SIAM J. COMPUT.
Vol. 23, No. 1, pp. 71–81, February 1994


006



# THE PROFILE MINIMIZATION PROBLEM IN TREES*

DAVID KUO† AND GERARD J. CHANG†

**Abstract.** The profile minimization problem is to find a one-to-one function $f$ from the vertex set $V(G)$ of a graph $G$ to the set of all positive integers such that $\sum_{x\in V(G)}\{f(x) - \min_{y\in N[x]} f(y)\}$ is as small as possible, where $N[x] = \{x\} \cup \{y : y \text{ is adjacent to } x\}$ is the closed neighborhood of $x$ in $G$. This paper gives an $O(n^{1.722})$ time algorithm for the problem in a tree of $n$ vertices.

**Key words.** sparse matrix, profile, labeling, tree, leaf, centroid, basic path, algorithm

**AMS subject classifications.** 05C78, 05C85, 68R10

**1. Introduction.** The profile minimization problem was introduced by [5], [6] as a technique for handling sparse matrices. For instance, in the finite element method [8], [9], we want to solve a system of linear equations $Ax = b$ where $A$ is a sparse symmetric $n \times n$ matrix. Suppose for each row $i$, $a_{ii} \neq 0$ and $p_i$ is the position of the first non-zero element in this row. We call

$$w_i = i - p_i = i - \min\{j : a_{ij} \neq 0\}$$

the *width* of row $i$, and call

$$P(A) = \sum_{i=1}^{n} w_i$$

the *profile* of matrix $A$. To store $A$, we need only store $w_i + 1$ elements in each row $i$, which are from position $p_i$ to position $i$. The total amount of storage for this scheme is then $P(A) + n$. In order to reduce the amount of storage, we need only permute the rows and columns of $A$ simultaneously such that the resulting matrix has minimum profile, i.e., we need to find a permutation matrix $Q$ such that the profile $P(QAQ^t)$ is minimized.

We can reformulate this problem in terms of graphs. Associate the matrix $A$ with a graph $G$ such that $V(G) = \{v_1, v_2, \ldots, v_n\}$ and $E(G) = \{(v_i, v_j) : i \neq j \text{ and } a_{ij} \neq 0\}$. Note that

$$P(A) = \sum_{i=1}^{n} w_i = \sum_{i=1}^{n} \left( i - \min_{v_j \in N[v_i]} j \right),$$

where $N[v_i] \equiv \{v_i\} \cup \{v_j : v_i \text{ is adjacent to } v_j\}$ is the *closed neighborhood* of $v_i$ in $G$. The row and column permutation $Q$ corresponds to a one-to-one function $f$ from $V(G)$ onto $\{1, 2, \ldots, n\}$ and $P(QAQ^t) = \sum_{x\in V(G)}(f(x) - \min_{y\in N[x]} f(y))$. This motivates the definition of the profile of a graph given below.

For technical reasons, however, we shall give a slightly more general definition than that described in the previous paragraph. A *labeling* of a graph $G$ is a one-to-one function $f$ from the vertex set $V(G)$ to the set of all positive integers. A labeling is *simple* if it maps $V(G)$ onto $\{1, 2, \ldots, |V(G)|\}$. For a labeling $f$, the *profile-width* of a vertex $x$ is defined as

$$w_f(x) = f(x) - \min_{y\in N[x]} f(y).$$

*Received by the editors March 11, 1991; accepted for publication (in revised form) July 30, 1992. This work was supported in part by the National Science Council of the Republic of China under grant NSC79-0208-M009-31.

†Institute of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan, Republic of China (gjchang@cc.nctu.edu.tw).

The *profile of G with respect to f* is

$$P_f(G) = \sum_{x \in V} w_f(x)$$

and the *profile* of $G$ is

$$P(G) = \min\{P_f(G) : f \text{ is a labeling of } G\}.$$

A labeling $f$ is *optimal* if $P_f(G) = P(G)$.

The purpose of this paper is to study the *profile minimization problem*, i.e., the problem of determining the profile $P(G)$ of a graph $G$, from an algorithmic point of view. The profile minimization problem is analogous to the *linear arrangement problem*, which is to find a labeling $f$ of a graph $G$ such that $\sum\{|f(x) - f(y)| : (x, y) \text{ is an edge in } G\}$ is minimized (see [1], [3], [7]). Reference [5] proved that the profile minimization problem is equivalent to the problem of interval graph completion, which is known to be NP-complete even when $G$ is stipulated to be an edge graph (see [4]). The main result of this paper is to give an $O(n^{1.722})$ time algorithm for the problem when $G$ is a tree of $n$ vertices.

The rest of this paper is organized as follows. In §2, we establish several basic properties that motivate the development of our algorithm. In particular, we prove that for a tree $T$ there exists a basic path $\alpha(x, y)$ such that $P(T) = P(T - \alpha(x, y)) + |E(T)|$. So the problem becomes that of finding a path $\alpha(x, y)$ such that $P(T - \alpha(x, y))$ is minimized. For the purposes of recurrence, we also introduce the problem of finding a path $\alpha(x, y)$ such that $P(T - \alpha(x, y))$ is minimized, with the boundary condition that $y$ is fixed. In order to determine the basic path, §3 develops theorems that narrow the possibilities for the basic path. For instance, we prove that $\alpha(x, y)$ contains centroids of the tree. This also means that the number of vertices of each component of $T - \alpha(x, y)$ is no more than half the number of vertices of $T$. This is important in determining the speed of our recursive algorithm. Section 4 uses these results to design an algorithm, and §5 analyzes the time complexity of the algorithm.

