Connected Domination and Spanning Trees with Many Leaves

Yair Caro [*] Douglas B. West [†] Raphael Yuster [‡]

Abstract

Let $G = (V, E)$ be a connected graph. A *connected dominating set* $S \subset V$ is a dominating set that induces a connected subgraph of G. The *connected domination number* of G, denoted $\gamma_c(G)$, is the minimum cardinality of a connected dominating set. Alternatively, $|V| - \gamma_c(G)$ is the maximum number of leaves in a spanning tree of G. Let δ denote the minimum degree of G. We prove that $\gamma_c(G) \le |V| \frac{\ln(\delta+1)}{\delta+1}(1 + o_\delta(1))$. Two algorithms that construct a set this good are presented. One is a sequential polynomial time algorithm, while the other is a randomized parallel algorithm in RNC.

Keywords: Connectivity, Domination, Spanning Trees
AMS Subject Classifications: 05C69, 05C40, 05C05
Running Head: Connected Domination

1 Introduction

All graphs considered here are finite, undirected and simple. For standard graph-theoretic terminology the reader is referred to [5]. A major area of research in graph theory is the theory of domination. Recently two books [14, 15] have been published that present most of the known results concerning domination parameters. Among the most popular of these parameters is the "connected domination number", which we study here.

A subset D of vertices in a graph G is a *dominating set* if every vertex not in D has a neighbor in D. If the subgraph induced by D is connected, then D is called a *connected dominating set*. The *domination number*, denoted $\gamma(G)$, and the *connected domination number*, denoted $\gamma_c(G)$, are the minimum cardinalities of a dominating set and a connected dominating set, respectively. A graph G has a connected dominating set if and only if G is connected; thus $\gamma_c(G)$ is well-defined on the class of connected graphs. Also, trivially, $\gamma_c(G) \ge \gamma(G)$. Results on the connected domination number appear in [3, 4, 7, 8, 10, 11, 16, 17, 23, 25, 26, 28].

[*]Department of Mathematics, University of Haifa-Oranim, Tivon 36006, Israel. email: yairc@macam98.ac.il
[†]Department of Mathematics, University of Illinois, Urbana, Illinois 61801 U.S.A. email: west@math.uiuc.edu
[‡]Department of Mathematics, University of Haifa-Oranim, Tivon 36006, Israel. email: raphy@macam98.ac.il

Spanning trees of connected graphs are a major topic of research in the area of graph algorithms (see, e.g. [9]). The problem of finding a spanning tree that maximizes the number of leaves is equivalent to the problem of computing $\gamma_c(G)$, because a vertex subset is a connected dominating set if and only its complement is (contained in) the set of leaves of a spanning tree.

The problem of connected domination would arise in "real life" in the following scenario. An existing computer network with direct connections described by a graph G must have the property that always any computer turned 'on' must be able to send a message to any other computer turned 'on'. One can make sure a computer is always on by connecting it to an (expensive) UPS (unlimited power supply) source. We can meet the requirement by connecting only the computers in a connected dominating set to such power sources.

In this paper we consider computational and extremal aspects of $\gamma_c(G)$. Finding the maximum number of leaves in a spanning tree of G is NP-hard (cf. [12] p. 206), so we will be content with constructing a spanning tree having many leaves. The NP-hardness of the optimization problem leads us to seek constructive proofs for related extremal problems. A constructive proof that all graphs in a particular class have spanning trees with at least μ leaves becomes an algorithm to produce such a tree for graphs in this class.

When G is a cycle, we can only guarantee 2 leaves. Let $G_{n,k}$ be the class of simple connected n-vertex graphs with minimum degree at least k. Let $l(n,k)$ be the maximum m such that every $G \in G_{n,k}$ has a spanning tree with at least m leaves. Alternatively, define $\gamma_c(n,k) = n - l(n,k)$ to be the minimum m such that every $G \in G_{n,k}$ has a connected dominating set of size at most m. Finally, define $\gamma(n,k)$ to be the minimum m such that each (not necessarily connected) n-vertex graph with minimum degree at least k has a dominating set of size at most m.

