COMBINATORICA 7 (3) (1987) 221—230

BOUNDS ON THE CONVEX LABEL NUMBER OF TREES

MARSHALL BERN, MARIA KLAWE and ALICE WONG

Received 27 May 1985

A convex labelling of a tree is an assignment of distinct non-negative integer labels to vertices such that whenever x, y and z are the labels of vertices on a path of length 2 then $y \leqq (x+z)/2$. In addition if the tree is rooted, a convex labelling must assign 0 to the root. The convex label number of a tree T is the smallest integer m such that T has a convex labelling with no label greater than m. We prove that every rooted tree (and hence every tree) with n vertices has convex label number less than $4n$. We also exhibit n-vertex trees with convex label number $4n/3+o(n)$, and n-vertex rooted trees with convex label number $2n+o(n)$.

1. Introduction

A labelling of an undirected graph $G=(V, E)$ is a one-to-one mapping L of its vertex set V to the nonnegative integers. A large number of interesting problems in graph theory ask for a labelling which satisfies some special property or is optimal according to some criterion (see [3] for a survey). For example, the min-cut linear arrangement problem is to find a labelling that minimizes the quantity $\max_i \left|\{(u, v)\in E\colon\ L(u)\leqq i<L(v)\}\right|$. A number of other examples are defined by criteria on the weights of the edges, where the weight of an edge (u, v) is the difference of the labels of its endpoints, that is $|L(u)-L(v)|$. For example, a graceful labelling of $G=(V, E)$ is a labelling such that the weights of the edges are the numbers $1, 2, 3, \ldots, |E|$. The optimal linear arrangement problem [8] asks for a labelling that minimizes the sum of the edge weights. Finally, the well known bandwidth problem [7, 14] is to find a labelling that minimizes the maximum edge weight.

Many graph labelling problems assume special significance when they are restricted to trees. Although the min-cut linear arrangement problem is NP-complete for general graphs, recently Yannakakis [16] found a polynomial time algorithm for this problem when it is restricted to trees (see [4], [5], [11] for earlier partial results). Another long standing problem, still unsolved, is to prove that every tree has a graceful labelling; over a hundred papers have appeared on this subject (see [1] for a bibliography). Related problems involving "harmonious", "d-graceful", and "elegant"

The research by M. B. and A. W. was partly supported by NSF grant MCS—8311422.
AMS subject classification (1980): 05 C 05

labellings also give rise to interesting conjectures in the case of trees (see [10], [12], and [2] respectively). The optimal linear arrangement for trees can be solved using dynamic programming [15], whereas the bandwidth problem remains NP-complete [7].

In this paper we consider the convex tree labelling problem and give some bounds on its solution. A one-to-one function L from the vertices of an unrooted tree T to the non-negative integers is called a convex labelling of T if for every path u, v, w of length 2 in T, L satisfies $L(v) \leqq (L(u)+L(w))/2$. If T is rooted with root r, then L must also satisfy $L(r)=0$. For any convex labelling L of T we define its norm $\|L\|$ as max $\{L(v)\colon v$ is a vertex of $T\}$. The convex label number of a tree T is defined as min $\{\|L\|\colon L$ is a convex labelling of $T\}$.

R. Jamison introduced the concept of convex labelling for unrooted trees, and observed that for any tree it is easy to prove an upper bound which is exponential in n, the number of vertices. He also conjectured that every n-vertex tree has convex label number $n-1$. Rall and Slater [13] disproved Jamison's conjecture by showing that if k is a constant, then for infinitely many n there is an n-vertex tree with convex label number greater than $n+k$. Rall and Slater also gave an $n^2/3$ upper bound for the convex label number of any n-vertex tree, and conjectured the existence of a linear upper bound. In this paper we prove their conjecture, namely that every rooted tree with n vertices has convex label number less than $4n$. We also improve Rall and Slater's lower bound by exhibiting n-vertex unrooted trees with convex label number at least $4n/3+o(n)$ and n-vertex rooted trees with convex label number at least $2n+o(n)$.

