Journal of Mathematical Research & Exposition
Vol.24, No.4, 610–618, November, 2004

A Linear Time Algorithm for Minimum-Weight Feedback Vertex Set Problem in Outerplanar Graphs *

ZHANG Shao-qiang[1], *WANG Xiao-li*[2,4], *LI Guo-jun*[2,3]

(1. School of Comp. & Info. Engineering, Tianjin Normal University, Tianjin 300074, China;
2. School of Mathematics and Systems Science, Shandong University, Jinan 250100, China;
3. Inst. of Software, Chinese Academy of Sciences, Beijing 100080, China;
4. Dept. of Math., Nanyang Normal University, Henan 473061, China)

Abstract: A subset of the vertex set of a graph is a feedback vertex set of the graph if the resulting graph is a forest after removing the vertex subset from the graph. In this paper, we study the minimum-weight feedback vertex set problem in outerplanar graphs and present a linear time algorithm to solve it.

Key words: outerplanar graphs; feedback vertex set; linear time algorithm.

Classification: AMS(2000) 05C85, 90B18, 90C35/CLC number: O157.5, TP301

Document code: A **Article ID**: 1000-341X(2004)04-0610-09

Given an undirected graph $G = (V, E)$ and nonnegative weights w_v for the vertices $v \in V$, the minimum-weight feedback vertex set problem (FVS) is to find a minimum-weight set of vertices F that meets every cycle of G. Alternatively, the problem is to find a minimum-weighted F such that $G[V-F]$ is acyclic, where $G[S]$ denotes the subgraph of G induced by the vertex set S. We say that a set of vertices F is an *FVS* if it is a feasible solution to the problem.

The minimum feedback vertex set problem has long been known to be NP-hard (See [6], Problem GT7); hence, researchers have attempted to find *approximation algorithms* for the problem. An α-approximation algorithm for FVS runs in polynomial time and finds an FVS whose weight is no more than α times the weight of an optimal FVS. The value α is sometimes called the *performance guarantee* of the algorithm. Until now, the best performance guarantee for FVS in general graphs is 2. Two slightly different 2-approximation algorithms were given by Bafna, Berman and Fujito[2] and Becker and Geiger[3]. In order to study wavelength conversion in optical networks based on *wavelength division multiplexing* (WDM), Kleinberg and Kumar[8] developed the first *polynomial-time approximation scheme* (PTAS) for the cardinality feedback vertex set problem in planar graphs.

*Received date: 2002-11-26
Foundation item: Supported by National Natural Science Found of China (60373025),
Biography: ZHANG Shao-qiang (1976-), male, Ph.D.

There may be exact polynomial time algorithms of the problem for some special graphs. For example, LI and LIU[9] provided a polynomial algorithm for the cardinality FVS problem in 3-regular simple graphs. Many problems can be solved in polynomial time in graphs with bounded tree-width by tree-decomposition[1]. A graph is *outerplanar* if it has a planar embedding with all vertices lying on a single face. The main contribution of this paper is to discard the tree-decomposition and provide a new linear time exact algorithm for the minimum-weight FVS problem restricted to outerplanar graphs by some "contraction operations", which contract the graph while preserving all important properties and information relevant to the problem.

1. Preliminaries

In this paper we consider only finite simple graphs $G = (V, E)$ with vertex set V and edge set E; that is, there are no *self-loops* or *multiple edges*. A *homeomorph* of a graph G is a graph obtained from G when its edges are subdivided into paths by inserting new vertices of degree two. An equivalent definition of outerplanar graphs is as follows:

Definition 1[5] *A graph is outerplanar if and only if it has no subgraph homeomorphic to $K_{2,3}$ (a complete bipartite on a set of two vertices and a set of three vertices) or K_4 (a complete graph on four vertices).*

Clearly, outerplanar graphs are special planar graphs. Since any n-vertex planar graph with $n \geq 3$ contains no more than $3n - 6$ edges[8], we get $O(|V|) = O(|E|)$ for any outerplanar graph $G = (V, E)$.

A graph is *2-edge-connected* if the deletion of a single edge does not disconnect the graph. A graph is *2-connected* if the deletion of a single vertex does not disconnected the graph. We can first run a linear (with respect to $O(|V| + |E|) = O(|V|)$) algorithm[4] to decompose the input graph into its 2-edge-connected components, then work with each 2-edge-connected component separately, since the edges that are *cut edges* cannot be on any cycle. Therefore, we can assume that the graph under consideration is a 2-edge-connected outerplanar graph.

