8=MC_{13}=\{15,16,17\}$. In Step 4, all σ values $\sigma=[3,1,2,2,1,1,1,1,2,2,2,2,1,1,3,1,1]$ are obtained. The details of the processes carried out in Step 5 is described below.

Algorithm 1: Algorithm FVS

Input: The corner points a_i, b_i, c_i, d_i of each trapezoid $T_i, i=1,2,\ldots,n$.
Output: The minimum FVS F.

```
(Step 1)
Compute all maximal cliques MC_j, j = 1,2,...,p.
Let p be the total number of maximal cliques. ;
(Step 2)
Compute all squares S_j, j = 1,2,...,q.
Let q be the total number of squares. ;
(Step 3)
Compute all Q_j for j = 1,2,...,r. ;
(Step 4)
Compute all σ(i) for i = 1,2,...,n. ;
(Step 5) /* Process A of Lemma 5. */
F := ∅ ;
for All Q_j, j = 1,2,...,r do
    /* Step (a) */
    if Q_j is a maximal clique ∧ |Q_j − F| ≥ 3 then
        if there exist two or more xs that minimize
        σ(x) for x ∈ Q_j − F then
            Let x_1 and x_2 be any two of them.
        end
        if there exists exactly one x that minimizes
        σ(x) for x ∈ Q_j − F then
            Let it be x_1.
            Let x_2 be any x that attains the
            second-minimum value of σ(x_2).
        end
        include all vertices of Q_j − F except x_1 and
        x_2 in F.
    end
    /* Step (b) */
    if Q_j is a square ∧ |Q_j − F| = 4 then
        F := F ∪ {x}, where x is one of the vertices
        such that σ(x) is the maximum values of
        x ∈ Q_j ;
    end
    /* Step (c) */
    for x ∈ Q_j do σ(x) := σ(x) − 1 ;
end
```

(Step 5)

Initially, $F=\emptyset$,
$\sigma=[3,1,2,2,1,1,1,1,2,2,2,2,1,1,3,1,1]$.

1st iteration

$Q_1=\{1,2,3\}$, $F=\{\underline{1}\}$,
$\sigma=[\underline{2},\underline{0},\underline{1},2,1,1,1,1,2,2,2,2,1,1,3,1,1]$.

2nd iteration

$Q_2 = \{1, 3, 4\}$, $F = \{1\}$,
$\sigma = [\underline{1}, 0, \underline{0}, \underline{1}, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 3, 1, 1]$.

3rd iteration
$Q_3 = \{1, 4, 5, 7\}$, $F = \{1, \underline{4}\}$,
$\sigma = [\underline{0}, 0, 0, \underline{0}, \underline{0}, 1, \underline{0}, 1, 2, 2, 2, 2, 1, 1, 3, 1, 1]$.

4th iteration
$Q_4 = \{6, 8, 9, 10\}$, $F = \{1, 4, \underline{9}\}$,
$\sigma = [0, 0, 0, 0, 0, \underline{0}, 0, \underline{0}, \underline{1}, \underline{1}, 2, 2, 1, 1, 3, 1, 1]$.

5th iteration
$Q_5 = \{9, 10, 11, 12\}$, $F = \{1, 4, 9\}$,
$\sigma = [0, 0, 0, 0, 0, 0, 0, 0, \underline{0}, \underline{0}, \underline{1}, \underline{1}, 1, 1, 3, 1, 1]$.

6th iteration
$Q_6 = \{11, 12, 15\}$, $F = \{1, 4, 9, \underline{15}\}$,
$\sigma = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \underline{0}, \underline{0}, 1, 1, \underline{2}, 1, 1]$.

7th iteration
$Q_7 = \{13, 14, 15\}$, $F = \{1, 4, 9, 15\}$,
$\sigma = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \underline{0}, \underline{0}, \underline{1}, 1, 1]$.

8th iteration
$Q_8 = \{15, 16, 17\}$, $F = \{1, 4, 9, 15\}$,
$\sigma = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \underline{0}, \underline{0}, \underline{0}]$.

After executing Step 5, $G[V - F]$ contains no triangle and no square. The algorithm FVS gives $F = \{1, 4, 9, 15\}$; this is the minimum FVS of G.

The algorithm FVS finds the minimum FVS of a trapezoid graph G. We analyze the complexity of this algorithm. In Step 1, all maximal cliques of G are computed by using Bera et al.'s algorithm [3]. This step takes $O(n^2 + \gamma n)$ time. In Step 2, all chordless cycles of length 4 in G are computed. This step can be executed in $O(n^{2.68})$ time by employing Alon et al.'s algorithm [1]. In Step 3, all Q_j, $j = 1, 2, \ldots, r$, are computed. In Step 4, $\sigma(i)$ are computed for all vertices $i \in V$. The complexities of Steps 3 and 4 depend on the number of maximal cliques obtained in Step 1. Thus, these steps can be executed in $O(\gamma n)$ time. In Step 5, the minimum FVS F is constructed. This step can be executed in $O(n^2 + \gamma n)$ time. Hence, we have the following theorem.

