Angewandte Mathematik und
Informatik
Universität zu Köln

Report No. 97-287

On computing all minimal solutions for feedback problems

by

B. Schwikowski, E. Speckenmeyer

1997

B. Schwikowski
DKFZ, Abt. Theoretische Bioinformatik, INF 280,
D-69120 Heidelberg
GMD-SCAI, D-53757 St. Augustin
E-mail: `schwikowski@dkfz-heidelberg.de`

E. Speckenmeyer
Universität Köln
Institut für Informatik
Pohligstr. 1
D-50969 Köln

1991 Mathematics Subject Classification: 68Q25,05C85,05C38,05C20,68Q35,05A15
Keywords: feedback problem;directed graph;exact enumeration problem

On computing all minimal solutions for feedback problems

Benno Schwikowski*

DKFZ–TBI, INF 280, D–69120 Heidelberg, and
GMD-SCAI, D–53757 St. Augustin

and

Ewald Speckenmeyer

Inst. f. Informatik, Pohligstr. 1, D–50969 Köln

Abstract

We present an algorithm that generates all (inclusion-wise) minimal feedback vertex sets of a directed graph $G = (V, E)$. The feedback vertex sets of G are generated with a polynomial delay of $\mathcal{O}(|V|^2(|V| + |E|))$. Variants of the algorithm generate all minimal solutions for the undirected case and the directed feedback arc set problem, both with a polynomial delay of $\mathcal{O}(|V|\,|E|\,(|V| + |E|))$.

1 Introduction

Generating all admissible configurations is a well-examined problem for many combinatorial problems. Typically, solutions are subsets of a finite set, and the set of solutions is *monotone*, i.e. the supersets of admissible solutions are also admissible. This is the case for the feedback problems that we examine, and thus the enumeration of all minimal admissible solutions provides a generic nonredundant description of the solution space.

For a directed graph, a *feedback vertex set* is a subset of its vertices that contains at least one vertex of any directed cycle. Finding all minimal feedback vertex sets is computationally demanding, since finding a feedback vertex set with minimum cardinality is NP-hard [4], and the output itself can have exponential size.

Related problems. Minimal feedback vertex sets are intimately related to the extremal solutions of other combinatorial optimization problems. A set F of vertices in an undirected graph $G = (V, E)$ is a vertex cover *iff* F is a feedback vertex set in the directed graph G' that has two directed arcs for each undirected arc in G. Thus, finding feedback vertex sets is a generalization to the problem of finding vertex covers. Further, if and only if

*Corresponding author. E-mail: schwikowski@dkfz-heidelberg.de

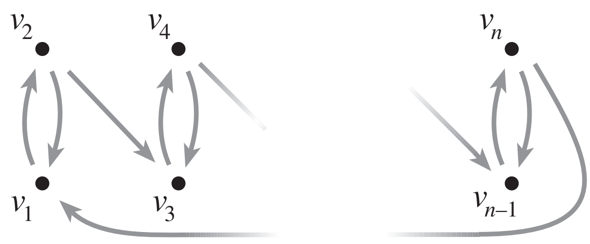

Figure 1: This graph has $2^{n/2}$ different minimal feedback vertex sets

F is a vertex cover, $V-F$ is an independent set, and only in this case, $V-F$ is a clique in the complement graph of G. Due to these close relationships, finding minimal feedback vertex sets can be regarded a generalization to the problem of finding minimal vertex covers, maximal independent sets, or maximal cliques in a graph.

Previous results for maximal independent sets and maximal cliques. Several authors [5, 3, 8] have stated algorithms that compute all maximal independent sets of a given graph. Some algorithms only need *polynomial delay*, where the time between the output of two successive configurations is bounded by a polynomial in the input size. Algorithms for generating all (maximal) cliques are surveyed in [6] and [7].

Our results. Our enumeration procedure of generating all minimal feedback vertex sets for a directed graph $G = (V, E)$ relies on an exhaustive search in a superstructure graph Φ, whose vertices represent the minimal feedback vertex sets of G. The vertex v_F of Φ representing the minimal feedback vertex set F of G is connected by an arc to the vertex $v_{F''}$, of Φ, representing the minimal FVS F'' of G, if F'' can be obtained by a local operation from F as follows. Delete a vertex v from F and add all vertices w to F reachable from F via an arc (v, w). Denote the feedback vertex set obtained in this way by F', which is not necessarily a minimal feedback vertex set. Determine a minimal feedback vertex set $F'' \subseteq F'$ in an arbitrarily but fixed way. Note that the superstructure graph Φ defined in this way has exactly one successor vertex F'' for every minimal feedback vertex set F and every $v \in F$.