The value $l(n,k)$ is known for $k \leq 5$. We have seen that $l(n,2) = 2$. For arbitrary k, C_n generalizes to a well-known graph in $G_{n,k}$ having no tree with "very many" leaves. Let $m = \lfloor \frac{n}{k+1} \rfloor$. Form cliques $R_1, \ldots, R_m$ of orders $k+1$ and $k+2$, altogether having n vertices. Place the cliques in a ring. Delete one edge $x_i y_i$ from each R_i, and restore minimum degree k by adding edges of the form $x_i y_{i+1}$. In forming a spanning tree of this graph, there must be a non-leaf other than $\{x_i, y_i\}$ in each R_i, except possibly for two of them. The construction yields $l(n,k) \leq \lceil \frac{k-2}{k+1} n \rceil + 2$.

This construction is essentially optimal when k is small. Linial and Sturtevant (unpublished) proved that $l(n,3) \geq \frac{n}{4} + 2$. For $k = 4$, the optimal bound $l(n,4) \geq \frac{2}{5}n + \frac{8}{5}$ was proved in Griggs and Wu [13] and in Kleitman and West [19] (two small graphs have no tree with $\frac{2}{5}n + 2$ leaves). In [13] it is also proved that $l(n,5) \geq \frac{3}{6}n + 2$. These proofs are algorithmic, constructing a tree with at least this many leaves in polynomial time. For $k \geq 6$ the exact value of $l(n,k)$ remains unknown.

The example above was thought to be essentially optimal for all k. However, Alon [1] proved by probabilistic methods that when n is large there exists a graph with minimum degree k and with no dominating set of size less than $(1 + o_k(1)) \frac{1+\ln(k+1)}{k+1} n$. Since connected dominating sets are also dominating sets, this shows that $\gamma_c(n,k) \geq (1 + o_k(1)) \frac{1+\ln(k+1)}{k+1} n$. Equivalently, $l(n,k) \leq$

$(1 + o_k(1))\frac{k-\ln(k+1)}{k+1}n$.

A well-known result of Lovász [21] (see [2]) states that $\gamma(G) \le n\frac{1+\ln(k+1)}{k+1}$ for every n-vertex graph G with minimum degree $k > 1$. Together with Alon's result, this yields $\gamma(n,k) = (1 + o_k(1))n\frac{1+\ln(k+1)}{k+1}$. Thus $\gamma(n,k)$ is asymptotically determined (it is known exactly for $k \le 3$ [22, 24]).

Kleitman and West [19] gave an upper bound for $\gamma_c(n,k)$ that is only 2.5 times larger than Alon's lower bound. They proved that if n is sufficiently large, then $\gamma_c(n,k) \le C\frac{\ln k}{k}n$, where C is very close to 2.5. Our main result in this paper improves this result of Kleitman and West and is asymptotically sharp. We summarize our result in the following theorem:

Theorem 1.1

$$\gamma_c(n,k) = (1 + o_k(1))n\frac{\ln(k+1)}{k+1}.$$

An interesting consequence of Theorem 1.1 is that $\gamma_c(n,k)$ behaves essentially like $\gamma(n,k)$ when k is sufficiently large.

We supply two proofs for Theorem 1.1. The first uses a probabilistic approach similar to the proof of the Lovász bound in [2] and to the proof in [6]. Our proof here is more complicated since we need to use several large deviation inequalities, along with a sharpened form of a result of Duchet and Meyniel [11]. The proof yields the following technical statement, which immediately implies Theorem 1.1.

Theorem 1.2 *If G is an n-vertex connected graph with minimum degree k, then*

$$\gamma_c(G) \le n\frac{145 + 0.5\sqrt{\ln(k+1)} + \ln(k+1)}{k+1}.$$

Furthermore, there exists a polynomial time randomized algorithm that generates a connected dominating set this good, with constant probability. This algorithm can also be implemented in parallel in RNC.

Our second proof for Theorem 1.1 yields a purely sequential algorithm for finding a spanning tree with the required number of leaves. At each stage in the algorithm we maintain a subtree of the final spanning tree, and vertices are added to the tree in ways that tend to increase the number of leaves. Unlike the first proof, this algorithm cannot be parallelized, but it has the advantage of being purely deterministic. We summarize this algorithm in the following theorem:

Theorem 1.3 *Given $\epsilon > 0$, and given k sufficiently large in terms of ϵ, every connected simple graph with order n and minimum degree k has a spanning tree with more than $(1-\frac{(1+\epsilon)\ln k}{k})n$ leaves. Furthermore, there is a polynomial time algorithm that constructs such a tree.*

Note that Theorem 1.1 follows also from Theorem 1.3. In the next section we prove Theorem 1.2, and in the final Section 3 we prove Theorem 1.3.

