Discussiones Mathematicae
Graph Theory 26 (2006) 5–18

ALGORITHMIC ASPECTS OF TOTAL *k*-SUBDOMINATION IN GRAPHS

[1]Laura M. Harris, [2]Johannes H. Hattingh

and

[1]Michael A. Henning *

[1]*School of Mathematical Sciences*
University of KwaZulu-Natal
Pietermaritzburg, 3209 South Africa

[2]*Department of Mathematics and Statistics*
Georgia State University
Atlanta, GA 30303–3083 USA

e-mail: jhhattingh@gsu.edu

Abstract

Let $G = (V, E)$ be a graph and let $k \in \mathbf{Z}^+$. A total k-subdominating function is a function $f\colon V \to \{-1, 1\}$ such that for at least k vertices v of G, the sum of the function values of f in the open neighborhood of v is positive. The total k-subdomination number of G is the minimum value of $f(V)$ over all total k-subdominating functions f of G where $f(V)$ denotes the sum of the function values assigned to the vertices under f. In this paper, we present a cubic time algorithm to compute the total k-subdomination number of a tree and also show that the associated decision problem is **NP**-complete for general graphs.

Keywords: total k-subdomination, algorithm, tree.

2000 Mathematics Subject Classification: 05C69.

---

*Research supported in part by the University of Natal and the National Research Foundation.

1. Introduction

Our mathematical model is a finite, simple graph $G = (V, E)$ with vertex set V and edge set E of order $n(G) = |V|$ and size $m(G) = |E|$.

A set S is a *dominating set* of G if every vertex of $V - S$ is adjacent to some vertex of S. The *domination number*, denoted by $\gamma(G)$, is defined as the minimum cardinality of a dominating set of G.

If every vertex of a graph is adjacent to some vertex of a set S, then S is called a *total dominating set* of G. For $\delta(G) \geq 1$, the *total domination number*, denoted by $\gamma_t(G)$, is defined as the minimum cardinality of a total dominating set of G. If $\delta(G) \geq 3$, then, by a result of Lam and Wei [17], $\gamma_t(G) \leq \frac{|V(G)|}{2}$.

The *open neighborhood* of a vertex v is $N(v) = \{u \mid uv \in E\}$. The *closed neighborhood* of v is $N[v] = N(v) \cup \{v\}$. An *opinion function* on G is a function $f\colon V \to \{-1, +1\}$; $f(v)$ is the *opinion* of the vertex v. The *weight* $w(f)$ of an opinion function f is the sum of its values, i.e., $w(f) = \sum_{v \in V} f(v)$, and for $S \subseteq V$ we define $f(S) = \sum_{v \in S} f(v)$, so $w(f) = f(V)$. For a vertex v in V, we denote $f(N(v))$ by $f[v]$ for notational convenience.

For a positive integer k, Cockayne and Mynhardt [4] define a *k-subdominating function* of G as an opinion function such that the sum of the function values, taken over closed neighborhoods of vertices, is at least one for at least k vertices of G. The minimum weight of such a function is defined as the *k-subdomination number* of G and denoted by $\gamma_{ks}(G)$ (studied in [3, 4] and elsewhere). In the special case where $k = |V|$, we have the *signed domination number* which is studied in [5, 6, 7, 11, 12, 13, 16, 18, 21, 23] and elsewhere. When $k = \lceil |V|/2 \rceil$, we have the *weak majority number* (also called the majority domination number) studied in [1, 2, 19] and elsewhere. When $k = \lceil (|V|+1)/2 \rceil$, we have the *strict majority function* studied in [15] and elsewhere.

An analogous theory for total k-subdominating functions that arise when "closed" neighborhood in the definition of a k-subdominating function is changed to "open" neighborhood was introduced in [10]. Here, the *vote* of a vertex v is defined as the sum of the opinions in $N(v)$, the open neighborhood of v. When the vote is positive, we say that v votes *aye*; otherwise, v votes *nay*. A *total k-subdominating function* of a graph G is an opinion function for which at least k of the vertices vote aye. The *weight* of an opinion function is the sum of its values. The *total k-subdomination number* of G, denoted by $\gamma_{tks}(G)$, is the minimum weight of a total k-subdominating function of G. The weight of a total k-subdominating function is small when, in our original scenario, the number of individuals with positive opinions needed to produce a global positive decision is small. If f is a total k-subdominating function of a graph G, we let $C_f(G) = \{v \in V \mid f[v] \geq 1\}$, and when the graph G is clear from context, we denote $C_f(G)$ simply by C_f. In the special case where $k = |V|$, the total k-subdomination number is the *signed total domination number* $\gamma_t^s(G)$ which is studied in [9, 14, 22]. Specifically, a linear time algorithm to compute the total signed domination number of a tree appears in [9].