A *plane graph* is a particular drawing of a planar graph in the plane without edge crossings. Each plane graph has exactly one unbounded face, called the *exterior face*; the other faces can be called *interior faces*. Clearly, a graph has a cycle if and only if it has an interior face. Thus, if f is an interior face of a plane graph, then any feedback vertex set contains a vertex on f. An *outerplane graph* is a planar embedding of an outerplanar graph with every vertex on the exterior face.

Let G be a 2-edge-connected outerplane graph. Clearly, the boundary of an interior face f forms a cycle of G, called a *minimal cycle*. The *degree* of a face f is the number of edges on the boundary of f. Since any outerplanar graph contains no homeomorph of $K_{2,3}$, any two minimal cycles(interior faces) share at most one edge with each other.

Given a plane graph G and its dual G^*, the *weak dual* of G is the graph obtained from the dual G^* by deleting the vertex corresponding to the exterior face of G. Since any outerplanar graph contains no homeomorph of K_4, we can observe that

(i) The weak dual of an outerplane graph is a forest.

(ii) The weak dual of a 2-connected outerplane graph is a tree.

Let v be a vertex of degree two with two neighbors v' and v'' in a given 2-edge-connected outerplanar graph G. If $w_v \geq min\{w_{v'}, w_{v''}\}$, then there is such an optimal FVS that contains no the vertex v. A 2-edge-connected outerplanar graph is *contracted* if there is no vertex of degree 2 v with two neighbors v' and v'' such that $w_v \geq \min\{w_{v'}, w_{v''}\}$. If a 2-edge-connected outerplanar graph G is not contracted, we can use the following algorithm to contract G.

Algorithm Contraction

1. Input a 2-edge-connected outerplanar graph $G = (V, E)$ and label all vertices *unvisited*.

2. Select a vertex v of degree two. Let vertices v' and v'' be its two neighbors. If $(v', v'') \notin E$ and $w_v \geq min\{w_{v'}, w_{v''}\}$, then $V \leftarrow V - \{v\}$ and $E \leftarrow (E - \{(v, v'), (v, v'')\}) \cup \{(v', v'')\}$. Else, label v *visited*.

3. Repeat Step 2 until all vertices of degree 2 are visited and there is no vertex to be removed. Output $G = (V, E)$.

In the algorithm contraction, a vertex may be visited more than once if we always selected a vertex in Step 2 randomly. However, the algorithm may run in linear time by depth-first search. It follows that in this paper we need only to consider the FVS problem with respect to a contracted 2-edge-connected outerplanar graph.

The edges on the boundary of the exterior face of a plane graph are called *outer edges* and the other edges are called *inner edges*. An interior face is called to be *initiatory* if its boundary contains at most one inner edge. It is clear to see that the boundary of any initiatory face is a triangle for a contracted 2-edge-connected outerplanar graph. Clearly, for each leaf (vertex of degree 1) of the weak dual of an outerplanar graph G, the corresponding face is initiatory. It is well-known that every nontrivial tree has at least two vertices of degree one. Analogically, we have

Theorem 1 *If G is a 2-edge-connected outerplanar graph, then G contains at least one initiatory faces.*

We can begin from an initiatory face to seek feedback vertices. A *maximal outerplanar graph* (MOG) is an outerplanar graph such that no edge can be added without violating this property[7].

Theorem 2 *Let $G(V, E)$ be a maximal outerplanar graph with $|V| \geq 3$, then (i) each interior face is triangular; (ii) connectivity of G is $\kappa(G) = 2$.*

In the following section, we shall consider the minimum-weight feedback vertex set of maximal outerplanar graphs. In Section 3, we shall consider the problem in general outerplanar graphs.

If G_1 and G_2 are subgraphs of G, the *union* $G_1 \cup G_2$ of G_1 and G_2 is the subgraph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. We use $w(S)$ to denote the weight of a given vertex set S. Other notation and terminology not defined here can be found in [4].

2. Feedback vertex sets in maximal outerplanar graphs

Given a maximal outerplanar graph $G(V, E)$, we shall study the feedback vertex set of G in this section.