2 The probabilistic proof

We begin with a lemma that sharpens a result of Duchet and Meyniel [11], who proved that $\gamma(G) \le \gamma_c(G) \le 3\gamma(G) - 2$.

Lemma 2.1 *Let G be a connected graph. If X is a dominating set of G that induces a subgraph with t components, then $\gamma_c(G) \le |X| + 2t - 2$. In particular,*

$$\gamma(G) \le \gamma_c(G) \le 3\gamma(G) - 2.$$

Proof: It suffices to show that whenever $t > 1$, we can find at most two vertices in $V \setminus X$ such that adding them to X decreases the number of components by at least one. Partition X into parts X_1 and X_2 such that X_1 and X_2 have no edge connecting them. Let $x_1 \in X_1$ and $x_2 \in X_2$ be two vertices whose distance in G is the smallest possible. The distance between x_1 and x_2 is at most 3, because otherwise there is a vertex (in the middle of a shortest path from x_1 to x_2) that has distance at least 2 from both X_1 and X_2 and is undominated. □

Proof of Theorem 1.2: The theorem clearly holds for $k < 100$, so we assume $k \ge 100$. Let $p = \frac{\ln(k+1)}{k+1}$; note that $0 \le p < 1$. Pick each vertex of V, randomly and independently, with probability p. Let X denote the set of vertices that are picked. Let Y denote the set of vertices that are not picked and have no picked neighbor. Finally, let z denote the number of components in the subgraph induced by X. Clearly, $x = |X|, y = |Y|$ and z are random variables. By definition, $S = X \cup Y$ is a dominating set of G. Since S has at most $z + y$ components, Lemma 2.1 yields $\gamma_c(G) \le x + y + 2(z + y) - 2 = x + 2z + 3y - 2$. It therefore suffices to show that with positive probability,

$$x + 2z + 3y - 2 \le n\frac{145 + 0.5\sqrt{\ln(k+1)} + \ln(k+1)}{k+1}. \tag{1}$$

Claim 1:

$$\text{Prob}\left[x > n\frac{\ln(k+1)}{k+1} + n\frac{0.5\sqrt{\ln(k+1)}}{k+1}\right] < 0.91.$$

Proof of Claim 1: The expectation of x is $E[x] = np = n\frac{\ln(k+1)}{k+1}$. We shall use the following large deviation result attributed to Chernoff (cf. [2], Appendix A):

$$\text{Prob}\Big[x - E[x] > a\Big] < \exp\Big(-a^2/(2pn) + a^3/(2p^2n^2)\Big).$$

(We use here the fact that x is a sum of n independent indicator random variables each having probability p of success). Substituting $n\frac{0.5\sqrt{\ln(k+1)}}{k+1}$ for a in this inequality yields

$$\text{Prob}\left[x > n\frac{\ln(k+1)}{k+1} + n\frac{0.5\sqrt{\ln(k+1)}}{k+1}\right] < \exp\left(-\frac{n^2\ln(k+1)}{8(k+1)^2 n\frac{\ln(k+1)}{k+1}} + \frac{n^3\ln^{3/2}(k+1)}{16(k+1)^3 n^2\frac{\ln^2(k+1)}{(k+1)^2}}\right)$$

$$= \exp\left(-\frac{n}{8(k+1)} + \frac{n}{16(k+1)\sqrt{\ln(k+1)}}\right) \le \exp\left(-\frac{1}{8} + \frac{1}{16\sqrt{\ln(k+1)}}\right)$$

$$\le \exp\left(-\frac{1}{8} + \frac{1}{16\sqrt{\ln(101)}}\right) < 0.91.$$

In the last inequality we used $n \ge k+1 \ge 101$. This establishes Claim 1.