The motivation for studying the total k-subdomination number is rich and varied from a modeling perspective. For example, by assigning the values -1 or $+1$ to the vertices of a graph we can model networks of people or organizations in which global decisions must be made (e.g. positive or negative responses or preferences). We assume that each individual has one vote and that each individual has an initial opinion. We assign $+1$ to vertices (individuals) which have a positive opinion and -1 to vertices which have a negative opinion. We also assume, however, that an individual's vote is affected by the opinions of neighboring individuals. In particular, each individual gives equal weight to the opinions of neighboring individuals (thus individuals of high degree have greater "influence"). A voter votes 'aye' if there are more vertices in its (open) neighborhood with positive opinion than with negative opinion, otherwise the vote is 'nay'. We seek an assignment of opinions that guarantee at least k vertices voting aye. We call such an assignment of opinions a k-positive assignment. Among all k-positive assignments of opinions, we are interested primarily in the minimum number of vertices (individuals) who have a positive opinion. The total k-subdomination number is the minimum possible sum of all opinions, -1 for a negative opinion and $+1$ for a positive opinion, in a k-positive assignment of opinions. The total k-subdomination number represents, therefore, the minimum number of individuals which can have positive opinions and in doing so force at least k individuals to vote aye.

In this paper, we present a cubic time algorithm to compute the total k-subdomination number of a tree and also show that the decision problem

TOTAL SUBDOMINATING FUNCTION (TSF)

INSTANCE: A graph $G = (V, E)$, positive integers c, d such that $\gcd(c, d) = 1$ and $0 < \frac{c}{d} \leq 1$ and an integer t.

QUESTION: Is there a total subdominating function f such that $f(V) \leq t$ and $|C_f| \geq \lceil \frac{c|V|}{d} \rceil$?
is **NP**-complete by describing a polynomial transformation from the following problem:

TOTAL DOMINATING SET, RESTRICTED TO 4-REGULAR GRAPHS (TDS)

INSTANCE: A 4-regular graph $G = (V, E)$ and a positive integer $k \leq \frac{|V|}{2}$.

QUESTION: Is there a total dominating set of cardinality k or less for G?

2. Complexity Result

In this section we will show that **TSF** is **NP**-complete by describing a polynomial transformation from **TDS**.

We first show that **TDS** is **NP**-complete by describing a polynomial transformation from the decision problem **DOMINATING SET**.

DOMINATING SET, RESTRICTED TO PLANAR CUBIC GRAPHS (DS)

INSTANCE: A planar cubic graph $G = (V, E)$ and a positive integer $k \leq \frac{|V|}{2}$.

QUESTION: Is there a total dominating set of cardinality k or less for G?

Starting with the graph G, take two copies of the vertex set of G (which will be independent sets), and join a vertex to all vertices in the other copy that are in its closed neighborhood in G. The resulting graph has total domination number equal to twice the domination number of G. This construction transforms a cubic graph into a 4-regular graph. Since **DS** is **NP**-complete [8], **TDS** is **NP**-complete.

If $\frac{c}{d} = 1$, then **TSF** is the **NP**-complete problem **TOTAL SIGNED DOMINATION** (see [9]). Hence, we also assume that $0 < \frac{c}{d} < 1$. For convenience, we set $q = \frac{c}{d}$, and denote $\min\{f(V(G)) \mid f$ is a total subdominating function with $|C_f| \geq \lceil q|V(G)\rceil$ by $\gamma_q(G)$.

We will need the following lemma.