In the next section we introduce some preliminary notions and go on to prove the lower bounds for rooted and unrooted trees. In section 3 we give a convex labelling algorithm for n-vertex rooted trees and prove that all labels used are less than $4n$. The paper concludes with a discussion of whether the algorithm could be modified to yield a convex labelling of norm less than $2n$.

2. Preliminaries and lower bounds

It is easy to see that if we are interested in minimizing $\|L\|$ for convex labellings L of a tree T, we need only consider those convex labellings which assign 0 to some vertex. Thus for the remainder of this section we will assume that L is a one-to-one function from the vertices of T to the nonnegative integers such that $L(r)=0$ for some vertex r in T. Moreover if T is rooted, r will be the root of T. Even when T is unrooted we will consider r as its root, thus uniquely defining for every vertex v other than r, its parent vertex which we will denote by $F(v)$.

We now give an equivalent definition of convex labelling which we will find more convenient to deal with. If v is a vertex at distance at least two from r, then $F(F(v))$, $F(v)$, v is a path of length 2 in T. Thus if L is a convex labelling we have $L(F(v)) \leqq (L(F(F(v)))+L(v))/2$, or equivalently $L(F(v))-L(F(F(v))) \leqq L(v)-L(F(v))$. Moreover it is not hard to see that if for every vertex w which has distance at least two from r we have $L(F(w))-L(F(F(w))) \leqq L(w)-L(F(w))$, and (as we assumed before) L is a one-to-one function from the vertices of T to the nonnegative integers with $L(r)=0$, then L is a convex labelling of T. Thus if we define the difference along the edge from $F(v)$ to v to be $L(v)-L(F(v))$, then L is a convex labelling if and only if the differences are nondecreasing along every path which begins at r. This manner of viewing convex labellings will be useful both in proving lower bounds and in constructing the algorithm. An immediate consequence of these observations is the following lemma.

Lemma 2.1. *Let L be a convex labelling of T, and suppose v and w are vertices such that v is on the path from r to w. Then*

$$\|L\| \geqq L(v)+t(L(v)-L(F(v))) \geqq (L(v)-L(F(v)))(t+1),$$

where t is the number of edges on the path from v to w. ∎

The preceding lemma together with the following corollary will be our main tools in proving the lower bounds.

Lemma 2.2. *Suppose L is a convex labelling of a tree T and that $x_1, \ldots, x_t$ is a path in T. If q is the integer such that $L(x_q)$ is the minimum value of L on the path $x_1, \ldots, x_t$, then for each u with $q \leqq u \leqq t$ we have $\|L\| \geqq \min \{2t-u-q, (u-q+1)(q-1)\}$.*

Proof. This is clearly true if $q=1$ so assume $q>1$. Since $L(x_q)$ is the minimum value of L on the path $x_1, \ldots, x_t$, it is easy to see that for $i>q$ we have $F(x_i)=x_{i-1}$, and for $i<q$ we have $F(x_i)=x_{i+1}$. Consider the possibilities for $L(x_q), \ldots, L(x_u)$. If $L(x_i)=L(x_q)+i-q$ for $q \leqq i \leqq u$, then $L(x_{q-1}) \geqq L(x_q)+u-q+1$, and so by Lemma 2.1 we have $\|L\| \geqq L(x_q)+u-q+1+(q-2)(u-q+1) \geqq (u-q+1)(q-1)$ since $L(x_q) \geqq 0$. If not, then for some i with $q<i \leqq u$ we must have $L(x_i)-L(x_{i-1}) \geqq 2$ and hence $L(x_u)-L(x_{u-1}) \geqq 2$. Again by Lemma 2.1 this implies

$$\|L\| \geqq L(x_q)+u-q+2(t-u) \geqq 2t-u-q. \quad ∎$$

Theorem 2.3. *For all sufficiently large n there is a rooted tree with n vertices that has convex label number at least $2n+o(n)$.*