**2. Motivating properties.** This section shows the existence of a basic path $\alpha(x, y)$ such that $P(T) = P(T - \alpha(x, y)) + |E(T)|$ and introduces the problem of finding a minimum such path with the boundary condition that $y$ is fixed. The following properties are obvious and their proofs are omitted.

PROPOSITION 2.1. *An optimal labeling of a connected graph $G$ maps $V(G)$ onto a set of consecutive integers.*

PROPOSITION 2.2 ([5]). *If $H$ is a subgraph of $G$, then $P(H) \le P(G)$.*

PROPOSITION 2.3 ([5]). *If $G$ has $m$ components $G_1, G_2, \ldots, G_m$, then $P(G) = \sum_{i=1}^m P(G_i)$.*

We can in fact assume that an optimal labeling of a graph is simple even if it is not connected. Suppose $T$ is a tree of $n$ vertices. For any leaf $x$ and any vertex $y$ in $T$, consider the unique $(x, y)$-path $\alpha(x, y) = (v_0, v_1, \ldots, v_r)$, where $v_0 = x$ and $v_r = y$. Suppose that for each $i$, $1 \le i \le r$, $T - \alpha(x, y)$ has $n_i$ components $T_{i1}, T_{i2}, \ldots, T_{in_i}$ each with a vertex $v_{ij}$ adjacent to $v_i$ in $T$ (see Fig. 2.1). Let $f_i$ be an optimal simple labeling of $F_i = \cup_{1 \le j \le n_i} T_{ij}$. We consider a simple labeling $f_{xy}$ defined by

$$f_{xy}(v) = \begin{cases} 1 & \text{if } v = v_0, \\ f_{xy}(v_{i-1}) + f_i(v) & \text{if } v \in V(F_i), \\ f_{xy}(v_{i-1}) + |V(F_i)| + 1 & \text{if } v = v_i. \end{cases}$$

See Fig. 2.2 for an example of $f_{xy}$ with $\alpha(x, y) = (a, b, c, d)$. Note that the numbers beside the vertices are their labels. Then

$$w_{f_{xy}}(v_0) = 0,$$
$$w_{f_{xy}}(v_i) = f_{xy}(v_i) - f_{xy}(v_{i-1}) = |V(F_i)| + 1 \quad \text{for } 1 \le i \le r,$$

and

$$w_{f_{xy}}(v) = w_{f_i}(v) \quad \text{for } v \in V(F_i).$$

Consequently

$$\begin{aligned} P_{f_{xy}}(T) &= \sum_{i=1}^{r}(|V(F_i)| + 1 + P(F_i)) \\ &= |E(T)| + \sum_{i=1}^{r} P(F_i) \qquad (2.1) \\ &= |E(T)| + \sum_{i=1}^{r}\sum_{j=1}^{n_i} P(T_{ij}). \end{aligned}$$

We call $\alpha(x, y)$ the *basic path* (with respect to the labeling $f_{xy}$). Note that

$$1 = f_{xy}(v_0) < f_{xy}(v_1) < \cdots < f_{xy}(v_r) = n.$$

In general, an optimal labeling of a tree is of this type.

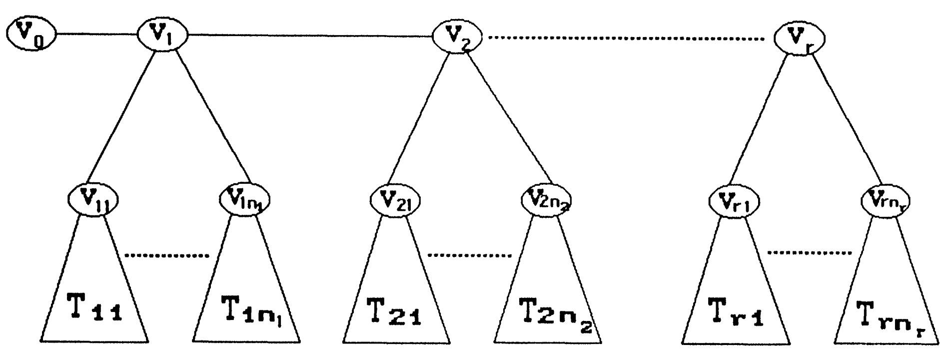


FIG. 2.1. *Tree T.*

THEOREM 2.4. *If $f$ is an optimal labeling of a tree $T$ of $n$ vertices, then $f = f_{xy}$ where $x = f^{-1}(1)$ is a leaf and $y = f^{-1}(n)$ is adjacent to at most one non-leaf vertex.*

*Proof.* Let $\alpha(z, u) = (v_0, v_1, \ldots, v_r)$ be a longest path containing both $x$ and $y$, say, $x = v_s$ and $y = v_t$ for $0 \le s < t \le r$. Note that since $r$ is the maximum, $v_0$ and $v_r$ are leaves. In this case $n_r = 0$ and $P_{f_{zu}}(T) = P_{f_{zu'}}(T)$, where $u' = v_{r-1}$.