We will show in section 2 that Φ is strongly connected and has a diameter of at most $|V|$. Applying exhaustive search to Φ then yields our main result, that all minimal feedback vertex sets of a directed graph G can be determined in time $\mathcal{O}\big(|V|^2(|V| + |E|)\big)$ for each vertex in Φ, see Theorem 1. In the sections 3 and 4 we apply this technique for enumerating all minimal feedback vertex sets of undirected graphs and for enumerating all minimal feedback arc sets of directed graphs. Since previous approaches for approximating these problems are completely different from each other [2], it is remarkable that we can apply the same technique to all three enumeration problems.

2 Feedback vertex sets of directed graphs

A *feedback vertex set (FVS)* of a directed graph $G = (V, E)$ is a set $F \subseteq V$ where $C \cap F \neq \emptyset$ for any directed cycle C of G. F is a *minimal feedback vertex set (MFVS)* if there is no feedback vertex set $F' \neq F$, $F' \subseteq F$. Our algorithm exploits a simple relation between MFVSs that allows for generating all MFVSs by local modification.

Let F be a MFVS of G, $v \in F$. By $N^+(v)$ we denote the set of vertices $v' \in V$ with $(v, v') \in E$. The FVS $F' = (F - v) \cup N^+(v)$ contains at least one MFVS F'' as a subset. We call each MFVS $F'' \subseteq F - v \cup N^+(v)$ a *(v-)successor* of F.

For any MFVS F and $v \in F$ there can be an exponential number of v-successors F''. This can be seen by adding to the graph G_n in Figure 1 the arcs $(v_1, v_2), (v_1, v_3), \ldots, (v_1, v_n)$. Observe that $F = \{v_1, v_3, \ldots, v_{n-1}\}$ is a MFVS of the resulting graph G'_n. Further, each set F'' of vertices that contains v_1, v_n and exactly one vertex of each set $\{v_{2i-1}, v_{2i}\}$, $i = 2, \ldots, n/2 - 1$, is a v_1-successor of F. Hence there are $2^{\frac{n}{2}-2}$ different v_1-successors of F in G_n.

For our purpose we will just need one v-successor of a MFVS F that can be chosen arbitrarily. We assume that a *successor function* $\mu_G : 2^V \times V \longrightarrow 2^V$ assigns some fixed v-successor F''_0 of F to any such pair (F, v). We also call $F''_0 = \mu_G(F, v)$ a *μ_G-successor* of F.

Transforming MFVSs. We now present an algorithm that, given two arbitrary MFVSs, F and F^*, transforms F into F^* by generating μ_G-successors.

Algorithm TRANSFORM-DIRECTED-MFVS $(G = (V, E), F, F^*, \mu_G)$

1 compute a topological order $\mathcal{T}$ of $G - F^*$;
2 $F_0 := F$, $k := 0$;
3 **while** $F_k \neq F^*$ **do**
4 let v_k be the minimal vertex of $F_k \cap (V - F^*)$ with respect to $\mathcal{T}$;
5 $F_{k+1} := \mu_G(F_k, v_k)$;
6 $k := k + 1$;
7 **od**
8 **output** $(F_0, \ldots, F_k)$;

Note that TRANSFORM-DIRECTED-MFVS is not a completely specified algorithm; the topological ordering in line 1 contains an ambiguity which can be resolved arbitrarily. Yet the following lemma asserts the correctness of TRANSFORM-DIRECTED-MFVS.

Lemma 1. *For any directed graph $G = (V, E)$, minimal feedback vertex sets F and F^*, and any successor function μ_G of G,* TRANSFORM-DIRECTED-MFVS(G, F, F^*, μ_G) *computes a sequence $F = F_0, \cdots, F_s = F^*$ where $s \leq |V| - |F^*|$, and F_{i+1} is a μ_G-successor of F_i for $i = 0, \ldots, s-1$.*

Proof: Because of line 3, TRANSFORM-DIRECTED-MFVS terminates only if $F_k = F^*$. Thus it remains to show that TRANSFORM-DIRECTED-MFVS terminates after at most $r = |V| - |F^*|$ iterations of the *while* loop.

W.l.o.g. we can assume that $|V| = \{1, \ldots, n\}$ and $(1, \ldots, r)$ is the topological order $\mathcal{T}$ of $G - F^*$. A topological order of $V - F^*$ always exists, since F^* is a feedback vertex set, and thus $G - F^*$ is acyclic.