Since G is 2-connected, the weak dual of G is a tree. Corresponding to each triangle f of G there is a vertex f^* of the weak dual of G; and corresponding to the edge e shared by two triangles f and g there is an edge e^* joining the corresponding vertices f^* and g^* of the weak dual of G. Clearly, any edge shared by two triangles can separate the graph into two parts. Given a subgraph G_1 of G, we say that G_1 is a *subgraph separated by edge* (u, v) *from* G if there is exactly one edge (u, v) with its two ends u and v in G_1 adjacent to vertices not in G_1.

As shown in Figure 1, we optionally select a triangle uvw from G. Since the weak dual of G is a tree, uvw separates G into three parts. Let G_{uv} be a subgraph of G sharing the edge (u, v) with the triangle uvw and separated by (u, v) from G. Analogously, G_{uw} and $\overline{G}_{vw}$ are defined to be separated by (u, w) and (v, w) from G, respectively. Obviously, any two of G_{uv}, G_{uw} and $\overline{G}_{vw}$ are edge-disjoint and $G = G_{uv} \cup G_{uw} \cup \overline{G}_{vw}$.

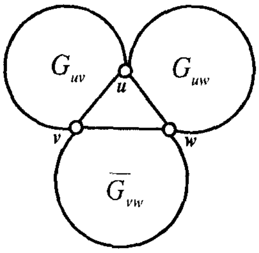

Figure 1

We define four feedback vertex sets of G_{uv} as follows:

(1) An optimal FVS in the case of containing u but no v, denoted by $F_u^{(u,v)}$;

(2) An optimal FVS in the case of containing v but no u, denoted by $F_v^{(u,v)}$;

(3) An optimal FVS in the case of containing both u and v, denoted by $F_{uv}^{(u,v)}$;

(4) An optimal FVS in the case of containing neither u nor v, denoted by $F_o^{(u,v)}$;

Since whether an optimal FVS of G_1 contains either u or v was memorized in the four feedback sets defined above, there must exist one among the four vertex sets which can be extended into an optimal FVS of G. Thus, we call the four feedback vertex sets of G_{uv} as *candidate sets associated with* (u, v), and use $\mathcal{F}_{uv}$ to denote the collection of the four candidate sets of G_{uv}, that is, $\mathcal{F}_{uv} = \{F_u^{(u,v)}, F_v^{(u,v)}, F_{uv}^{(u,v)}, F_o^{(u,v)}\}$. Similarly, $\mathcal{F}_{uw} = \{F_u^{(u,w)}, F_w^{(u,w)}, F_{uw}^{(u,w)}, F_o^{(u,w)}\}$.

Suppose we have obtained two collections of candidate sets $\mathcal{F}_{uv}$ of G_{uv} and $\mathcal{F}_{uw}$ of G_{uw}. We now compute the four candidate sets of $G_{vw} = G_{uv} \cup G_{uw} \cup uvw$.

1. Find a set of minimum weight, denoted by $F_v^{(v,w)}$, in $\{F_v^{(u,v)} \cup F_u^{(u,w)}, F_v^{(u,v)} \cup F_o^{(u,w)}, F_{uv}^{(u,v)} \cup F_u^{(u,w)}, F_{uv}^{(u,v)} \cup F_o^{(u,w)}\}$ (Since $w(F_o^{(u,v)} \cup F_o^{(u,w)} \cup \{v\}) \geq w(F_v^{(u,v)} \cup F_o^{(u,w)})$, we can ignore the union of $F_o^{(u,v)}$, $F_o^{(u,w)}$ and v.). Clearly, $F_v^{(v,w)}$ is an optimal FVS of G_{vw} in the case of containing v but no w.

2. Find a set of minimum weight, denoted by $F_w^{(v,w)}$, in $\{F_u^{(u,v)} \cup F_w^{(u,w)},\ F_o^{(u,v)} \cup F_w^{(u,w)},\ F_u^{(u,v)} \cup F_{uw}^{(u,w)},\ F_o^{(u,v)} \cup F_{uw}^{(u,w)}\}$ (Ignore $F_o^{(u,v)} \cup F_o^{(u,w)} \cup \{w\}$.). Clearly, $F_w^{(v,w)}$ is an optimal FVS of G_{vw} in the case of containing w but no v.