Proof. Let T be the path $x_1, x_2, \ldots, x_n$ with root x_q where $q=\lfloor\sqrt{n}\rfloor$, and let $u=3q$. Now if n is large enough we have $q \leqq u \leqq n$. Thus by Lemma 2.2, if L is a convex labelling of T then $\|L\| \geqq \min \{2n-4q, (2q+1)(q-1)\}=2n+o(n)$. ∎

Theorem 2.4. *For all sufficiently large n there is a tree with n vertices with convex label number at least $4n/3+o(n)$.*

Proof. Roughly speaking, the n-vertex tree with convex label number at least $4n/3+o(n)$ is comb shaped with spine of length $n-n^{2/3}$ and teeth of length $n^{1/3}$ uniformly spaced along the spine at intervals of length $n^{2/3}$. More precisely let $k= =\lfloor n^{1/3}\rfloor$, and assume $k>11$. Note that this implies $n>11k^2$. Let S be a path $v_1, \ldots$ $\ldots, v_m$ where $m=n-k(k-1)$, and for $1 \leqq i \leqq k-1$ let T_i be a path of k vertices. Then T is the n-vertex tree formed by adding an edge from one end of T_i to $v_{k^2 i}$ for $1 \leqq i \leqq k-1$. If $s=k^2 i$ for some i, we call the vertex v_s a *junction vertex*. In other words v_s is a junction vertex if some tooth T_i is adjoined to the spine S at v_s.

Let L be a convex labelling of T and let v_j be the vertex of S such that $L(v_j)$ is the minimum value of L on S. Thinking of the vertex r with $L(r)=0$ as the root, it is easy to see that either v_j is r, or v_j is a junction vertex and r is a vertex of the tooth which is adjoined to S at v_j. We will assume that $j \leqq n/2$ as similar arguments apply in the case $j>n/2$. The remainder of the proof depends on whether $j \leqq n/3$.

First suppose $j \leqq n/3$. Since $n > 11k^2$ there is some junction vertex v_s with $j < s \leqq j+k^2$. Let x_1 be the leaf at the end of the tooth which is adjoined to the spine at v_s, and let $x_1, \ldots, x_t$ be the path from x_1 to v_m in T. Thus $x_1, \ldots, x_k$ is the tooth and $x_{k+1}, \ldots, x_t$ is the path $v_s, \ldots, v_m$ along the spine. Since $j < s$, the minimum value of L on the path $x_1, \ldots, x_t$ occurs at $x_{k+1} = v_s$. Letting $u = \lfloor (4/3)k^2 \rfloor$, we have $k+1 \leqq u \leqq t$, and by Lemma 2.2, $\|L\| \geqq \min \{2t - \lfloor (4/3)k^2 \rfloor - k - 1, \ (\lfloor (4/3)k^2 \rfloor - k)k\}$. Obviously

$$(\lfloor (4/3)k^2 \rfloor - k)k = 4n/3 + o(n)$$

since $k = \lfloor n^{1/3} \rfloor$. Now

$$t = m - s + k + 1 > (n - k(k-1)) - (n/3 + k^2) + k + 1,$$

and from this it is easy to see that $2t - \lfloor (4/3)k^2 \rfloor - k - 1 = 4n/3 + o(n)$ also.

Now suppose that $n/3 < j \leqq n/2$. For this case our general strategy is to show that if L does not satisfy $\|L\| \geqq 4n/3 + o(n)$, then L must have the following properties. On an interval immediately to the left of v_j the differences must be less than 4, on an interval slightly to the right of v_j the differences must be exactly 2, and the value on a particular tooth vertex close to v_j must not be too large. Finally we will show that these properties are contradictory. First observe that we may assume that $L(v_{i-1}) - L(v_i) < 4$ for $n/3 - (11/3)k^2 < i \leqq j$, since otherwise Lemma 2.1 would give $\|L\| \geqq 4(i-1) = 4n/3 + o(n)$. We will refer to this as the left-side assumption.