Let k be a non-negative integer. Then

$$(F_k \cap (V - F^*) = \emptyset) \iff (F_k \subseteq F^*) \iff (F_k = F^*),$$

due to the minimality of F_k and F^*. Thus, if the condition in line 3 holds, the statement in line 4 is well-defined.

Further note that $v' > v_k$ holds for all $v' \in (F_k - v_k)$, because of the minimality of v_k w.r.t. $\mathcal{T}$, and $v' > v_k$ for all $v' \in F^*$. Moreover, $v' > v_k$ also holds for all $v' \in N^+(v_k)$, according to the fact that $v_k \in V - F^*$ and $(1, \ldots, r)$ is a topological order of $G - F^*$.

Therefore we have $v' > v_k$ for all $v' \in (F_k - v_k) \cup N^+(v_k)$, and thus, all $v' \in F_{k+1}$, because $F_{k+1} = \mu_G(F_k, v_k) \subseteq F_k - v_k \cup N^+(v_k)$. Particularly, $v' > v_k$ for $v' = v_{k+1} \in F_{k+1}$ in line 4, hence

$$v_{k+1} > v_k.$$

Consequently, $v_0 < v_1 < v_2 < \ldots$ Since $v_k \in (V - F^*) = \{1, \ldots, r\}$ for all non-negative k, the algorithm can perform at most $r = |V| - |F^*|$ *while* loops and outputs $F = F_0, \cdots, F_s = F^*$ with $s \leq |V| - |F^*|$, which proves the claim. ∎

Computing all minimal feedback vertex sets. It can now be seen that all minimal solutions can be generated by exhaustive search in the *superstructure graph* $\Phi(G, \mu_G)$.

The vertex set of $\Phi(G, \mu_G)$ consists of all MFVSs F of G, and for each such F there are directed arcs from F to each μ_G-successor of F. Starting with an initial MFVS $F = F_0$, all successors of F in $\Phi(G, \mu_G)$ are generated ("expansion" of F). Then a "still unexpanded" solution is determined and the process reiterates until all generated solutions have been expanded.

Lemma 1 asserts that $\Phi(G, \mu_G)$ is strongly connected. Hence indeed all minimal solutions are generated by an exhaustive search on $\Phi(G, \mu_G)$. For this purpose, the following algorithm uses a queue Q and a dictionary D.

Algorithm GENERATE-MFVS (G, μ_G)

```
1   compute a minimal admissible solution F_0;
2   insert F_0 into Q and into D;
3   while Q is not empty do
4           remove any set F from Q;
5           output F;
6           for each μ_G-successor F' of F do
7               if F' is not contained in D
8                  insert F' into D and Q;
9               fi
10          od
11  od
```

Minimizing a feedback vertex set by "removing redundant vertices". Starting from a given FVS X, a MFVS $F' \subseteq X$ can be computed by checking for each $v \in X$ if $X - v$ is a FVS for G and, if this holds, v is removed from X. When this has been done once for each $v \in X$, the remaining FVS $F' \subseteq X$ is minimal. Concerning the computational complexity of the whole operation, a single check for $v \in F$ can be performed using depth-first search in time $\mathcal{O}\big(|V| + |E|\big)$. Minimizing a FVS can thus be accomplished in $\mathcal{O}\big(|V|\,(|V| + |E|)\big)$.

Overall Computational Complexity. Generating the initial MFVS F_0 in line 1 of GENERATE-MFVS is accomplished in $\mathcal{O}\big(|V|\,(|V| + |E|)\big)$ by removing redundant vertices, starting with $X = V$. Removing redundant vertices can also be used to compute a μ_G-successor of F in line 6. The minimization starts with $X = F - v \cup N^+(v)$ with $v \in F$. One μ_G-successor is computed in time $\mathcal{O}\big(|V|\,(|V| + |E|)\big)$. Using a lexicographical order of V and *tries* [1] for the implementation of D, operations on D and Q can be executed in time $\mathcal{O}\big(|V|\big)$ per operation.

For a MFVS F of a directed graph, there are at most $|V|$ μ_G-successors F' to consider in the *for* loop of lines 6–10. Thus, one *while* loop is executed in time $\mathcal{O}\big(|V|^2(|V|+|E|)\big)$, which makes a polynomial delay for the successive output of MFVS.

This proves the following theorem.