3. Find a set of minimum weight, denoted by $F_{vw}^{(v,w)}$, in $\{F_v^{(u,v)} \cup F_w^{(u,w)},\ F_v^{(u,v)} \cup F_{uw}^{(u,w)},\ F_{uv}^{(u,v)} \cup F_w^{(u,w)},\ F_{uv}^{(u,v)} \cup F_{uw}^{(u,w)}\}$ (Since $w(F_o^{(u,v)} \cup F_o^{(u,w)} \cup \{v,w\}) \geq w(F_v^{(u,v)} \cup F_w^{(u,w)})$, we ignore the union of $F_o^{(u,v)}$, $F_o^{(v,w)}$ and $\{v,w\}$). Clearly, $F_{vw}^{(v,w)}$ is an optimal FVS of G_{vw} in the case of containing both w and v.

4. Find a set of minimum weight, denoted by $F_o^{(v,w)}$, in $\{F_u^{(u,v)} \cup F_u^{(u,w)},\ F_u^{(u,v)} \cup F_o^{(u,w)},\ F_o^{(u,v)} \cup F_u^{(u,w)}\}$ (Since $w(F_o^{(u,v)} \cup F_o^{(u,w)} \cup \{u\}) \geq w(F_u^{(u,v)} \cup F_o^{(u,w)})$, we can ignore the union of $F_o^{(u,v)}$, $F_o^{(u,w)}$ and u.). Clearly, $F_o^{(v,w)}$ is an optimal FVS of G_{vw} in the case of containing neither w nor v.

Therefore, we can use constant time comparisons to obtain a collection of candidate sets associated with (v,w) of G_{vw}, denoted by $\mathcal{F}_{vw} = \{F_v^{(v,w)},\ F_w^{(v,w)},\ F_{vw}^{(v,w)},\ F_o^{(v,w)}\}$. If (u,v) is an outer edge, then we can obtain four candidate sets associated with (u,v) as follows:

$$F_u^{(u,v)} = \{u\},\ F_v^{(u,v)} = \{v\}, F_{uv}^{(u,v)} = \{u,v\},\ F_o^{(u,v)} = \emptyset;$$

Since an initiatory triangle contains at least two outer edges, we can break triangles from an initiatory triangle. Pick a vertex u of degree two with its two neighbors v and w. Clearly, uvw is an initiatory triangle and both (u,v) and (u,w) are outer edges. Using the collections of candidate sets $\mathcal{F}_{uv}$ and $\mathcal{F}_{uw}$, we can compute a collection of candidate sets $\mathcal{F}_{vw}$ associated with (v,w). Then delete u, (u,v) and (u,w) from G. (v,w) becomes a new outer edge of the residual graph. Now, the entire process is repeated on the subgraph $G[V-u]$. The process terminates when the residual graph contains no triangle. In particular, we can select a triangle containing an outer edge (v',v'') of G and set the last triangle to be broken is the triangle containing $(v'v'')$. Finally, we would obtain a collection of candidate sets associated with (v',v'') of G. We can obtain an optimal FVS finally because the optimal solutions for all cases was kept in memory at each iterative step. The algorithm is given below.

Algorithm MOG

1. Input a maximal outerplanar graph G and select an outer edge (v',v'') from G. There are no labels for all edges.

2. If G consists of exactly one edge (v',v''), output a candidate set of minimum weight in $\mathcal{F}_{v'v''}$ and stop. Else, go to Step 3.

3. Select a vertex u of degree two with two neighbors v and w and $(v',v'') \notin \{(u,v),\ (u,w)\}$. If there is no label on (u,v), then set $\mathcal{F}_{uv} = \{F_u^{(u,v)},\ F_v^{(u,v)},\ F_{uv}^{(u,v)},\ F_o^{(u,v)}\} = \{\{u\},\ \{v\},\ \{u,v\},\ \emptyset\}$. If there is no label on (u,w), then set $\mathcal{F}_{uw} = \{F_u^{(u,w)},\ F_w^{(u,w)},\ F_{uw}^{(u,w)},\ F_o^{(u,w)}\} = \{\{u\},\ \{w\},\ \{u,w\},\ \emptyset\}$. Use $\mathcal{F}_{uv}$ and $\mathcal{F}_{uw}$ to compute $\mathcal{F}_{vw}$. Then set $G \leftarrow G[G-u]$ and go to Step 2.