Lemma 1. *If c, d, p are positive integers such that $0 < q = \frac{c}{d} < 1$, then there exist positive integers ℓ and r such that $8 \leq \ell \leq d^2(\lceil \frac{p}{2} \rceil + 4)$, $r < d^2(\lceil \frac{p}{2} \rceil + 4)$ and $q = \frac{p+r}{2p+r+\ell}$.*

Proof. Since $c < d$, we have $c \geq 1$, $d \geq 2$ and $d - c \geq 1$. Let $t = \lceil \frac{p}{2} \rceil + 4$. Then $dt(d-c) \geq 2t$ and $cdt \geq 2t$. However, $2t \geq p + 8$, whence $dt(d-c) \geq p+8$ and $cdt > p$. Let t be the smallest positive integer such that $dt(d-c) \geq p + 8$ and $cdt > p$. It follows that $t \leq \lceil \frac{p}{2} \rceil + 4$. Let $r = cdt - p$ and $\ell = ddt - cdt - p$. Note that r and ℓ are both positive integers such that $r, \ell < ddt \leq d^2(\lceil \frac{p}{2} \rceil + 4)$. Furthermore, $\ell \geq 8$ and $q = \frac{p+r}{2p+r+\ell}$. ■

Theorem 1. *The decision problem* **TSF** *is* **NP***-complete.*

Proof. Obviously, **TSF** is in **NP**.

Let G be a 4-regular graph, $p = n(G)$ and k be an integer such that $k \leq p/2$. By Lemma 1, there exists positive integers r, ℓ such that $\ell \geq 8$ and $q = \frac{p+r}{2p+r+\ell}$. Let H be the graph constructed from G as follows: Take a complete graph F on $p + \ell$ vertices, a fixed subset $U \subseteq V(F)$ with $|U| = 3$ and an empty graph L on r vertices, and let H be obtained from the disjoint union of F, G, and L by joining each vertex of U to every vertex in $V(G) \cup V(L)$. Since $n(H) = 2p + r + \ell < 2(p + d^2(\lceil \frac{p}{2} \rceil + 4))$, the graph H can be constructed from G in polynomial time.

We start by showing that if S is a total dominating set of G of cardinality at most k, then there is a total subdominating function f of H of weight at most $2k - 2p - r - \ell + 6$ such that $|C_f| \geq qn(H)$. Define $f : V(H) \to \{-1, 1\}$ by $f(v) = 1$ if $v \in S \cup U$, while $f(v) = -1$ otherwise.

Let $v \in V(G)$. Since S is a total dominating set of G, v is adjacent to some vertex $u \in S$ for which $f(u) = 1$. Since G is 4-regular and $f(U) = 3$, we have $f[v] \geq 1$. It is clear that $f[w] = 3$ for each vertex $w \in V(L)$, so that $f[v] \geq 1$ for at least $p + r = q(2p + r + \ell) = qn(H)$ vertices. This shows that f is a total subdominating function of H of weight $2|S| - 2p - r - \ell + 6 \leq 2k - 2p - r - \ell + 6$.

For the converse, assume that $\gamma_q(H) \leq 2k - 2p - r - \ell + 6$. Among all the minimum total subdominating functions of H, let f be one that assigns the value $+1$ to as many vertices of U as possible. Let P and M be the sets of vertices in H that are assigned the values $+1$ and -1, respectively, under f. Then $|P| + |M| = 2p + r + \ell$, and $|P| - |M| = \gamma_q(H)$. Before proceeding further we prove three claims.

Claim 1. $|P| \leq k + 3$.

Proof. Suppose $|P| \geq k + 4$. Then $|M| \leq 2p + r + \ell - k - 4$, so that $\gamma_q(H) = |P| - |M| \geq 2k - 2p - r - \ell + 8$, which contradicts the fact that $\gamma_q(H) \leq 2k - 2p - r - \ell + 6$. ♢

Claim 2. $f[v] \le 0$ for all $v \in V(F)$.

Proof. Suppose there exists a $v \in V(F)$ such that $f[v] \ge 1$. If $v \in U$, then, since v dominates H, it follows that $0 = 1 - 1 \le f[v] + f(v) = f(V(H)) = \gamma_q(H) \le 2k - 2p - r - \ell + 6$, whence $k \le \frac{p}{2} \le p + \frac{r}{2} < k$, which is a contradiction. Hence $v \in V(F) - U$. Since $N(v) = V(F) - \{v\}$ and $f[v] \ge 1$, it follows that more than half of the vertices of the set $V(F) - \{v\}$ have the value 1 assigned to them under f. This implies that $|P| \ge \frac{p+\ell}{2} = \frac{p}{2} + \frac{\ell}{2} \ge \frac{p}{2} + 4$. By Claim 1 and the fact that $k \le \frac{p}{2}$, it follows that $|P| \le \frac{p}{2} + 3$, which is a contradiction. $\diamondsuit$

As $f[v] \ge 1$ for at least $qn(H) = q(2p+r+\ell) = p+r$ vertices, and $f[v] \le 0$ for all $v \in V(F)$ (cf. Claim 2), it follows that $f[v] \ge 1$ for all $v \in V(G) \cup V(L)$.