Theorem 1. *Given any directed graph G, Algorithm* GENERATE-MFVS *can be used to compute all minimal feedback vertex sets of G with a polynomial delay of $\mathcal{O}\big(|V|^2(|V| + |E|)\big)$.*

Note that memory requirements are polynomial for graphs with a polynomial number of MFVS, but potentially exponential for the general case.

3 Feedback vertex sets of undirected graphs

The algorithm for the undirected case and its proof of correctness are similar to the directed case. The concepts adapt to the undirected case as follows.

Let $G = (V, E)$ be an undirected graph. W.l.o.g. we assume G to be connected. By $N(v)$ we will denote the set of $w \in V$ s.t. $\{v, w\} \in E$.

In the directed case the proof of correctness relies on the topological order of the "remainder graph" $G - F^*$. There, $G - F^*$ is successively "cleared" by replacing a vertex $v_k \in F_k$ by a μ_G-successor $\mu_G(F_k, v_k)$. For undirected graphs G, the arcs of $G - F^*$ are undirected. In order to "clear" $G - F^*$, a direction will be associated with each of its arcs, and the additional directionality will be reflected by a third parameter in the definition of a μ_G-successor.

Basic definitions. For a MFVS F of G, $v \in F$, $w \in N(v)$, observe that $F' = F - v \cup (N(v) - w)$ is a FVS of G. This is because any cycle that contains v also contains at least one vertex of $N(v) - w$.

We call each MFVS $F'' \subseteq F - v \cup (N(v) - w)$ a *(v, w)-successor of F*. In analogy to the directed case, we assume that a function $\mu_G : 2^V \times V \times V \longrightarrow 2^V$ assigns a fixed (v, w)-successor F''_0 of F to any such triplet (F, v, w). $F''_0 = \mu_G(F, v, w)$ is also called *μ_G-successor of F*.

Let us assume that F^* is a MFVS of G. Then $G' = G - F^*$ is a union of undirected trees. Choosing a vertex in each tree in G' and directing the arcs away from these "root vertices" yields a directed acyclic graph that we call $T(G')$.

With each vertex $v \in G'$ we now associate a vertex $p_T(v)$ from G'. When v is a root vertex in $T(G')$, we set $p_T(v) := w$ for any $w \in N(v)$. Otherwise, v has a unique predecessor w in $T(G')$ and we set $p_T(v) := w$.

Given the undirected graph G, two feedback vertex sets F and F^* and a successor function μ_G, the following algorithm transforms F into F^* by generating μ_G-successors.

Algorithm TRANSFORM-UNDIRECTED-MFVS $(G = (V, E), F, F^*, \mu_G)$

compute a topological order $\mathcal{T}$ of $T(G - F^*)$;
$F_0 := F$, $k := 0$;
while $F_k \neq F^*$ **do**
    let v_k be the minimal vertex of $F_k \cap (V - F^*)$ with respect to $\mathcal{T}$;
    $F_{k+1} := \mu_G(F_k, v_k, p_T(v_k))$;
    $k := k + 1$;
od
output $(F_0, \ldots, F_k)$;

The following lemma asserts the correctness of the algorithm.

Lemma 2. *For any undirected graph G, minimal feedback vertex sets F and F^*, and any successor function μ_G of G,* Transform-Undirected-MFVS(G, F, F^*, μ_G) *computes a sequence $F = F_0, \cdots, F_s = F^*$ where $s \leq |V| - |F^*|$, and F_{i+1} is a μ_G-successor of F_i for $i = 0, \ldots, s-1$.*

The *proof* translates almost literally from the directed case.

Algorithm. Analogously to the directed case, Lemma 2 asserts that the superstructure graph $\Phi(G, \mu_G)$ is strongly connected. We conclude that an exhaustive search on $\Phi(G, \mu_G)$ discovers all MFVSs of G. Thus, using the notion of a μ_G-successor for undirected graphs, algorithm Generate-MFVS(G, μ_G) indeed generates all MFVSs of G.

Computational complexity. Minimizing a FVS of an undirected graph can be accomplished by iteratively removing redundant vertices. The procedure is analogous to section 2, taking time $\mathcal{O}\big(|V|\,(|V| + |E|)\big)$. Further, for a MFVS F of an undirected graph there are at most $2\,|E|$ μ_G-successors to consider in the *for* loop in lines 6–10 of Generate-MFVS. This is because for each arc $\{v, w\} \in E$ there can be at most two μ_G-successors F' of F, namely $\mu_G(F, v, w)$ and $\mu_G(F, w, v)$. Thus the delay between the output of successive MFVSs is $\mathcal{O}\big(|V|\,|E|\,(|V| + |E|)\big)$. This establishes the following theorem.