Theorem 1 *Given a trapezoid graph G, the algorithm FVS finds the minimum FVS of G in $O(n^{2.68} + \gamma n)$ time, where γ is the total number of factors included in all maximal cliques.*

5 Concluding Remarks

In this paper, we proposed an algorithm that runs in $O(n^{2.68} + \gamma n)$ time to find the minimum FVS on a trapezoid graph. Our algorithm employ the algorithms for finding the maximal cliques and squares by Bera et al. [3] and Alon et al. [1]. The method can be understood intuitively. The complexity of our algorithm depend on the number of all maximal cliques of a trapezoid graph. Bera et al.'s algorithm [3] can find all maximal cliques efficiently when G does not contain a large number of maximal cliques. Then, our algorithm is useful when a given trapezoid graph is edge dense and does not have a large number of maximal cliques. Reducing the complexity of the algorithm and extending the results to other graphs are issues left for future research.

References

[1] N. Alon, R. Yuster, and U. Zwick, "Finding and counting given length cycles," Algorithmica, vol.17, no.3, pp.209–223, 1997.

[2] S.C. Barman, S. Mondal, and M. Pal, "An efficient algorithm to find next-to-shortest path on trapezoid graphs," Advances in Applied Mathematical Analysis, vol.2, no.2, pp.97–107, 2007.

[3] D. Bera, M. Pal, and T.K. Pal, "An efficient algorithm to generate all maximal cliques on trapezoid graphs," Intern. J. Computer Mathematics, vol.79, no.10, pp.1057–1065, 2002.

[4] F. Carrabs, R. Cerulli, M. Gentili, and G. Parlato, "A linear time algorithm for the minimum weighted feedback vertex set on diamonds," Inf. Process. Lett., vol.94, no.1, pp.29–35, 2005.

[5] D.G. Corneil and P.A. Kamula, "Extensions of permutation and interval graphs," Congr. Numer., vol.58, pp.267–275, 1987.

[6] I. Dagan, M.C. Golumbic, and R.Y. Pinter, "Trapezoid graphs and their coloring," Discrete Appl. Math., vol.21, no.1, pp.35–46, 1988.

[7] S. Felsner, R. Müller, and L. Wernisch, "Trapezoid graphs and generalization, geometry and algorithms," Discrete Appl. Math., vol.74, no.1, pp.13–32, 1997.

[8] R. Focardi, F.L. Luccio, and D. Peleg, "Feedback vertex set in hypercubes," Inf. Process. Lett., vol.76, no.1-2, pp.1–5, 2000.

[9] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to Theory of NP-completeness, W. H. Freeman, San Francisco, 1979.

[10] D. Gusfield, "A graph theoretic approach to statistical data security," SIAM J. Comput., vol.17, no.3, pp.552–571, 1998.

[11] A.V. Hudli and R.V. Hudli, "Finding small feedback vertex sets for VLSI circuits," Microprocessors and Microsystems, vol.18, no.7, pp.393–400, 1994.

[12] D.B. Johnson, "Finding all the elementary circuits of a directed graph," SIAM J. Comput., vol.4, no.1, pp.77–84, 1975.

[13] E. Köhler, "Connected domination and dominating clique in trapezoid graphs," Discrete Appli. Math., vol.99, pp.91–110, 2000.

[14] C.J. Kuo, C.C. Hsu, H.R. Lin, and K.K. Lin, "An efficient algorithm for minimum feedback vertex sets in rotator graphs," Inf. Process. Lett., vol.109, no.9, pp.450–453, 2009.

[15] Y.D. Liang, "On the feedback vertex set problem in permutation graphs," Inf. Process. Lett., vol.52, no.3, pp.123–129, 1994.

[16] C.L. Lu and C.Y. Tang, "A linear-time algorithm for the weighted feedback vertex problem on interval graphs," Inf. Process. Lett., vol.61, no.2, pp.107–111, 1997.

[17] F.L. Luccio, "Almost exact minimum feedback vertex set in meshes and butterflies," Inf. Process. Lett., vol.66, no.2, pp.59–64, 1998.

[18] T.H. Ma and J.P. Spinrad, "An O(n^2) time algorithm for the 2-chain cover problem and related problems," Proc. 2nd ACM-SIAM symposium on Discrete algorithms, San Francisco, 1991.

[19] A. Silberschatz, P.B. Galvin, and G. Gagne, Operating Systems Concepts, John Wiley and Sons, New York, 2003.

[20] F.H. Wang, Y.L. Wang, and J.M. Chang, "Feedback vertex sets in star graphs," Inf. Process. Lett., vol.89, no.4, pp.203–208, 2004.

[21] M. Yannakakis, "Node-deletion problem on bipartite graphs," SIAM J. Comput., vol.10, no.2, pp.310–327, 1981.