Claim 3. $f(U) = 3$.

Proof. Suppose that $f(u) = -1$ for some $u \in U$. If $f(v) = -1$ for all $v \in V(G)$, then $f[v] \le -3$ for all $v \in V(G)$, which is a contradiction. It follows that there exists a $v \in V(G)$ such that $f(v) = 1$. Define $g : V(H) \to \{-1, 1\}$ by $g(w) = f(w)$ if $w \in V(H) - \{u, v\}$, $g(v) = -1$ and $g(u) = 1$, and consider a vertex $x \in V(G) \cup V(L)$. Note that if $x \notin N(v)$ or $x = v$, then $g[x] = f[x] + 2$, while if $x \in N(v)$, then $g[x] = f[x]$. It follows that $g[v] \ge 1$ for at least q of the vertices of H while the weights of g and f are equal. Hence g is a total subdominating function of H of weight $\gamma_q(H)$ that assigns the value $+1$ to more vertices of U than does f, contradicting our choice of f. $\diamondsuit$

Let $S = P \cap V(G)$. Since $f[v] \ge 1$ for all $v \in V(G)$, it follows that every $v \in V(G)$ is adjacent to some vertex in S, which shows that S is a total dominating set of G. Since $f(U) = 3$, Claim 1 implies that $|S| \le k$, which completes the proof. ■

3. Computing $\gamma_{tks}(T)$ for a Tree T

In this section, we will present a cubic time algorithm to compute the total signed k-subdomination number of a tree.

The tree T will be rooted and represented by the resulting parent array **parent**$[1 \ldots n]$. We make use of the well-known fact that the tree T can be constructed recursively from the single vertex K_1 using only one rule of composition, which combines two trees (G, x) and (H, y), by adding an edge between x and y and calling x the root of the larger tree F. We express this as follows: $(F, x) = (G, x) \circ (H, y)$. With each such subtree (F, x), we associate the following data structure:

1. **record**$[x]$**.numvertices:** the number of vertices in the subtree (F, x).
2. **record**$[x]$**.degree:** $\deg_F(x)$.
3. **record**$[x]$**.sum**$[f(x), t, k]$: the minimum weight of a function $f : V(F) \to \{-1, 1\}$ such that x is assigned $f(x)$, $|t| \leq \deg_T(x) - \deg_F(x)$ (representing all possible sums of assignments of -1 and $+1$ to the vertices of $N_T(x) - N_F(x)$ and $|\{v \mid f(N_F(v)) + t \geq 1$ when $v = x$ and $f(N_F(v)) \geq 1$ when $v \neq x\}| \geq k$, where $1 \leq k \leq$ **record**$[x]$**.numvertices**.

Our input consist of the order of the tree T, say **n**, and the **parent** array of the tree, rooted at a certain vertex. The root of the tree T is labeled with 1, the vertices on the next level are labeled with 2 through 2 plus the number of vertices on level 2, and so on. Using the **parent** array, we compute $\deg_T(x)$ for each vertex x, $x = 1, \ldots, n$. We then initialize the variable **record**$[x]$ for each vertex x, where $x = 1, \ldots, n$. Let x be an arbitrary vertex of T. Initially, $(F, x) = (K_1, x)$, whence **record**$[x]$**.numvertices**=1 and **record**$[x]$**.degree**=0. Suppose t is an integer such that $|t| \leq \deg_T(x) - \deg_F(x) = \deg_T(x)$, representing all possible sums of assignments of -1 and $+1$ to the vertices of $N_T(x) - N_F(x) = N_T(x)$. Then $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, \deg_T(x)\}$. The only way for $f(N_F(x)) + t = t \geq 1$, is for $t \geq 1$ if $\deg_T(x)$ is odd and for $t \geq 2$ if $\deg_T(x)$ is even. Thus, we have the following initializations:

Case 1. $\deg_T(x)$ is odd and $t \in \{1, 3, \ldots, \deg_T(x)\}$ or $\deg_T(x)$ is even and $t \in \{2, 4, \ldots, \deg_T(x)\}$. Then **record**$[x]$**.sum**$[f(x), t, 1] =$ **record**$[x]$**.sum**$[f(x), t, 0] = f(x)$ where $f(x) \in \{-1, 1\}$.