Theorem 2. *Given any undirected graph G, Algorithm* Generate-MFVS *can be used to compute all minimal feedback vertex sets of G with a polynomial delay of $\mathcal{O}\big(|V|\,|E|\,(|V| + |E|)\big)$.*

4 Minimal feedback arc sets of directed graphs

We can use the algorithm for feedback vertex sets from section 2 to calculate feedback arc sets. This is based upon the close relationship between the feedback arc sets of a graph and the feedback vertex sets of its line graph. The *line graph* G' of a directed graph $G = (V, E)$ is a directed graph G' that has a vertex $v'(e)$ for each arc $e \in E$ and an arc $e' = (v'(e_1), v'(e_2))$ for any two arcs $e_1 = (x, y) \in E$ and $e_2 = (y, z) \in E$. Notice that each cycle in G corresponds to a cycle in G' and vice-versa. Hence the feedback arc sets of G correspond to the feedback vertex sets of G'. Since G' has $\mathcal{O}\big(|E|\big)$ vertices and $\mathcal{O}\big(|E|^2\big)$ arcs, it follows from Theorem 1 that we can calculate the feedback arc sets G with a time complexity of $\mathcal{O}\big(|E|^4\big)$ per minimal feedback arc set.

We present a variation that only uses time $\mathcal{O}\big(|V|\,|E|\,(|V| + |E|)\big)$ per minimal solution. Still, the procedure will be quite similar to the method outlined in section 2. Basically vertices and arcs exchange their roles.

Definitions. Let $G = (V, E)$ be a directed graph, $F \subseteq E$ be a minimal feedback arc set *(MFAS)*, i.e. $G - F = (V, E \setminus F)$ is acyclic and F is minimal with this property.

For $e = (v, w) \in E$, we set $S(e) := w$, for $X \subseteq E$ we define $S(X) := \cup_{e \in X} S(e)$. We set $A^-(w) := \{(x, w) \in E\}$, and $A^+(w) := \{(w, x) \in E\}$.

Notice that, for any $w \in V$, each cycle containing an arc in $A^-(w)$ must also contain an arc in $A^+(w)$. Thus, $F' = F - A^-(w) \cup A^+(w)$ is a FAS of G for any $w \in V$. We call each MFAS $F'' \subseteq F - A^-(w) \cup A^+(w)$ a *(w-)successor of F*.

We assume that an arbitrary w-successor $F'' = \mu_G(F, w)$, of F is fixed for every MFAS F and $w \in S(F)$. We call μ_G a *successor function* and F'' a *μ_G-successor* of F. The following algorithm transforms a MFAS F into a MFAS F^* by generating μ_G-successors.

Algorithm TRANSFORM-DIRECTED-MFAS $(G = (V, E), F, F^*, \mu_G)$

1 compute a topological order $\mathcal{T}$ of $G - F^*$;
2 $F_0 := F$, $k := 0$;
3 **while** $F_k \neq F^*$ **do**
4   let v_k be the minimal vertex of $S(F_k \cap (E - F^*))$ with respect to $\mathcal{T}$;
5     $F_{k+1} := \mu_G(F_k, v_k)$;
6     $k := k + 1$;
7 **od**
8 **output** $(F_0, \ldots, F_k)$;

Figure 2 illustrates the situation in line 4 of the algorithm. The dashed arcs are the members of the current solution F_k. By moving from F_k to a v_k-successor F_{k+1}, the algorithm iteratively clears the shaded area of dashed arcs, from left to right.

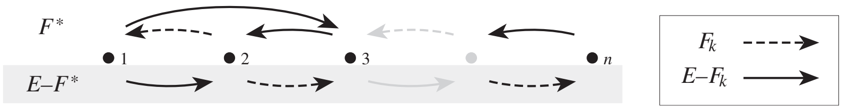

Figure 2: Situation during the execution of line 4 in TRANSFORM-DIRECTED-MFAS. v_k is the leftmost target of a dashed arc in the shaded area; in this case, v_k=3

Lemma 3. *For any directed graph $G = (V, E)$, minimal feedback arc sets F and F^*, and any successor function μ_G of G,* TRANSFORM-DIRECTED-MFAS (G, F, F^*, μ_G) *computes a sequence $F = F_0, \cdots, F_s = F^*$ where $s \leq |V| - 1$, and F_{i+1} is a μ_G-successor of F_i for $i = 0, \ldots, s - 1$.*

Proof: Notice that, in line 1, a topological order $\mathcal{T}$ of $G - F^*$ always exists, since F^* is a feedback arc set, and thus $G - F^*$ is acyclic. W.l.o.g. we can assume that $|V| = \{1, \ldots, n\}$ and $\mathcal{T} = (1, \ldots, n)$.