Claim 2:

$$\text{Prob}\left[y > 25\frac{n}{k+1}\right] < 0.04.$$

Proof of Claim 2: For each vertex v, the probability that $v \in Y$ is exactly $(1-p)^{d_v+1}$ where d_v is the degree of v. Since $d_v \ge k$ it follows that the expectation of y satisfies $E[y] \le n(1-p)^{k+1}$. By using the well-known inequality from elementary calculus

$$\left(1 - \frac{\ln(k+1)}{k+1}\right)^{k+1} < \frac{1}{k+1},$$

we obtain $E[y] < n/(k+1)$. It now follows immediately from Markov's inequality that

$$\text{Prob}\left[y > 25\frac{n}{k+1}\right] > \frac{1}{25} \;=\; 0.04.$$

This establishes Claim 2.

It is not easy to bound z directly. Instead, we will show that the total number of vertices in small components in X is rather small. We say that a vertex $v \in V$ is *weakly dominated* if v has fewer than $0.1\ln(k+1)$ neighbors in X. We now bound the probability that a vertex is weakly dominated. Let X_v denote the number of neighbors of v in X. Clearly,

$$E[X_v] = pd_v \ge pk = \frac{\ln(k+1)}{k+1}k \ge \frac{100}{101}\ln(k+1).$$

Claim 3:

$$\text{Prob}\Big[X_v < 0.1\ln(k+1)\Big] < (\frac{1}{k+1})^{0.4}.$$

Proof of Claim 3: Once again, we use a large deviation inequality. However, now we need to bound the lower tail, so we use the inequality (cf. [2] Appendix A)

$$\text{Prob}\Big[X_v - E[X_v] < -a\Big] < \exp\left(-\frac{a^2}{2E[X_v]}\right),$$

which is valid for every $a > 0$. Using $a = (1 - 101/1000)E[X_v]$, we obtain

$$\text{Prob}\Big[X_v < 0.1\ln(k+1)\Big] \le \text{Prob}\left[X_v < \frac{101E[X_v]}{1000}\right] = \text{Prob}\left[X_v - E[X_v] < -(1 - \frac{101}{1000})E[X_v]\right]$$

$$< \exp\left(-\frac{(1-101/1000)^2 E[X_v]^2}{2E[X_v]}\right) = \exp\left(-\frac{(1-101/1000)^2}{2}E[X_v]\right)$$

$$< \exp\left(-\ln(k+1)\frac{100}{101}\frac{(1-101/1000)^2}{2}\right) < \left(\frac{1}{k+1}\right)^{0.4}.$$

This establishes Claim 3.

The event that a vertex v is picked for X is independent of the event that v is weakly dominated. Therefore, by Claim 3, the probability that a vertex is weakly dominated and in X is

$$\text{Prob}\Big[v \in X \text{ and } X_v < 0.1\ln(k+1)\Big] < p\left(\frac{1}{k+1}\right)^{0.4} = \frac{\ln(k+1)}{(k+1)^{1.4}}.$$

Thus, the expected number of weakly dominated vertices in X is at most

$$n\frac{\ln(k+1)}{(k+1)^{1.4}}.$$

Let U be the set of weakly dominated vertices in X. By Markov's inequality,

$$\text{Prob}\left[|U| > 20n\frac{\ln(k+1)}{(k+1)^{1.35}}\right] < \frac{n\frac{\ln(k+1)}{(k+1)^{1.4}}}{20n\frac{\ln(k+1)}{(k+1)^{1.35}}} = \frac{1}{20(k+1)^{0.05}} \le \frac{1}{20\cdot 101^{0.05}} < 0.04.$$

From Claim 1, Claim 2, and the last inequality, it follows that with probability *at least*

$$1 - 0.91 - 0.04 - 0.04 = 0.01 > 0$$

all of the following events happen simultaneously:

$$x \le n\frac{\ln(k+1)}{k+1} + n\frac{0.5\sqrt{\ln(k+1)}}{k+1}, \qquad y \le 25\frac{n}{k+1}, \qquad |U| \le 20n\frac{\ln(k+1)}{(k+1)^{1.35}}. \tag{2}$$