Case 2. $\deg_T(x)$ is odd and $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, -1\}$ or $\deg_T(x)$ is even and $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, 0\}$. Then **record**$[x]$**.sum**$[f(x), t, 1]$ is undefined, and **record**$[x]$**.sum**$[f(x), t, 0]$=$f(x)$ where $f(x) \in \{-1, 1\}$.

Inputting the **parent** array takes $O(n)$ steps, while computing the **degree** array from the **parent** array also takes $O(n)$ steps. Initializing the array **record** takes

$$O\Big(\sum_{x=1}^{n} \deg_T(x)\Big) = O(2m(T)) = O(2(n-1)) = O(n)$$

steps. Thus, the overall complexity here is $O(n)$.

Our next result shows that our algorithm is correct.

Theorem 2. *Suppose* (G, x) *and* (H, y) *are two disjoint rooted subtrees, and let* $(F, x) = (G, x) \circ (H, y)$. *Let* $s \in \{-1, 1\}$, t *be an integer such that* $|t| \le \deg_T(x) - \deg_F(x)$ *with* $t \equiv \deg_T(x) - \deg_F(x) (mod\, 2)$, *and* k *be an integer with* $0 \le k \le |V(F)|$. *Then*
$\mathbf{record}[x]\mathbf{.sum}[s, t, k] = \min\{\mathbf{record}[x]\mathbf{.sum}[s, t+s', j] + \mathbf{record}[y]\mathbf{.sum}[s', s, k-j] \,|\, s' \in \{-1, 1\}, 0 \le j \le k\} = \min\{\mathbf{record}[x]\mathbf{.sum}[s, t+s', j] + \mathbf{record}[y]\mathbf{.sum}[s', s, k-j] \;\; | \, s' \in \{-1, 1\}, \max\{0, k - |V(H)|\} \le j \le \min\{k, |V(G)|\}$.
Moreover, $|t| \le \deg_T(x) - \deg_F(x)$ *if and only if* $-(\deg_T(x) - \deg_G(x) - 1) \le t \le \deg_T(x) - \deg_G(x) - 1$.

Proof. Suppose $f : V(F) \to \{-1, 1\}$ such that

$$f(V(F)) = \mathbf{record}[x].\mathbf{sum}[s, t, k].$$

Let g (respectively, h) be the restriction of f on $V(G)$ (respectively, $V(H)$) and $s^* = h(y) = f(y)$. Note that $f(N_F(x)) + t = g(N_G(x)) + t + s^*$ and $f(N_F(v)) = g(N_G(v))$ for all $v \in V(G) - \{x\}$, while $f(N_F(y)) = h(N_H(y)) + s$ and $f(N_F(v)) = g(N_H(v))$ for all $v \in V(H) - \{y\}$. Thus, $k \le |\{v \,|\, f(N_F(v)) + t \ge 1$ when $v = x$ and $f(N_F(v)) \ge 1$ when $v \ne x\}| = |\{v \,|\, g(N_G(v)) + t + s^* \ge 1$ when $v = x$ and $g(N_G(v)) \ge 1$ when $v \ne x\}| + |\{v \,|\, h(N_H(v)) + s \ge 1$ when $v = y$ and $h(N_H(v)) \ge 1$ when $v \ne y\}|$. If $j = |\{v \,|\, g(N_G(v)) + t + s^* \ge 1$ when $v = x$ and $g(N_G(v)) \ge 1$ when $v \ne x\}|$, then $k - j \le |\{v \,|\, h(N_H(v)) + s \ge 1$ when $v = y$ and $h(N_H(v)) \ge 1$ when $v \ne y\}|$. It now follows that $\mathbf{record}[x]\mathbf{.sum}[s, t+s^*, j] + \mathbf{record}[y]\mathbf{.sum}[s^*, s, k-j] \le g(V(G)) + h(V(H)) = \mathbf{record}[x]\mathbf{.sum}[s, t, k]$. Hence, $\min\{\, \mathbf{record}[x]\mathbf{.sum}[s, t+s', j] + \mathbf{record}[y]\mathbf{.sum}[s', s, k-j] \;\; | \, s' \in \{-1, 1\}, 0 \le j \le k\} \le \mathbf{record}[x]\mathbf{.sum}[s, t, k]$.