Suppose $T$ and $\alpha(z, u)$ are as shown in Fig. 2.1. Let $f_{ij} = f|_{V(T_{ij})}$ be the labeling $f$ restricted on $V(T_{ij})$. Then, by definition,

$$P(T) = P_f(T) \ge \sum_{i=0}^{r} w_f(v_i) + \sum_{i=1}^{r}\sum_{j=1}^{n_i} P_{f_{ij}}(T_{ij}). \qquad (2.2)$$

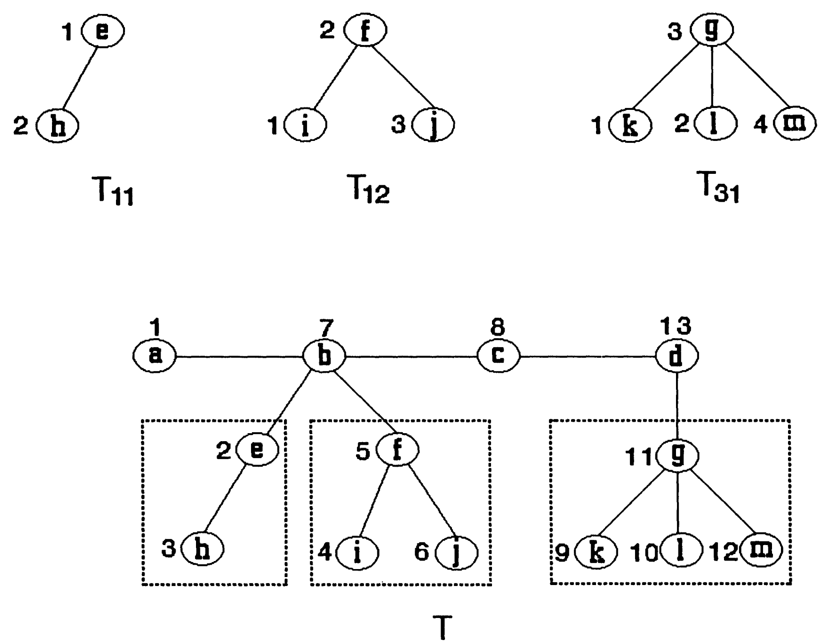


FIG. 2.2. *An example of* $f_{xy}$.

Note that

$$\sum_{i=0}^{r} w_f(v_i) \geq \sum_{i=s+1}^{t} w_f(v_i) \geq \sum_{i=s+1}^{t} \{f(v_i) - f(v_{i-1})\} = n - 1 = |E(T)|. \tag{2.3}$$

Consequently, by (2.1),

$$P(T) \geq |E(T)| + \sum_{i=1}^{r}\sum_{j=1}^{n_i} P(T_{ij}) = P_{f_{zu}}(T) \geq P(T). \tag{2.4}$$

Therefore, all inequalities in (2.2) to (2.4) are equalities. This implies the following:

(1) each $f_{ij}$ is an optimal labeling for $T_{ij}$,

(2) $w_f(v_0) = w_f(v_1) = \cdots = w_f(v_s) = 0$,

(3) $w_f(v_{t+1}) = w_f(v_{t+2}) = \cdots = w_f(v_r) = 0$,

(4) $f(v_{i-1}) = \min_{y \in N[v_i]} f(y)$ for $s + 1 \leq i \leq t$.

Statement (2) implies that $s = 0$, otherwise $w_f(v_{s-1}) = f(v_{s-1}) - f(v_s) > 0$. That is, $x = z$, which is a leaf. Statement (3) implies that $r-1 \leq t$, otherwise either $f(v_{t+1}) > f(v_{t+2})$ or $f(v_{t+1}) < f(v_{t+2})$, i.e., either $w_f(v_{t+1}) > 0$ or $w_f(v_{t+2}) > 0$. So either $y = u$ or $y = u'$. In the former case, $y = u$ is a leaf. In the latter case, $y = u'$ is adjacent to at most one non-leaf vertex, otherwise we can choose a longer $\alpha(z, u)$. In this case, since $P_{f_{zu'}}(T) = P_{f_{zu}}(T)$, we replace $u$ by $u'$ and assume $y = u$. Note that in this case $n_r > 0$. So, now $\alpha(x, y) = \alpha(z, u)$. Finally, statement (4) implies the following:

(5) $1 = f(v_0) < f(v_1) < \cdots < f(v_r) = n$,

(6) $f(v_{i-1}) < f(v_{ij})$ for $1 \leq i \leq r$ and $1 \leq j \leq n_i$.