We now fix a choice of X where all these events happen simultaneously. Every component of X that contains no weakly dominated vertex has size at least $0.1\ln(k+1)$. Thus, the number z of components in X satisfies

$$z \le \frac{x}{0.1\ln(k+1)} + 20n\frac{\ln(k+1)}{(k+1)^{1.35}} \le \frac{n\frac{\ln(k+1)}{k+1} + n\frac{0.5\sqrt{\ln(k+1)}}{k+1}}{0.1\ln(k+1)} + 20n\frac{\ln(k+1)}{(k+1)^{1.35}}$$

$$\le \frac{10n}{k+1} + \frac{5n}{k+1} + \frac{20n}{k+1} = \frac{35n}{k+1}. \tag{3}$$

(We have used that $k \ge 100$ implies $(k+1)^{0.35} > \ln(k+1)$). Finally, (2) and (3) yield

$$x + 2z + 3y - 2 < n\frac{145 + 0.5\sqrt{\ln(k+1)} + \ln(k+1)}{k+1},$$

as required in (1). □

We end this section by describing a parallel implementation of Theorem 1.2. The reader unfamiliar with the PRAM model of parallel computing is referred to [18]. Let M be a Boolean array of order n. At the end of the algorithm, the set of indices with $M(i) = True$ will be a connected dominating set in G. In constant parallel time, initialize all of M to $False$. Next, each vertex picks itself for X with probability p. If $v \in X$, then we put $M(v) = True$. Since each vertex is selected independently, this step also runs in constant parallel time.

Each $v \notin X$ now checks to see whether all its neighbors are also not in X (the set of such vertices is denoted Y in the proof). In the CRCW PRAM model this test can also be done in constant time (it is a boolean "and" operation for each $v \notin X$). If $v \in Y$, then we also put $M(v) = True$. Clearly, $X \cup Y$ is a dominating set.

Using well-known NC algorithms for finding components (such as [27]), we can compute the components of the subgraph of G induced by $X \cup Y$ in $O(\log n)$ parallel time on a CRCW PRAM. If there is only one component, then we are finished.

Otherwise, we proceed as follows: We compute distances joining all pairs of vertices of G (namely, the $n \cdot n$ matrix A of the distances). It is well known that computing A can be done in NC using iterated matrix multiplication. We now create a graph H whose vertices are the components of $X \cup Y$, and whose edges connect two components C_1 and C_2 of $X \cup Y$ if and only if for some $c_1 \in C_1$ and some $c_2 \in C_2$, the distance between c_1 and c_2 is at most 3. Given A, one can clearly construct H in constant time on a CRCW PRAM.

By the proof of Lemma 2.1, H is connected. Thus, H has a spanning tree. Spanning trees are also known to be computed in NC (in fact, spanning trees are by-products of the algorithms that find components). Since H has at most $z + y$ vertices (recall that z is the number of components of X), the spanning tree has at most $z + y - 1$ edges. Each such edge corresponds to a path of length 3 between two components of $X \cup Y$. Thus, the total number of internal vertices in these paths is at most $2(z + y - 1)$. If u is such an internal vertex we put $M(u) = True$. This can be done in constant time on a CRCW PRAM.

Thus, we conclude that finding a connected dominating set of G is in NC. The proof of Theorem 1.2 shows that with constant positive probability, this set has size at most $n\frac{145+0.5\sqrt{\ln(k+1)}+\ln(k+1)}{k+1}$. This proves the existence of the desired RNC algorithm. In fact, the overall running time is $O(\log n)$ on a CRCW PRAM using a polynomial number (in fact, $O(n^3)$) parallel processors.

3 The deterministic proof

Before giving the proof of Theorem 1.3, we describe the method as it applies to the iterative construction of a tree with at least $n/4 + 1.5$ leaves when $k = 3$. When $G \neq K_4$, we can start with

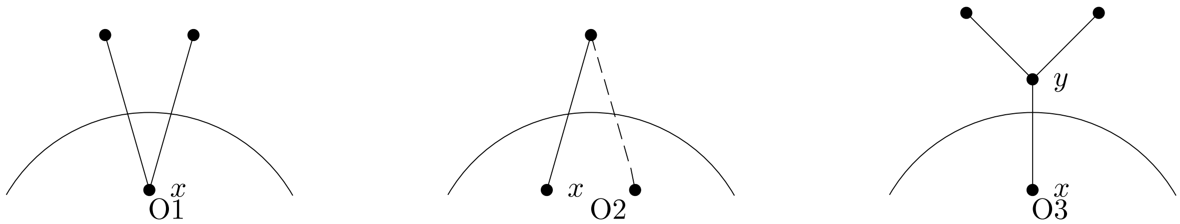

Figure 1: The admissible operations for $k = 3$.