It is easy to see that the main loop of Algorithm MOG executes elementary operations totally within constant times. Therefore, the complexity of the algorithm is linear.

3. Feedback vertex sets in general outerplanar graphs

Now we consider feedback vertex sets in general outerplanar graphs. Let G be a contracted 2-edge-connected outerplanar graph.

A *block of a graph* is a maximal connected subgraph that has no cut vertices. Since we only consider 2-edge-connected outerplanar graphs in this paper, every block of G is 2-connected. We can find all blocks and cut vertices of G by a spanning tree T. Corresponding to each $e \notin T$ there is a unique cycle in $T + e$. If such two cycles have at least two common vertices, then they are in an identical block. Therefore, we can find all blocks and cut vertices of G in $O(|V| + |E|) = O(|V|)$ time[4].

If there are at least two blocks in G, then there are at least two blocks that each contain exactly one cut vertex. Given a block b containing exactly one cut vertex u of G, we define two feedback vertex sets:

(i) an optimal FVS of b in the case of containing the cut vertex u, denoted by F_u^b;

(ii) an optimal FVS of b in the case of containing no u, denoted by F_o^b.

It is obvious to see that either of the two feedback vertex sets of b can be extended into an optimal FVS of G. We call the two feedback vertex sets of b as *candidate sets associated with* u, and use $\mathcal{F}_u^b$ to denote the collection of the two candidate sets of b. i.e., $\mathcal{F}_u^b = \{F_u^b, F_o^b\}$. Now, we will design a polynomial algorithm to find a collection of candidate sets associated with u of b.

Let f be an interior face of degree k in b and C the minimal cycle corresponding to f. As shown in Figure 2, edges (v_1, v_2), (v_2, v_3), $\ldots$, (v_{k-1}, v_k) of C are shared with subgraphs $G_{1,2}$, $G_{2,3}$, $\ldots$, $G_{k-1,k}$ of b, respectively. Suppose that we have obtained $k-1$ collections of candidate sets of $G_{1,2}$, $G_{2,3}$, $\ldots$, $G_{k-1,k}$, denoted by $\mathcal{F}_{v_1v_2}, \mathcal{F}_{v_2v_3}, \ldots, \mathcal{F}_{v_{k-1}v_k}$, respectively. For each $G_{i,i+1}$, $i = 1, 2, \ldots, k-1$, we denote

$$\mathcal{F}_{v_iv_{i+1}} = \{F_{v_i}^{(v_i,v_{i+1})}, F_{v_{i+1}}^{(v_i,v_{i+1})}, F_{v_iv_{i+1}}^{(v_i,v_{i+1})}, F_o^{(v_i,v_{i+1})}\}.$$

We now use these collections of candidate sets to find a collection of candidate sets associated with the last edge (v_1, v_k) of C, denoted by $\mathcal{F}_{v_1v_k}$, of $G_{1,k} = G_{1,2} \cup G_{2,3} \cup \cdots \cup G_{k-1,k} \cup C$.

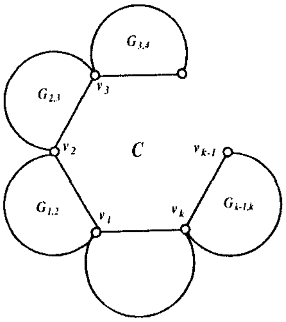

Figure 2

If $k = 3$, we can use the method in Section 3 directly. Else, we define $G_{\langle 1,3\rangle} = G_{1,2} \cup G_{2,3}$, $G_{\langle 1,4\rangle} = G_{\langle 1,3\rangle} \cup G_{3,4}$, ..., $G_{\langle 1,k\rangle} = G_{\langle 1,k-1\rangle} \cup G_{k-1,k}$. For each $i \in \{3, \ldots, k\}$, we can define four candidate sets of $G_{\langle 1,i\rangle}$ associated with v_1 and v_i as follows:

1. an optimal FVS of $G_{\langle 1,i\rangle}$ in the case of containing v_1 but no v_i, denoted by $F_{v_1}^{\langle v_1,v_i\rangle}$;
2. an optimal FVS of $G_{\langle 1,i\rangle}$ in the case of containing v_i but no v_1, denoted by $F_{v_i}^{\langle v_1,v_i\rangle}$;
3. an optimal FVS of $G_{\langle 1,i\rangle}$ in the case of containing both v_1 and v_i, denoted by $F_{v_1 v_i}^{\langle v_1,v_i\rangle}$;
4. an optimal FVS of $G_{\langle 1,i\rangle}$ in the case of containing either v_1 nor v_i, denoted by $F_o^{\langle v_1,v_i\rangle}$.