On the other hand, statement (1) and Proposition 2.1 imply that each $f(V(T_{ij}))$ contains consecutive integers. From this, together with statements (5) and (6), we obtain $f = f_{zu} = f_{xy}$. □

COROLLARY 2.5. *For any tree $T$ there is an optimal labeling $f_{xy}$ in which both $x$ and $y$ are leaves.*

From now on, all optimal labelings we consider are as specified in Corollary 2.5. The path $\alpha(x, y)$ is called a *basic path* for $P(T)$.

Theorem 2.4 and (2.1) tell us that in order to find the profile of a tree $T$ we need only find a basic path $\alpha(x, y)$ whose deletion results in a forest with the smallest possible profile.

For technical reasons, we now consider the following restricted path deletion problem. Suppose $y$ is a fixed vertex in tree $T$; find a path $\alpha(x, y)$ ending at $y$ such that $P(T - \alpha(x, y))$ is minimum. We use $P'(T, y)$ to denote this minimum value. We also call $\alpha(x, y)$ the *basic path* for $P'(T, y)$.

Suppose $f_{xy}$ is an optimal labeling of $T$ and the tree $T$ is as shown in Fig. 2.1. Denote by ${}^kT$ (respectively $T^k$) the subtree of $T$ that contains $v_0, v_1, \ldots, v_k, F_1, \ldots, F_k$ (respectively $v_k, \ldots, v_r, F_k, \ldots, F_r$). From Theorem 2.4 and (2.1), we obtain the following corollary.

COROLLARY 2.6. *For a basic path $(v_0, v_1, \ldots, v_r)$ for $P(T)$, the following hold:*

(1) *$(v_0, v_1, \ldots, v_k)$ is a basic path for $P'({}^kT, v_k)$ and $P'({}^kT, v_k) = \sum_{i=1}^{k} P(F_i) = \sum_{i=1}^{k}\sum_{j=1}^{n_i} P(T_{ij})$ for $1 \le k \le r$.*

(2) *$(v_k, v_{k+1}, \ldots, v_r)$ is a basic path for $P'(T^k, v_k)$ and $P'(T^k, v_k) = \sum_{i=k}^{r} P(F_i) = \sum_{i=k}^{r}\sum_{j=1}^{n_i} P(T_{ij})$ for $1 \le k \le r$.*

(3) *$P(T) = |E(T)| + P'({}^sT, v_s) + P'(T^t, v_t) + \sum_{i=s+1}^{t-1} P(F_i)$ for $1 \le s < t \le r$.*

PROPOSITION 2.7. *$P(T) \le P'(T, y) + |E(T)|$ for any vertex $y$ in $T$.*

**3. Main theorems.** This section develops theorems that restrict the possibilities of the basic paths for $P(T)$ and $P'(T, y)$. In particular, the basic path $\alpha(x, y)$ for $P(T)$ contains the centroids of $T$. We also prove that the basic path for $P'(T, u)$ is either $\alpha(x, u)$ or $\alpha(y, u)$, and the deletion of the basic path for $P'(T, u)$ from $T$ results in a forest each of whose components has at most $2|V(T)|/3$ vertices. These results are the keystone of our algorithm for the profile maximization problem.

A *centroid* of a tree of $n$ vertices is a vertex whose deletion results in a forest each of whose components has at most $\lfloor \frac{n}{2} \rfloor$ vertices. It is well known that a tree has either exactly one centroid or exactly two adjacent centroids (see [2]). A "from leaves to center" method can be employed to derive the centroids of a tree. This method requires linear time.

THEOREM 3.1. *Any basic path $\alpha(x, y)$ for $P(T)$ contains all centroids of $T$.*

*Proof.* Suppose there is a centroid of $T$ not in the basic path

$$\alpha(x, y) = (x, \ldots, v_1, u, v_2, \ldots, y).$$

Then $T$ is of the form shown in Fig. 3.1, with $|V(T')| \ge n/2$ where $n = |V(T)|$. By Corollary 2.6 (3), we have

$$P(T) = |E(T)| + P'(T_1, v_1) + P'(T_2, v_2) + \sum_{i=3}^{k} P(T_i) + P(T'). \tag{3.1}$$

Up to a symmetric argument, we may assume that $|V(T_1)| \le |V(T_2)|$. Let $\alpha(z, v)$ be a basic path for $P(T')$. Corollary 2.6 (3) and Proposition 2.2 give

$$P(T') \ge |E(T')| + P'(T_a, a) + P'(T_b, b) + \sum_{j=1}^{m} P(F_j). \tag{3.2}$$

We also assume that $|V(T_a)| \le |V(T_b)|$. Now consider the labeling $f_{vy}$ for $T$. By (2.1) and Corollary 2.6, we have

(3.3)

$$P_{f_{vy}}(T) = |E(T)| + P'(T_b, b) + P'(T_2, v_2) + P(T_a) + \sum_{j=1}^{m} P(F_j) + P(T_1) + \sum_{i=3}^{k} P(T_i).$$