On the other hand, suppose $g \colon V(G) \to \{-1, 1\}$ such that

$$g(V(G)) = \mathbf{record}[x].\mathbf{sum}[s, t+s', j]$$

and $h : V(H) \to \{-1, 1\}$ such that

$$h(V(H)) = \textbf{record}[y].\textbf{sum}[s', s, k-j].$$

Define $f : V(F) \to \{-1, 1\}$ by $f(v) = g(v)$ if $v \in V(G)$ and $f(v) = h(v)$ for all $v \in V(H)$. As before, $f(N_F(x)) + t = g(N_G(x)) + t + s'$ and $f(N_F(v)) = g(N_G(v))$ for all $v \in V(G) - \{x\}$, while $f(N_F(y)) = h(N_H(y)) + s$ and $f(N_F(v)) = g(N_H(v))$ for all $v \in V(H) - \{y\}$. Thus, $|\{v \,|\, f(N_F(v)) + t \geq 1$ when $v = x$ and $f(N_F(v)) \geq 1$ when $v \neq x\}| = |\{v \,|\, g(N_G(v)) + t + s' \geq 1$ when $v = x$ and $g(N_G(v)) \geq 1$ when $v \neq x\}| + |\{v \,|\, h(N_H(v)) + s \geq 1$ when $v = y$ and $h(N_H(v)) \geq 1$ when $v \neq y\}| \geq j + (k-j) = k$. Hence, $\textbf{record}[x].\textbf{sum}[s, t, k] \leq f(V(F)) = g(V(G)) + h(V(H)) = \textbf{record}[x].\textbf{sum}[s, t+s', j] + \textbf{record}[y].\textbf{sum}[s', s, k-j]$. Thus, $\textbf{record}[x].\textbf{sum}[s, t, k] \leq \min\{\textbf{record}[x].\textbf{sum}\ [s, t+s', j] + \textbf{record}[y].\textbf{sum}\ [s', s, k-j] \ \,|\, s' \in \{-1, 1\}, 0 \leq j \leq k\}$.

Since $0 \leq j \leq |V(G)|$ and $j \leq k$, we have $0 \leq k - j \leq |V(H)|$, so that $0 \geq j - k \geq -|V(H)|$, whence $j \geq k - |V(H)|$. We conclude that $\max\{0, k - |V(H)|\} \leq j \leq \min\{k, |V(G)|\}$.

Lastly, $|t| \leq \deg_T(x) - \deg_F(x)$ if and only if $-\deg_T(x) + \deg_G(x) + 1 \leq t \leq \deg_T(x) - \deg_G(x) - 1$, since $\deg_F(x) = \deg_G(x) + 1$. ■

At the conclusion of our algorithm, $T = F$, and so $t = 0$. Clearly, $\gamma_{tks}(T) = \min\{$ $\textbf{record}[1].\textbf{sum}[1, 0, k]$,$\textbf{record}[1].$ $\textbf{sum}[-1, 0, k]$ $\}$.

We now present the algorithm, omitting the initialization phase.

Algorithm: To compute $\gamma_{tks}(T)$ for a tree T.

```
for oldRoot ← n downto 2 do
begin
  resultRecord.numvertices ← record[oldRoot].numvertices +
                              record[parent[oldRoot]].numvertices
  resultRecord.degree ← record[parent[oldRoot]].degree + 1
  range ← degree[parent[oldRoot]] - resultRecord.degree
  for newRootValue ← -1 to 1 step 2 do
    for newRootExcess ← -range to range step 2 do
      for k ← 0 to resultRecord.numvertices do
      begin
        minimum ← ∞
        startValue ← max(0, k - record[oldRoot].numvertices)
        stopValue ← min(k, record[parent[oldRoot]].numvertices)
        for j ← startValue to stopValue do
```

```
      begin
        for oldRootValue ← -1 to 1 step 2 do
        begin
          number ← degree[parent[oldRoot]]
                     - record[parent[oldRoot]].degree - 1
          if -number ≤ newRootExcess ≤ number then
          begin
            summand1 ← record[parent[oldRoot]].
                         sum[newRootValue, newRootExcess + oldRootValue, j]
            summand2 ← record[oldRoot].
                         sum[oldRootValue, newRootValue, k-j]
          temp ← summand1 + summand2
        end
        if (temp < minimum)
        then minimum ← temp
      end
    end
    resultRecord.sum[newRootValue, newRootExcess, k] ← minimum
  end
  record[parent[oldRoot]] ← resultRecord
end
for k ← 0 to n do
  output (k, min(record[1].sum[1, 0, k],record[1].sum[-1, 0, k]))
```

The complexity of the above part of the algorithm, excluding the output phase, is

$$
\begin{aligned}
& O\Bigg(\sum_{n-\text{oldroot}=0}^{(n-2)} 2 \times \deg_T\,[\text{parent}[\text{oldroot}]] \times n \times n \times 2 \Bigg) \\
& = O\Bigg(4n^2 \sum_{v \in V(T)} \deg_T(v) \Bigg) \\
& = O\left(4n^2 2m(T) \right) = (4n^2 \times 2 \times (n-1)) \\
& = O(n^3),
\end{aligned}
$$

while the complexity of the output phase is $O(n)$. Thus, the overall complexity of the algorithm is $O(n^3)$.