We use $\mathcal{F}_{\langle v_1,v_i\rangle}$ to denote the collection of the four sets above. We can compare all of possible unions of two candidate sets picked from $\mathcal{F}_{v_1 v_2}$ and $\mathcal{F}_{v_2 v_3}$, respectively, to find a collection of candidate sets $\mathcal{F}_{\langle v_1,v_3\rangle}$. Analogically, we can compute $\mathcal{F}_{\langle v_1,v_4\rangle}$, ..., $\mathcal{F}_{\langle v_1,v_k\rangle}$.

Notice that the candidate set $F_o^{\langle v_1,v_k\rangle}$ in $\mathcal{F}_{\langle v_1,v_k\rangle}$ possibly contains no vertex of the cycle C. If $F_o^{\langle v_1,v_k\rangle}$ contains vertices of C, then $\mathcal{F}_{v_1 v_k} = \mathcal{F}_{\langle v_1,v_k\rangle}$. Otherwise, we assume that $F_o^{\langle v_1,v_k\rangle}$ contains no vertex of C, that is, $F_o^{\langle v_1,v_k\rangle} = F_o^{(v_1,v_2)} \cup F_o^{(v_2,v_3)} \cup \cdots \cup F_o^{(v_{k-1},v_k)}$. In other words, $F_o^{\langle v_1,v_i\rangle} = F_o^{\langle v_1,v_{i-1}\rangle} \cup F_o^{(v_{i-1},v_i)}$, $i = 3, 4, \ldots, k$. We know that each $F_o^{\langle v_1,v_i\rangle}$ is a minimum-weight set of $\{F_{v_{i-1}}^{\langle v_1,v_{i-1}\rangle} \cup F_o^{(v_{i-1},v_i)}$, $F_o^{\langle v_1,v_{i-1}\rangle} \cup F_{v_{i-1}}^{(v_{i-1},v_i)}$, $F_{v_{i-1}}^{\langle v_1,v_{i-1}\rangle} \cup F_{v_{i-1}}^{(v_{i-1},v_i)}$, $F_o^{\langle v_1,v_{i-1}\rangle} \cup F_o^{(v_{i-1},v_i)}\}$. Now, let $\widetilde{F}_o^{\langle v_1,v_i\rangle}$ be a minimum-weight set of $\{F_{v_{i-1}}^{\langle v_1,v_{i-1}\rangle} \cup F_o^{(v_{i-1},v_i)}$, $F_o^{\langle v_1,v_{i-1}\rangle} \cup F_{v_{i-1}}^{(v_{i-1},v_i)}$, $F_{v_{i-1}}^{\langle v_1,v_{i-1}\rangle} \cup F_{v_{i-1}}^{(v_{i-1},v_i)}\}$, $i = 3, 4, \ldots, k$. Clearly, each $\widetilde{F}_o^{\langle v_1,v_i\rangle}$ must contain exactly v_{i-1} of C and each $\widetilde{F}_o^{\langle v_1,v_i\rangle} \cup (\bigcup_{j=i}^{k-1} F_o^{(v_j,v_{j+1})})$ must break all cycles of $G_{1,k}$. Therefore, we select a minimum-weight set from $\{\widetilde{F}_o^{\langle v_1,v_i\rangle} \cup (\bigcup_{j=i}^{k-1} F_o^{(v_j,v_{j+1})})$, $i = 3, 4, \ldots, k\}$ as the candidate set containing neither v_1 nor v_k of $G_{1,k}$. Clearly, the other candidate sets of $G_{1,k}$ are equal to the corresponding candidate sets of $G_{\langle 1,k\rangle}$, respectively.

In addition, since we want to find a collection of candidate sets associated with the cut vertex u, we can select an outer edge (u, u') containing u and try to find a collection of candidate sets associated with (u, u'). The algorithm for finding a collection of candidate sets associated with u of b is given below.