Equations (3.1) to (3.3) together lead to that $|E(T')| \leq P(T_a) - P'(T_a, a) + P(T_1) - P'(T_1, v_1)$. Then $|E(T')| \leq |E(T_a)| + |E(T_1)|$ by Proposition 2.7. Thus

$$\begin{aligned} |E(T_1)| &\geq |E(T')| - |E(T_a)| \\ &> |E(T')|/2 \qquad (\text{since } |E(T_a)| \leq |E(T_b)| \text{ and } T' - (T_a \cup T_b) \neq \emptyset) \\ &\geq (n - |E(T')|)/2 \qquad (\text{since } |E(T')| \geq n/2) \\ &\geq |E(T_1)| \qquad (\text{since } |E(T_1)| \leq |E(T_2)|), \end{aligned}$$

which is a contradiction. □

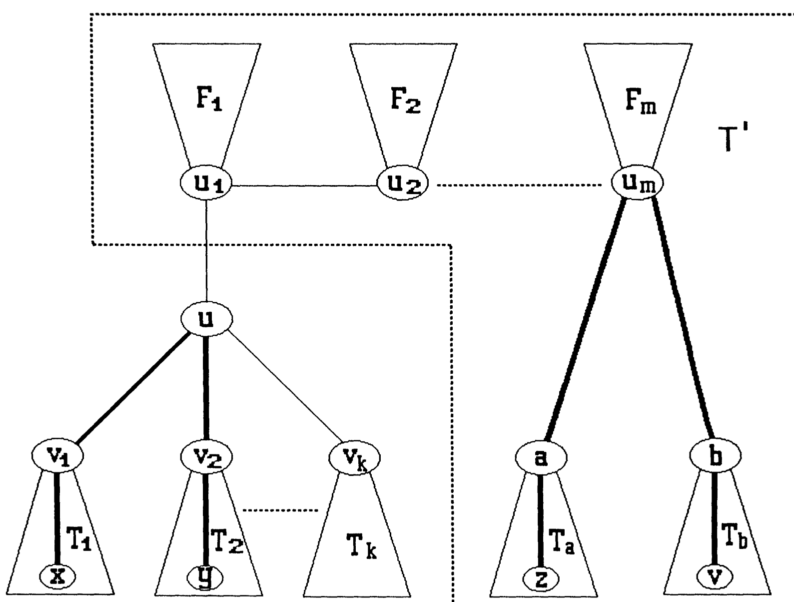


FIG. 3.1.

Similar arguments lead to the following theorem.

THEOREM 3.2. *Suppose $y$ is a fixed vertex of a tree $T$ of $n$ vertices. For any basic path $\alpha(xy)$ of $P'(T, y)$, every component of $T - \alpha(x, y)$ has at most $2n/3$ vertices.*

*Proof.* The proof of this theorem is exactly the same as that for Theorem 3.1, except now we assume $|E(T')| > 2n/3$ and $|E(T_a)| \leq |E(T_b)|$, and there is no assumption that $|E(T_1)| \leq |E(T_2)|$. However, we still have $|E(T')| \leq |E(T_a)| + |E(T_1)|$. Then

$$\begin{aligned} |E(T_1)| &\geq |E(T')| - |E(T_a)| \\ &\geq |E(T')|/2 \qquad (\text{since } |E(T_a)| \leq |E(T_b)|) \\ &> n - |E(T')| \qquad (\text{since } |E(T')| > 2n/3) \\ &\geq |E(T_1)|, \end{aligned}$$

which is a contradiction. □

Figure 3.2 gives an example in which a basic path $\alpha(x, y)$ for $P'(T, y)$ does not contain the centroid $z$ of $T$.

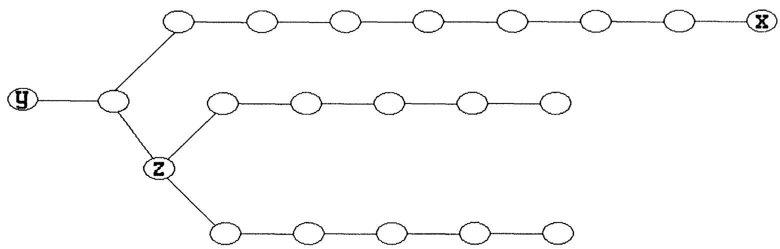


FIG. 3.2.

THEOREM 3.3. *If* $\alpha(x, y)$ *is a basic path for* $P(T)$ *and* $u$ *is a fixed vertex in* $T$, *then either* $\alpha(x, u)$ *or* $\alpha(y, u)$ *is a basic path for* $P'(T, u)$.

*Proof.* Suppose $\alpha(x, y) = (x, \ldots, v_1, u_1, v_2, \ldots, y)$ and $(u_1, u_2, \ldots, u_r = u)$ is the unique path from $\alpha(x, y)$ to $u$, as shown in Fig. 3.3. Let $\alpha(z, u)$ be a basic path for $P'(T, u)$.