4. A Cubic Algorithm to Compute $\gamma_{ks}(T)$ of a Tree T

A "quadratic" time algorithm to compute the signed k-subdomination number of a tree appears in [20]. Unfortunately, the initialization phase of the algorithm is omitted and other aspects of the algorithm are not clear either. Also, the complexity analysis of the algorithm seems to be incorrect. In this section, we present a cubic algorithm to compute $\gamma_{ks}(T)$ of a tree T. The approach here is similar to what we described in the previous section. Here we have the following data structure, associated with the subtree (F, x).

1. **record**$[x]$**.numvertices:** the number of vertices in the subtree (F, x).
2. **record**$[x]$**.degree:** $\deg_F(x)$.
3. **record**$[x]$**.sum**$[f(x), t, k]$: the minimum weight of a function $f : V(F) \to \{-1, 1\}$ such that x is assigned $f(x)$, $|t| \leq \deg_T(x) - \deg_F(x)$ (representing all possible sums of assignments of -1 and $+1$ to the vertices of $N_T(x) - N_F(x)$ and $|\{v \mid f(N_F[v]) + t \geq 1$ when $v = x$ and $f(N_F[v]) \geq 1$ when $v \neq x\}| \geq k$, where $1 \leq k \leq$ **record**$[x]$**.numvertices**.

The initialization phase here proceeds as follows.

Let x be an arbitrary vertex of T. Initially, $(F, x) = (K_1, x)$, whence **record**$[x]$**.numvertices**$=1$ and **record**$[x]$**.degree**$=0$. Suppose t is an integer such that $|t| \leq \deg_T(x) - \deg_F(x) = \deg_T(x)$, representing all possible sums of assignments of -1 and $+1$ to the vertices of $N_T(x) - N_F(x) = N_T(x)$. Then $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, \deg_T(x)\}$. The only way for $f(N_F(x)) + f(x) + t = f(x) + t \geq 1$, is for $t \geq 2 - f(x)$ if $\deg_T(x)$ is odd and for $t \geq 1 - f(x)$ if $\deg_T(x)$ is even. Thus, we have the following initializations:

Case 1. $\deg_T(x)$ is odd and $t \in \{2 - f(x), 4 - f(x), \ldots, \deg_T(x)\}$ or $\deg_T(x)$ is even and $t \in \{1 - f(x), 3 - f(x), \ldots, \deg_T(x)\}$. Then **record**$[x]$**.sum**$[f(x), t, 1]$ = **record**$[x]$**.sum**$[f(x), t, 0] = f(x)$ where $f(x) \in \{-1, 1\}$.

Case 2. $\deg_T(x)$ is odd and $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, -f(x)\}$ or $\deg_T(x)$ is even and $t \in \{-\deg_T(x), -\deg_T(x) + 2, \ldots, -1 - f(x)\}$. Then **record**$[x]$**.sum**$[f(x), t, 1]$ is undefined, and **record**$[x]$**.sum**$[f(x), t, 0] = f(x)$ where $f(x) \in \{-1, 1\}$.

A result analogous to Theorem 2 appears in [20].

We are now in a position to state the algorithm, again omitting the pseudocode for the initialization phase. Note that the initialization phase of the algorithm has complexity

$$O\Big(\sum_{v\in V(T)} \deg_T(v)\Big) = O(2m(T)) = O(2(n-1)) = O(n).$$

Thus, the overall complexity of the algorithm is also $O(n^3)$.