Because of line 3, the algorithm terminates only if $F_k = F^*$. Thus it remains to show that TRANSFORM-DIRECTED-MFAS terminates after at most $|V| - 1$ iterations of the *while* loop.

Let k be a non-negative integer. Then

$$(F_k \cap (E - F^*) = \emptyset) \iff (F_k \subseteq F^*) \iff (F_k = F^*),$$

due to the minimality of F_k and F^*. Thus, if the condition in line 3 holds, the statement in line 4 is well-defined.

Further note that $v' > v_k$ holds for all $v' \in S((F_k - A^-(v_k)) \cap (E - F^*))$, because of the minimality of v_k w.r.t. $\mathcal{T}$. Moreover, $v' > v_k$ also holds for all $v' \in S(A^+(v_k) \cap (E - F^*))$, according to the fact that $(1, \ldots, n)$ is a topological order of $G - F^*$.

Hence we have $v' > v_k$ for all $v' \in S((F_k - A^-(v_k) \cup A^+(v_k)) \cap (E - F^*))$, and thus, all $v' \in S(F_{k+1} \cap (E - F^*))$, because $F_{k+1} = \mu_G(F_k, v_k) \subseteq S(F_k - A^-(v_k) \cup A^+(v_k))$. Particularly, $v' > v_k$ holds for $v' = v_{k+1} \in F_{k+1}$ in line 4, hence

$$v_{k+1} > v_k.$$

Consequently, $v_0 < v_1 < v_2 < \ldots$ Further observe that 1 is the first vertex of the topological order $\mathcal{T}$ of $G - F^*$, thus 1 cannot be contained in $S(E - F^*)$. But $v_0 \in S(E - F^*)$ due to line 4 of the algorithm. Hence $v_k \in \{2, \ldots, |V|\}$ for all non-negative k, thus the algorithm can perform at most $|V| - 1$ *while* loops and outputs $F = F_0, \cdots, F_s = F^*$ with $s \leq |V| - 1$, which proves the claim. ■

Algorithm. Again the above lemma asserts that the superstructure graph $\Phi(G, \mu_G)$ is strongly connected. Thus, when applied to the feedback arc set problem, Algorithm GENERATE-MFVS indeed computes all minimal feedback arc sets of G for any successor function μ_G.

Computational complexity. We examine the computational complexity of one *while* loop in Algorithm GENERATE-MFVS. Minimizing a feedback arc set is accomplished by removing redundant arcs, in analogy to the minimization procedure of section 2. Since $\mathcal{O}(|E|)$ arcs have to be checked, the complexity for this operation is in $\mathcal{O}(|E|\,(|V| + |E|))$. Since there can be at most $|V|$ μ_G-successors $\mu_G(F, w)$ for any MFAS F, the *while* loop takes at most time $\mathcal{O}(|V|\,|E|\,(|V| + |E|))$.

Theorem 3. *Given any directed graph G, Algorithm* GENERATE-MFVS *can be used to compute all minimal feedback arc sets of G with a polynomial delay of $\mathcal{O}(|V|\,|E|\,(|V| + |E|))$.*

References

[1] E. Fredkin, Trie memory, Comm. ACM 3 (Sep. 1960) 490–499.

[2] D. S. Hochbaum, Various notions of approximations: Good, better, best and more, in: D. S. Hochbaum, ed., Approximation Algorithms for NP-hard problems (PWS Publishing, 1995), chap. 9, p. 350.

[3] D. S. Johnson, M. Yannakakis and C. H. Papadimitriou, On generating all maximal independent sets, Inf. Proc. Lett. 27 (1988) 119–123.

[4] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller and J. W. Thatcher, eds., Complexity of Computer Computations (Plenum Press, New York, London, 1972), 85–103.

[5] E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, SIAM J. Comput. 9 (1980) 558–565.

[6] P. M. Pardalos and J. Xue, The maximum clique problem, J. Glob. Opt. (1994) 301–328.

[7] E. M. Reingold, J. Nievergelt and N. Deo, Combinatorial Algorithms: Theory and Practice (Prentice-Hall, Englewood Cliffs, USA, 1977), chap. 8.

[8] S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (Sep. 1977) 505–517.