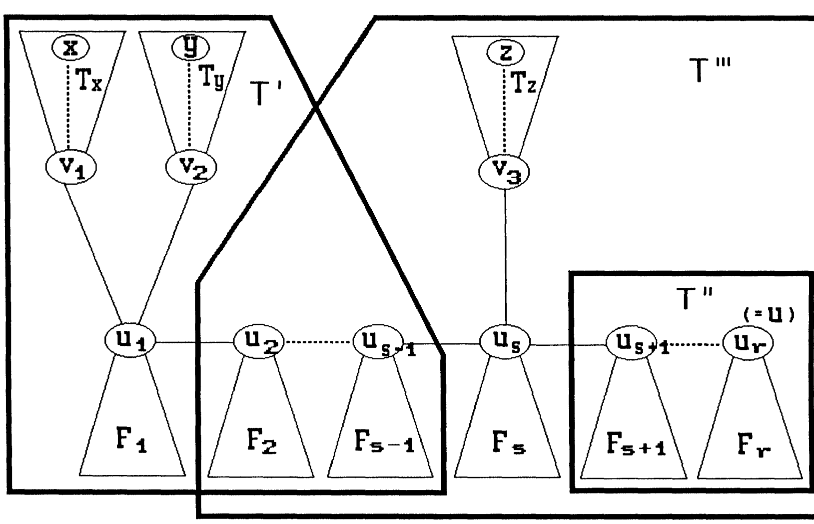


FIG. 3.3.

*Case* 1. $z \in V(T_x)$. In this case, $u_s = u_1$, $v_3 = v_1$, and $T_z = T_x$. By Corollary 2.6 (1), $\alpha(x, v_1)$ is a basic path for $P'(T_x, v_1)$, and so $P(T_x - \alpha(x, v_1)) \leq P(T_z - \alpha(z, v_3))$. Then

$$\begin{aligned} P'(T, u) &= P(T - \alpha(z, u)) \\ &= P(T_z - \alpha(z, v_3)) + P(T_y) + \sum_{i=1}^{r} P(F_i) \\ &\geq P(T_x - \alpha(x, v_1)) + P(T_y) + \sum_{i=1}^{r} P(F_i) \\ &= P(T - \alpha(x, u)). \end{aligned}$$

Hence $\alpha(x, u)$ is also a basic path for $P'(T, u)$.

*Case* 2. $z \in V(T_y)$. By a similar argument, $\alpha(y, u)$ is also a basic path for $P'(T, u)$.

*Case* 3. $z \notin V(T_x)$ and $z \notin T(T_y)$. Let $T'$, $T''$, and $T'''$ be subtrees, as shown in Fig. 3.3. Note that in the case of $s = 1$, $T' = T_x \cup T_y$ is not a tree. Now

$$P(T - \alpha(z, u)) = P'(T_z, v_3) + P(T') + \sum_{i=s}^{r} P(F_i). \tag{3.4}$$

Note that $P'(T_z, v_3) = P(T_z - \alpha(z, v_3))$. By Proposition 2.2, we have

$$P(T') \geq P(T_x) + P(T_y) + \sum_{i=1}^{s-1} P(F_i). \tag{3.5}$$

Since $\alpha(x, y)$ is a basic path for $P(T)$, we have $P_{f_{xz}}(T) \geq P_{f_{xy}}(T)$. By (2.1) and Corollary 2.6 (3) we have

$$\begin{aligned} &|E(T)| + P(T_x - \alpha(x, v_1)) + P(T_y) + \sum_{i=1}^{s} P(F_i) + P(T'') + P(T_z - \alpha(z, v_3)) \\ &\geq |E(T)| + P'(T_x, v_1) + P'(T_y, v_2) + P(F_1) + P(T'''). \end{aligned} \tag{3.6}$$

Note that $P'(T_x, v_1) = P(T_x - \alpha(x, v_1))$. Again, by Proposition 2.2,

$$P(T''') \geq \sum_{i=2}^{s} P(F_i) + P(T_z) + P(T''). \tag{3.7}$$

Equations (3.4) to (3.7) together lead to

$$P(T - \alpha(z, u)) \geq P'(T_y, v_2) + P(T_x) + \sum_{i=1}^{r} P(F_i) + P(T_z) = P(T - \alpha(y, u)).$$

Hence $\alpha(y, u)$ is a basic path for $P'(T, u)$. □

**4. The algorithm.** We can use the theorems in §3 to design an efficient algorithm for the profile minimization problem in a tree $T$. By Theorem 3.1, the basic idea of our algorithm is to find a centroid $z$ first in linear time. Suppose $T - z = \cup_{1 \leq i \leq m} T_i$, where $u_i$ is the only vertex of $T_i$ that is adjacent to $z$ in $T$ (see Fig. 4.1). To use Corollary 2.6 (3), we need to find all

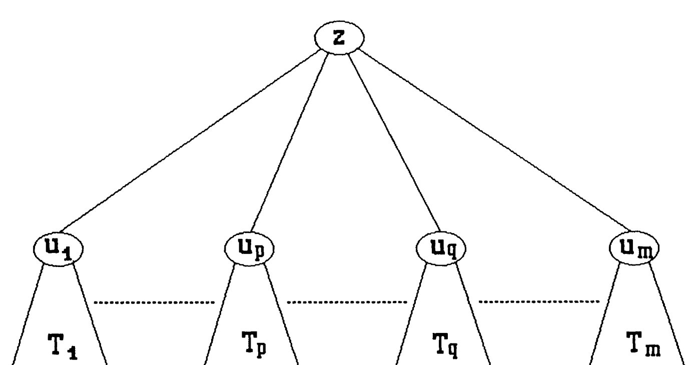


FIG. 4.1.

profiles $P(T_i)$ and $P'(T_i, u_i)$ recursively. In the following, Algorithm PROFILE finds $P(T)$ and Algorithm PROFILE1 finds $P'(T, u)$. Note that, in order to make use of Theorem 3.3, Algorithm PROFILE not only has to output the value $P(T)$ but also a basic path.