Algorithm Block

1. Input a block $b = (V_b, E_b)$ where V_b is the set of vertices and E_b is the set of edges of b. Select an outer edge (u, u') of b containing the cut vertex u and compute the candidate sets of each outer edge except (u, u').

2. If b contains at least one cycle, go to Step 3. Else, go to Step 4.

3. Pick such a cycle $C = (V_c, E_c)$ that all the edges except (v, w) are outer edges and $(u, u') \notin E_c - \{(v, w)\}$(notice that the edge (u, u') just is the edge (v, w) provided that b is C). Compute the collection of candidate sets associated with (v, w), $\mathcal{F}_{vw}$. $b \leftarrow b[V_b - (V_c - \{v, w\})]$ and go to Step 2.

4. Select an optimal candidate set F_u^b containing u and an optimal candidate set F_o^b containing no u from $\mathcal{F}_{vw}$ and output them.

From the definition of $\mathcal{F}_{\langle v_1,v_i\rangle}$ and each $\widetilde{F}_o^{\langle v_1,v_i\rangle}$, the complexity of algorithm block is linear in the size of the block. If G is a block itself, we can use algorithm block to find an optimal FVS of G. So we suppose G consists of at least two blocks. Pick a block containing exactly one cut vertex, supposing it is v, of G. Possibly there are more than one block containing the unique cut vertex v. Let $b_1, b_2, \ldots, b_l$ be these blocks each of which contains the unique cut vertex v.

For each b_i, $1 \le i \le l$, using algorithm block, we can compute a collection of candidate sets $\mathcal{F}_v^{b_i} = \{F_v^{b_i}, F_o^{b_i}\}$. Let $B = b_1 \cup \cdots \cup b_l$. We can define two candidate sets of B as follows:

1. an optimal FVS of B in the case of containing v, denoted by F_v^B;
2. an optimal FVS of B in the case of containing no v, denoted by F_o^B.

We use $\mathcal{F}_v$ to denote the collection of the two candidate sets of B. Since $b_1, b_2, \ldots, b_l$ only share the vertex v with one another, we can set $F_o^B = F_o^{b_1} \cup F_o^{b_2} \cup \cdots \cup F_o^{b_l}$. For each $\mathcal{F}_v^{b_i}$, $1 \le i \le l$, we compare the two candidate sets of $\mathcal{F}_v^{b_i}$. If $w(F_v^{b_i}) \le w(F_o^{b_i})$, then we set $F^{b_i} = F_v^{b_i}$. Else, $F^{b_i} = F_o^{b_i}$. Clearly, if $\bigcup_{i=1}^{l} F^{b_i}$ contains v, then we can set $F_v^B = \bigcup_{i=1}^{l} F^{b_i}$. Else, compare each $F_v^{b_i} \cup (\bigcup_{k \ne i} F_o^{b_k})$, $1 \le i \le l$, to find a minimum-weight set which can be set as F_v^B.

If we have obtained $\mathcal{F}_v$ for the cut vertex v, then we only retain v and delete the other vertices and all edges of B from G. If the residual graph is not a single vertex, then we can observe that v is shared with two outer edges of a block in the residual graph. Suppose that (u, v) is an outer edge incident on v in a block b'. If $\mathcal{F}_{uv} = \{F_u^{(u,v)}, F_v^{(u,v)}, F_{uv}^{(u,v)}, F_o^{(u,v)}\} = \{\{u\}, \{v\}, \{u,v\}, \emptyset\}$, then we modify the collection of candidate sets associated with (u, v) as follows:

$$\mathcal{F}_{uv} = \{F_u^{(u,v)}, F_v^{(u,v)}, F_{uv}^{(u,v)}, F_o^{(u,v)}\} = \{F_o^B \cup \{u\}, F_v^B, F_v^B \cup \{u\}, F_o^B\}$$

Then, we label (u, v) *changed*. Therefore, we can "add" the collection of candidate sets $\mathcal{F}_v$ of B to an outer edge that has not been labelled *changed* in an block of the residual graph. We can first label all outer edges *unchanged*. If the collection of candidate sets of an outer edge has been modified, we label the edge *changed*.