a star at a vertex of degree at least 4 including all its neighbors, or with a double star where both centers have degree 3. (A double star is a tree with exactly two adjacent non-leaves called centers). Let T denote the current tree, with s vertices and l leaves. If x is a leaf of T, then the *external degree* of x, denoted $d'(x)$, is the number of neighbors it has in $G - V(T)$. An *expansion at* x is performed by adding to T the $d'(x)$ edges from x to $N(x) - V(T)$. We grow T by *operations*, which are sequences of expansions. After each operation, all edges from T to $G - V(T)$ are incident to leaves of T.

A leaf x of T with $d'(x) = 0$ is *dead*; no expansion is possible at a dead leaf, and it must be a leaf in the final tree. Let m be the number of dead leaves in T. We call an operation *admissible* if its effect on T satisfies the "augmentation inequality" $3\Delta l + \Delta m \geq \Delta s$, where $\Delta l, \Delta m, \Delta s$ denote the change in the numbers of leaves, dead leaves, and vertices, respectively, in T.

If T is grown to a spanning tree with L leaves by admissible operations, then all leaves eventually die. We begin with 4 leaves and 6 vertices if G is 3-regular (the double-star case above); otherwise with r leaves and $r+1$ vertices for some $r > 3$. In any case, we start with at least four leaves, so the total of Δm from the augmentation inequalities for the operations is at most $L - 4$. Summing the augmentation inequalities over all operations yields $3(L-4)+(L-4) \geq n-6$ if G is 3-regular and $3(L-r)+(L-4) \geq n-r-1$ otherwise. These simplify to $4L \geq n+6$ and $4L \geq n+2r+1 \geq n+7$, respectively, which yield $L \geq n/4 + 1.5$.

The proof for $k = 3$ is now completed by providing a set of admissible operations that can be applied until T absorbs all vertices. The three operations suggested in Figure 1 suffice. If some leaf x of the current tree has external degree at least two, we perform operation O1, which is an expansion at x. Otherwise, if two leaves x and y of T have external degree one, and both have the same unique neighbor outside of T, we perform an expansion at one of them, say x. Note that y becomes dead after the expansion. This is operation O2. The only remaining case is where some leaf has external degree 1, and the unique neighbor outside of T, denoted y, has no other neighbor in T. So, y has at least two other neighbors outside of T. We perform an expansion at x and then an expansion at y. This is operation O3. Note that all operations satisfy the augmentation inequality.

We use the notions of deadness, augmentation inequality, and admissible operation to develop

Figure 2: The operations O_i and P_i.

an algorithm to grow a tree with many leaves when the minimum degree is large. Our operations generalize operations O1 and O3 in Figure 1.

Proof of Theorem 1.3: We describe a polynomial time algorithm that grows the desired tree. Beginning with a star at a vertex of degree k, we again proceed by expanding the current tree T, which has order s, leaf count l, and external degree $d'(x)$ at each leaf x. We seek operations satisfying the augmentation inequality $r\Delta l + \Delta M \geq (r-1)\Delta s$, where the parameter r depends on k and M is a measure of total "deadness" of leaves. The final value of M is a multiple counting of the leaves of the final tree. Each expansion at a leaf adds all outside neighbors, and an operation with one or more expansions is *admissible* if it satisfies the augmentation inequality.

For coefficients $r > \alpha_0 \geq \alpha_1 \geq \cdots \geq \alpha_r = 0$ to be chosen later, we define $M = \sum_{i=0}^{r-1} \alpha_i m_i$, where T has m_i leaves with external degree i. For the final tree, $M = \alpha_0 L$. If we grow a tree by admissible operations, then summing the augmentation inequalities yields $r(L-k) + \alpha_0 L \geq (r-1)(n-k-1)$. Solving for L yields $L \geq \frac{(r-1)n+k+1-r}{r+\alpha_0}$. We choose $r < k$; this permits dropping the additive constant. We then divide top and bottom by r and apply $1/(1+\frac{\alpha_0}{r}) > 1 - \frac{\alpha_0}{r}$ to obtain $L > (1-\frac{1}{r})(1-\frac{\alpha_0}{r})n > (1-\frac{1+\alpha_0}{r})n$.