ALGORITHM PROFILE
*Input*: A tree $T$ of $n$ vertices.
*Output*: A basic path $\alpha(x, y) = (v_0, v_1, \ldots, v_r)$ for $P(T)$ and the values $P(T)$ and $P(T_{ij})$ for $1 \leq i \leq r-1$ and $1 \leq j \leq n_i$.
*Method*:

1. find a centroid $z$ of $T$.
2. let $T - z = \cup_{1\leq k\leq m} T_k$ and $z$ be adjacent to $u_k \in V(T_k)$ for $1 \leq k \leq m$.
3. for each $1 \leq k \leq m$, recursively call PROFILE for $T_k$ to get a basic path $\alpha(x_k, y_k)$ and values $P(T_k)$ and $P(T_{kij})$, where $T_{kij}$ are the components of $T_k - \alpha(x_k, y_k)$.
4. for each $1 \leq k \leq m$, recursively call PROFILE1 for $(T_k, u_k)$ to get a basic path $\alpha(z_k, u_k)$ and values $P'(T_k, u_k)$ and $P(T'_{kij})$, where $T'_{kij}$ are the components of $T_k - \alpha(z_k, u_k)$.
5. let $P(T) = n + \min_{1\leq p<q\leq m}\{P'(T_p, u_p) + P'(T_q, u_q) + \sum_{i\neq p,q}(T_i)\}$, where $p^*$ and $q^*$ attain the above minimum.
6. let $\alpha(x, y) = \alpha(z_{p^*}, u_{p^*}) + z + \alpha(u_{q^*}, z_{q^*})$.
7. combine profiles $P(T_{p^*ij})$, $P(T_k)$ for $k \neq p^*, q^*$, and $P(T_{q^*ij})$ to get profiles $P(T_{ij})$.

To find $P'(T, u)$, we note that by Theorem 3.3, either $\alpha(x, u)$ or $\alpha(y, u)$ is a basic path for $P'(T, u)$. So we consider the configuration in Fig. 3.3 with $T_z$ omitted.

ALGORITHM PROFILE1
*Input*: Tree $T$ of $n$ vertices with a basic path $\alpha(x, y) = (v_0, v_1, \ldots, v_r)$ for $P(T)$ and the values $P(T_{ij})$ for $1 \leq i \leq r-1$ and $1 \leq j \leq n_i$. $u$ is a fixed vertex in $T$.
*Output*: A basic path $\alpha(z, u) = (v'_0, v'_1, \ldots, v'_{r'})$ for $P'(T, u)$ and the values $P'(T, u)$ and $P(T'_{ij})$ for $1 \leq i \leq r'$ and $1 \leq j \leq n'_i$.
*Method*:

1. identify the path $(u_1, u_2, \ldots, u_r)$ as in Fig. 3.3.
2. recursively use PROFILE to solve $P(T_x)$, $P(T_y)$, $P(F_i)$ (in fact $P(T_{ij})$ for each component in $F_i$) for $1 \leq i \leq r$.
3. $a = P'(T_x, v_1) + P(T_y) + \sum_{i=1}^{r} P(F_i)$,
   $b = P'(T_y, v_2) + P(T_x) + \sum_{i=1}^{r} P(F_i)$,
   where $P'(T_x, v_1)$ and $P'(T_y, v_2)$ can be computed from the input values $P(T_{ij})$.
4. $P'(T, u) = \min\{a, b\}$.
   if $a \leq b$ then $z = x$ else $z = y$.
5. combine part of the profiles $P(T_{ij})$, $P(T_x)$, or $P(T_y)$, and $P(F_i)$ to get profiles $P(T'_{ij})$.

**5. Time complexity.** This section shows that the time complexities of the above two algorithms are $O(n^{1.722})$. Let $f(n)$ (respectively, $g(n)$) be the time complexity for Algorithm PROFILE (respectively, PROFILE1).

In Algorithm PROFILE, Step 3 (respectively, 4) needs $\sum_{i=1}^{m} f(n_i)$ (respectively, $\sum_{i=1}^{m} g(n_i)$) time, where $n_i = |V(T_i)|$ for $1 \leq i \leq m$. All other steps need $O(n)$ time. Note that for Step 5 we only have to find the smallest and the second smallest values of $P'(T_i, w_i) - P(T_i)$. Therefore

$$f(n) = \sum_{i=1}^{m}\{f(n_i) + g(n_i)\} + c_1 n, \tag{5.1}$$

where

$$\sum_{i=1}^{m} n_i = n - 1 \quad \text{and} \quad n_i \le n/2 \quad \text{for} \quad 1 \le i \le m.$$