Algorithm: To compute $\gamma_{ks}(T)$ for a tree T.

```
for oldRoot ← n downto 2 do
begin
  resultRecord.numvertices ← record[oldRoot].numvertices +
                             record[parent[oldRoot]].numvertices
  resultRecord.degree ← record[parent[oldRoot]].degree + 1
  range ← degree[parent[oldRoot]] - resultRecord.degree
  for newRootValue ← -1 to 1 step 2 do
    for newRootExcess ← -range to range step 2 do
      for k ← 0 to resultRecord.numvertices do
      begin
        minimum ← ∞
        startValue ← max(0, k - record[oldRoot].numvertices)
        stopValue ← min(k, record[parent[oldRoot]].numvertices)
        for j ← startValue to stopValue do
        begin
          for oldRootValue ← -1 to 1 step 2 do
          begin
            number ← degree[parent[oldRoot]]
                     - record[parent[oldRoot]].degree - 1
            if -number ≤ newRootExcess ≤ number then
            begin
              summand1 ← record[parent[oldRoot]].
                         sum[newRootValue, newRootExcess + oldRootValue,j]
              summand2 ← record[oldRoot].
                         sum[oldRootValue, newRootValue, k-j]
              temp ← summand1 + summand2
            end
            if (temp < minimum)
            then minimum ← temp
          end
        end
        resultRecord.sum[newRootValue, newRootExcess, k] ← minimum
      end
  record[parent[oldRoot]] ← resultRecord
end
for k ← 0 to n do
  output (k, min(record[1].sum[-1, 0, k],record[1].sum[1, 0, k]))
```

References

[1] L.W. Beineke and M.A. Henning, *Opinion function on trees*, Discrete Math. **167–168** (1997) 127–139.

[2] I. Broere, J.H. Hattingh, M.A. Henning and A.A. McRae, *Majority domination in graphs*, Discrete Math. **138** (1995) 125–135.

[3] G. Chang, S.C. Liaw and H.G. Yeh, *k-subdomination in graphs*, Discrete Applied Math. **120** (2002) 55–60.

[4] E.J. Cockayne and C. Mynhardt, *On a generalisation of signed dominating functions of graphs*, Ars Combin. **43** (1996) 235–245.

[5] J.E. Dunbar, S.T. Hedetniemi, M.A. Henning and P.J. Slater, *Signed domination in graphs*, Graph Theory, Combinatorics and Applications, John Wiley & Sons, Inc. **1** (1995) 311–322.

[6] O. Favaron, *Signed domination in regular graphs*, Discrete Math. **158** (1996) 287–293.

[7] Z. Furedi and D. Mubayi, *Signed domination in regular graphs and set-systems*, J. Combin. Theory (B) **76** (1999) 223–239.

[8] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, New York, 1979).

[9] L. Harris and J.H. Hattingh, *The algorithmic complexity of certain functional variations of total domination in graphs*, Australasian J. Combin. **29** (2004) 143–156.

[10] L. Harris, J.H. Hattingh and M.A. Henning, *Total k-subdominating functions on graphs*, submitted for publication.

[11] J.H. Hattingh, M.A. Henning and P.J. Slater, *The algorithmic complexity of signed domination in graphs*, Australasian J. Combin. **12** (1995) 101–112.

[12] M.A. Henning, *Domination in regular graphs*, Ars Combin. **43** (1996) 263–271.

[13] M.A. Henning, *Dominating functions in graphs*, Domination in Graphs: Volume II (Marcel-Dekker, Inc., 1998) 31–62.

[14] M.A. Henning, *Signed total domination in graphs*, to appear in Discrete Math.

[15] M.A. Henning and H. Hind, *Strict open majority functions in Graphs*, J. Graph Theory **28** (1998) 49–56.

[16] M.A. Henning and P.J. Slater, *Inequalities relating domination parameters in cubic graphs*, Discrete Math. **158** (1996) 87–98.

[17] P. Lam and B. Wei, *On the total domination number of graphs*, submitted for publication.

[18] J. Matousek, *On the signed domination in graphs*, Combinatoria **20** (2000) 103–108.

[19] X. Yang, X. Huang and Q. Hou, *On graphs with equal signed and majority domination number*, manuscript (personal communication by X. Yang).

[20] H. Yeh and G.J. Chang, *Algorithmic aspects of majority domination*, Taiwanese J. Math. **1** (1997) 343-350.

[21] B. Zelinka, *Some remarks on domination in cubic graphs*, Discrete Math. **158** (1996) 249–255.

[22] B. Zelinka, *Signed total domination number of a graph*, Czechoslovak Math. J. **51** (2001) 225–229.

[23] Z. Zhang, B. Xu, Y. Li and L. Liu, *A note on the lower bounds of signed domination number of a graph*, Discrete Math. **195** (1999) 295–298.

Received 15 September 2003
Revised 24 August 2005