We now again consider all blocks that each contain a unique cut vertex v' in the residual graph. We repeat the procedure until the residual graph is a single vertex. Notice that in the first step of Algorithm Block, the selected edge (u, u') must be an edge labelled *unchanged* and for each unchanged outer edge (v, w) excluding (u, u'), set $\mathcal{F}_{vw} = \{F_v^{(v,w)}, F_w^{(v,w)}, F_{vw}^{(v,w)}, F_o^{(v,w)}\} = \{\{v\}, \{w\}, \{v,w\}, \emptyset\}$. Below we give the main algorithm.

Algorithm FVS

1. Input an outerplanar graph G. Find all blocks and cut vertices of G. Label each outer edge *unchanged*.

2. If G is a block, invoke algorithm block to output an optimal FVS of G. Else, go to Step 3.

3. Pick blocks containing exactly one cut vertex v. Let $b_1, b_2, \ldots, b_l$ be all of the blocks that each contain the unique cut vertex v. Set $B = \bigcup_{i=1}^{l} b_i$. Invoke algorithm block to compute the collection of candidate sets $\mathcal{F}_v^{b_i} = \{F_v^{b_i}, F_o^{b_i}\}$ for each b_i, $i = 1, 2, \ldots, l$. Then compute the collection of candidate sets $\mathcal{F}_v = \{F_v^B, F_o^B\}$ of B.

4. Delete all edges and all vertices of B except v from G. Redefine G as the residual graph. If G is a single vertex, go to Step 5. Else, select an unchanged outer edge (u, v) incident on v from a block and modify the candidate sets of (u, v) as follows:

$$F_v^{(u,v)} \leftarrow F_v^B,\ F_u^{(u,v)} \leftarrow F_o^B \cup \{u\},\ F_{uv}^{(u,v)} \leftarrow F_v^B \cup \{u\},\ F_o^{(u,v)} \leftarrow F_o^B.$$

Then label (u, v) *changed* and return to Step 3.

5. Output a minimum-weight candidate set from $\mathcal{F}_v$.

Since we can find all blocks' candidate sets in $O(|V|)$ time, the complexity of algorithm FVS is linear.

References:

[1] ARNBORG S, LAGERGREN J, SEESE D. *Easy problem for tree-decomposable graphs* [J]. Journal of Algorithms, 1991, **12**(2): 308–340.

[2] BAFNA V, BERMAN P, FUJITO T. *Constant Ratio Approximation of the Weighted Feedback Vertex Set Problem for Undirected Graphs* [C]. ISAAC'95 Algorthms and Computations, Lecture Notes in Computer Science, Spring, Berlin, 1995, **1004**: 142–151.

[3] BECKER A, GEIGER D. *Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem* [J]. Artificial Intelligence, 1996, **83**: 167–188.

[4] BONDY J A, MURTY U S R. *Graph Theory with Applications* [M]. the Macmillan Press, London, 1976.

[5] CHARTRAND G, HARARY F. *Planar permutation graphs* [J], Ann. Inst. Henri Poincarè, 1967, **3**: 433–438.

[6] GAREY M R, JOHNSON D S. *Computers and Intractability* [M]. W.H. Freeman and Company, New York, 1979.

[7] HARARY F. **Graph Theory** [M]. Addison-Wesley Publishing Company. 1971.

[8] KLEINBERG J, KUMAR A. *Wavelength Conversion in Optical Networks* [C]. Proceedings of the 10th ACM-SIAM symposium on Discrete Algorithm, 1999.

[9] LI De-ming, LIU Yan-pei, *A polynomial algorithm for finding the minimum feedback vertex set of a 3-regular simple graph* [J]. Acta Mathematica Scientia, 1999, **19**(4): 375–381.

一个求外平面图最小顶点赋权反馈点集的线性时间算法

张 少 强[1], 王 骁 力[1,4], 李 国 君[1,3]

(1. 天津师范大学计算机与信息工程学院，天津 300074;

2. 山东大学数学与系统科学学院，山东济南 250100;

3. 中国科学院软件研究所，北京 100080; 4. 南阳师范学院数学系，河南 南阳 473061)

摘 要: 若从一个图中去掉某些顶点后得到的导出子图是无圈图，则所去的那些顶点组成的集合就是原图的反馈点集. 本文主要考虑外平面图中的反馈点集并给出了一个求外平面图最小顶点赋权反馈点集的线性时间算法.

关键词: 外平面图; 反馈点集; 线性时间算法.