We define operations O_i and P_i for each i, applied only when the maximum external degree of current leaves is i. The operation O_i is a single expansion at a vertex of external degree i. The operation P_i is expansion at a vertex of external degree i and expansion at one of its new neighbors that introduces the maximum number of additional leaves. The operations are depicted in Figure 2. When the maximum external degree is i, we perform a P_i if the number of vertices introduced by the second expansion is at least $\beta_i = 2r + \alpha_i - i$; if no such P_i exists, we apply an O_i. By construction, some such operation is always available until we grow a spanning tree. It remains to choose r and the constants α_i so that all operations are admissible and so that $(1+\alpha_0)/r < (1+\epsilon)\ln k/k$.

The net change to M by O_i or P_i includes $-\alpha_i$ for the loss of x as a leaf; other changes are gains. We ignore contributions to M from deadness of new leaves, since we have no control over their external degree. For each edge between a new vertex y and a current leaf z other than x, we gain $\alpha_{j-1} - \alpha_j \geq 0$ if this edge reduces $d'(z)$ from j to $j-1$. Note that $j \leq i$. We choose constants

$c_1 \geq \cdots \geq c_r \geq 0$ and define α_i to be $\sum_{j=i+1}^{r} c_j$. This yields $\alpha_{j-1} - \alpha_j = c_j \geq c_i$ if $j \leq i$. Hence $\Delta M \geq -\alpha_i + qc_i$, where q is the number of edges from new vertices to old leaves other than x.

It thus suffices to show that $\Delta s - r(\Delta s - \Delta l) - \alpha_i + c_i q \geq 0$ for each operation performed when the maximum external degree is i. If when applying P_i the second expansion introduces t leaves, then $\Delta s = t + i$ and $\Delta l = t + i - 2$. The desired inequality becomes $t + i - 2r - \alpha_i + c_i q \geq 0$, which holds since we use P_i only when $t \geq 2r + \alpha_i - i$.

When we apply O_i at x, our inability to apply P_i ensures that each of the i new vertices has at most $2r + \alpha_i - i$ neighbors not yet in T and at most i neighbors among x and the other new vertices. Hence it has at least $k - 2r - \alpha_i$ edges to other leaves of T. With i new vertices, this yields $q \geq i(k - 2r - \alpha_i)$. With $\Delta s = i$ and $\Delta l = i - 1$, the desired inequality becomes $c_i i(k - 2r - \alpha_i) \geq r - i + \alpha_i$.

We must choose r and nonincreasing $\{c_i\}$ to satisfy this inequality for all i. We set $c_i = x/i$ for $1 \leq i \leq r$, where x is a positive constant to be chosen in terms of ϵ. The desired inequality becomes $x(k - 2r - \alpha_i) \geq r - i + \alpha_i$. Since $\alpha_i \geq \alpha_{i+1}$ and $i < i + 1$, it suffices to choose r so that the inequality holds when $i = 1$, where it becomes $k - (2 + 1/x)r \geq (1 + 1/x)\alpha_1 - 1/x$. Our choice of c_i yields $\alpha_0 = x \sum_{i=1}^{r} \frac{1}{i} \leq x[\ln r + (1/2r) + 0.577]$. (Knuth [20][pages 73-78] discusses $\sum_{i=1}^{r} 1/i$). Similarly, $\alpha_1 \leq x[\ln r + (1/2r) - 0.423] < x \ln r$. Hence it suffices to choose r so that $k - (2 + 1/x)r \geq (1 + x) \ln r$. We choose $r = \lceil \frac{k}{2+1/x} - \frac{1+x}{2} \ln k \rceil$; this satisfies the last inequality.

We have achieved $\frac{1+\alpha_0}{r} = \frac{(2x+1)\ln k}{k} + o(\frac{\ln k}{k})$. By making $x < \epsilon/2$ and k sufficiently large, we have the desired lower bound on the number of leaves in the final tree. □

Acknowledgment

The authors wish to thank Noga Alon, Teresa W. Haynes and Jerry Griggs for helpful references.