Similarly, in Algorithm PROFILE1, Step 2 needs $\sum_{i=1}^{m} f(n_i)$ time and all other steps need $O(n)$ time. Thus, by Theorem 3.2,

$$g(n) = \sum_{i=1}^{m} f(n_i) + c_2 n, \tag{5.2}$$

where

$$\sum_{i=1}^{m} n_i \le n - 1 \quad \text{and} \quad n_i \le 2n/3 \quad \text{for} \quad 1 \le i \le m.$$

To solve (5.1) and (5.2), we first choose a number $\sigma > 1$, which is very close to 1, say, $\sigma = 1.001$. Then choose $1 < \lambda < 2$ such that

$$\epsilon = \left(\frac{2}{3}\right)^{\lambda} + \left(\frac{1}{3}\right)^{\lambda} < 1$$

and

$$\delta = (1 + \sigma\epsilon) 2 \left(\frac{1}{2}\right)^{\lambda} < 1.$$

Note that a simple computer program gives that $\lambda = 1.722$ for $\sigma = 1.001$.

THEOREM 5.1. *There exists a constant c such that $f(n) \le cn^{\lambda}$ and $g(n) \le c\sigma\epsilon n^{\lambda}$ for all n, i.e., $f(n) = O(n^{\lambda})$ and $g(n) = O(n^{\lambda})$.*

*Proof.* The proof is by induction on $n$. Assume that there exists a constant $c$ such that $f(n') \le cn'^{\lambda}$ and $g(n') \le c\sigma\epsilon n'^{\lambda}$ for all $n' < n$. We also assume that $c \ge c_1/(1-\delta)$ and $c \ge c_2/(\sigma - 1)\epsilon$.

For $0 < a \le b$, consider the function $h(x) = (b+x)^{\lambda} + (a-x)^{\lambda} - b^{\lambda} - a^{\lambda}$ where $0 \le x \le a$. Note that $h'(x) = \lambda(b+x)^{\lambda-1} - \lambda(a-x)^{\lambda-1} \ge 0$. So $h$ is an increasing function and then $h(x) \ge h(0) = 0$ for $0 \le x \le a$. Thus

$$b^{\lambda} + a^{\lambda} \le (b+x)^{\lambda} + (a-x)^{\lambda} \quad \text{for} \quad 0 \le x \le a \le b. \tag{5.3}$$

By (5.1) and the induction hypothesis, we have $f(n) \le c(1+\sigma\epsilon)\sum_{i=1}^{m} n_i^{\lambda} + c_1 n$ where $\sum_{i=1}^{m} n_i = n-1$ and $n_i \le n/2$ for $1 \le i \le m$. Repeatedly apply (5.3) to get $\sum_{i=1}^{m} n_i^{\lambda} \le \left(\frac{n}{2}\right)^{\lambda} + \left(\frac{n}{2} - 1\right)^{\lambda}$. Therefore $f(n) \le c(1+\sigma\epsilon)2\left(\frac{1}{2}\right)^{\lambda} n^{\lambda} + c_1 n = c\delta n^{\lambda} + c_1 n$. By the choice of $c$, $c_1 n \le c(1-\delta)n \le c(1-\delta)n^{\lambda}$. Then $f(n) \le cn^{\lambda}$.

By (5.2) and the induction hypothesis, we have $g(n) \le c\sum_{i=1}^{m} n_i^{\lambda} c_2 n$ where $\sum_{i=1}^{m} n_i \le n-1$ and $n_i \le 2n/3$ for $1 \le i \le m$. Repeatedly apply (5.3) to get $\sum_{i=1}^{m} n_i^{\lambda} \le (2n/3)^{\lambda} + \left(\frac{n}{3}\right)^{\lambda} = \epsilon n^{\lambda}$. Also, by the choice of $c$, $c_2 n \le c(\sigma-1)\epsilon n \le c(\sigma-1)\epsilon n^{\lambda}$. Then $g(n) \le c\epsilon n^{\lambda} + c(\sigma-1)\epsilon n^{\lambda} = c\sigma\epsilon n^{\lambda}$. ◻

**Acknowledgments.** We thank the two anonymous referees for many useful suggestions on the revision of this paper.

## REFERENCES

[1] D. Adolphson and T. C. Hu, *Optimal linear ordering*, SIAM J. Appl. Math., 25 (1973), pp. 403–423.

[2] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley, Reading, MA, 1990.

[3] F. R. K. Chung, *On optimal linear arrangements of trees*, Comput. Math. Appl., 10 (1984), pp. 43–60.

[4] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[5] Y. Lin and J. Yuan, *Profile minimization problem for matrices and graphs*, preprint, Dept. of Math., Zhengzhou University, People's Republic of China, 1990.

[6] ———, *Minimum profile of grid networks in structure analysis*, to appear.

[7] Y. Shiloach, *A minimum linear arrangement algorithm for undirected tree*, SIAM J. Comp., 8 (1979), pp. 15–32.

[8] R. P. Tewarson, *Sparse Matrices*, Academic Press, New York, 1973.

[9] O. C. Zienkiewicz, *Finite Element Method in Engineering Science*, McGraw Hill, London, 1971.