References

[1] N. Alon, *Transversal numbers of uniform hypergraphs*, Graphs and Combinatorics 6 (1990), 1–4.

[2] N. Alon and J. H. Spencer, *The Probabilistic Method*, John Wiley and Sons Inc., New York, 1991.

[3] L. Arseneau, A. Finbow, B. Hartnell, A. Hynick, D. Maclean and L. O'sullivan, *On minimal connected dominating sets*, JCMCC 24 (1997), 185–191.

[4] C. Bo and B. Liu, *Some inequalities about the connected domination number*, Disc. Math. 159 (1996), 241–245.

[5] B. Bollobás, *Extremal Graph Theory*, Academic Press, London, 1978.

[6] Y. Caro, *On k-domination and k-transversal numbers of graphs and hypergraphs*, Ars Combin. 29A (1990), 49-55.

[7] E.J. Cockayne, T.W. Haynes and S.T. Hedetniemi, *Extremal graphs for inequalities involving domination parameters*, Manuscript, 1997.

[8] C.J. Colbourn and L.K. Stewart, *Permutation graphs: Connected domination and Steiner trees*, Discrete Math. 86 (1990), 179-189.

[9] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to algorithms*, The MIT Press, 1990.

[10] A. D'Atri and M. Moscarini, *Distance-hereditary graphs, Steiner trees, and connected domination*, SIAM J. Comput. 17 (1988), 521–538.

[11] P. Duchet and H. Meyniel, *On Hadwiger's number and stability numbers*, Annal. Disc. Math. 13 (1982), 71–74.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[13] J.R.Griggs and M. Wu, *Spanning trees in graphs with minimum degree* 4 *or* 5, Disc. Math. 104 (1992), 167-183.

[14] T. Haynes, S.T. Hedetniemi and P. Slater, *Domination in Graphs: The Theory*, Marcel Dekker Publishers, New York, 1997.

[15] T. Haynes, S.T. Hedetniemi and P. Slater, *Domination in Graphs: Selected Topics*, Marcel Dekker Publishers, New York, 1997.

[16] S.T. Hedetniemi and R. Laskar, *Connected domination in graphs*, In: Graph Theory and Combinatorics, B. Bollobás ed. Academic Press, London 1984, pp. 209–218.

[17] S.T. Hedetniemi and R. Laskar, *Bibliography of domination in graphs and some basic definitions of domination parameters*, Disc. Math. 86 (1990), 257–277.

[18] R.M. Karp and V. Ramachandran, *Parallel algorithms for shared memory machines*, in: "Handbook of Theoretical Computer Science" Vol. A, Chapter 17, J. van Leeuwen ed., Elsevier (1990), 871-941.

[19] D. J. Kleitman and D. B. West, *Spanning trees with many leaves*, SIAM J. Disc. Math. 4 (1991), 99-106.

[20] D.E. Knuth, *The art of Computer Programming*, Addison-Wesley, Reading, Mass. 1973.

[21] L. Lovász, *On the ratio of optimal and integral fractional covers*, Disc. Math. 13 (1975), 383–390.

[22] W. McCuaig and B. Shepherd, *Domination in graphs with minimum degree two*, J. Graph Theory 13 (1989), 749–762.

[23] M. Moscarini, *Doubly chordal graphs, Steiner trees, and connected domination*, Networks 23 (1993), 59–69.

[24] B. Reed, *Paths, stars and the number three*, Combinatorics, Probability and Computing 5 (1996), 267-276.

[25] E. Sampathkumar and H. Walikar, *The connected domination number of a graph*, Math. Phys. Sci. 13 (1979), 607–613.

[26] L.A. Sanchis, *On the number of edges of a graph with a given connected domination number*, Manuscript, 1998.

[27] Y. Shiloach and U. Vishkin, *An* $O(\log n)$ *parallel connectivity algorithm*, J. Algorithms 3 (1982), 57-63.

[28] K. White, M. Farber, and W.R. Pulleyblank, *Steiner trees, connected domination and strongly chordal graphs*, Networks 15 (1985), 109-124.