Next let v_s be the junction vertex with $j+3 \leqq s < j+3+k^2$. We will show we may assume that $L(v_i) - L(v_{i-1}) = 2$ for $s \leqq i \leqq s + (2/3)k^2$. If $L(v_i) - L(v_{i-1}) = 1$ for some such i, then $L(v_u) - L(v_{u-1}) = 1$ for $j+1 \leqq u \leqq i$. This implies $L(v_{j-1}) - L(v_j) \geqq i - j + 1 \geqq 4$ which contradicts the left-side assumption. Thus $L(v_i) - L(v_{i-1}) \geqq 2$. On the other hand, $L(v_i) - L(v_{i-1}) \leqq 2$ since otherwise Lemma 2.1 would imply $\|L\| \geqq 3(m-i) = 3n/2 + o(n)$ as $i \leqq n/2 + 3 + (5/3)k^2$ and $m \geqq n - k^2$.

Let x be the tooth vertex adjacent to v_s. If $L(x) \geqq L(v_s) + (4/3)k^2$ we are done since Lemma 2.1 applied to the tooth would yield $\|L\| \geqq k((4/3)k^2) = 4n/3 + o(n)$. Thus suppose $L(x) < L(v_s) + (4/3)k^2$. Let p be maximal such that $L(v_p) > L(x)$ and $p \leqq j$. To see that such a p must exist, first note that we have $L(x) < L(v_s) + (4/3)k^2 \leqq \leqq L(v_j) + 2(s-j) + (4/3)k^2 < L(v_j) + 2(3+k^2) + (4/3)k^2 = L(v_j) + 6 + (10/3)k^2$. Next let $u = j - \lceil (11/3)k^2 \rceil + 1$. Note that $u \geqq 1$ since $j > n/3 > (11/3)k^2$. Now $L(v_u) \geqq L(v_j) + +j - u - 1 \geqq L(v_j) + (11/3)k^2 - 2 > L(x)$, since $k > 11$. This shows that p exists, and moreover, $p \geqq u > n/3 - (11/3)k^2$. In addition, since $L(v_j) < L(x)$, we have $p < j$. Hence, by the maximality of p we have $L(v_{p+1}) < L(x)$.

Finally we observe that since $L(v_{s+i}) = L(v_s) + 2i$ for $0 \leqq i \leqq (2/3)k^2$ there exists some $i > 0$ such that $L(v_{s+i-1}) = L(x) - 1$ and $L(v_{s+i}) = L(x) + 1$. Thus we have $L(v_{p+1}) < L(x)$, $L(v_{s+i-1}) = L(x) - 1$, $L(v_{s+i}) = L(x) + 1$, and $L(v_p) > L(x)$. This implies $L(v_p) - L(v_{p+1}) \geqq 4$, contradicting the left-side assumption since $n/3 - (11/3)k^2 < p + 1 \leqq j$. ∎

3. The upper bound

In this section we prove that every rooted tree with n vertices has convex label number less than $4n$, by presenting an algorithm which constructs a convex labelling with norm $< 4n$. The labelling algorithm can be implemented with $O(n \log n)$ running time.

We will use the term *progression with period d* to denote an infinite arithmetic progression of non-negative integers with period d, whose first element, a, satisfies $0 \leqq a \leqq d-1$. A subprogression of a progression P is any progression which is a (not necessarily proper) subset of P. It is easy to see that if $d|d'$, then every progression with period d is the disjoint union of d'/d subprogressions with period d'.

The algorithm processes the vertices of the tree in three phases. During the first phase each vertex v is assigned a power of 2, $d(v)$, which can be thought of as representing the ideal difference between the label of v and its parent. If it were possible to achieve these ideal differences, the largest label used would be less than $2n$. Duiing the next phase the vertices are processed in increasing order of $d(v)$, and each vertex v is assigned a progression $P(v)$ with period $d(v)$. These progressions will satisfy a somewhat complicated set of conditions. Most of the complexity in proving the correctness of the algorithm goes into proving that such a set of progressions can always be found. During the final stage of the algorithm each vertex is assigned a label which depends only on its progression and the label and ideal difference of its parent. The upper bound of $4n$ is proved by noting that for each vertex v, the actual difference between its label and the label of its parent is bounded by twice its ideal difference.

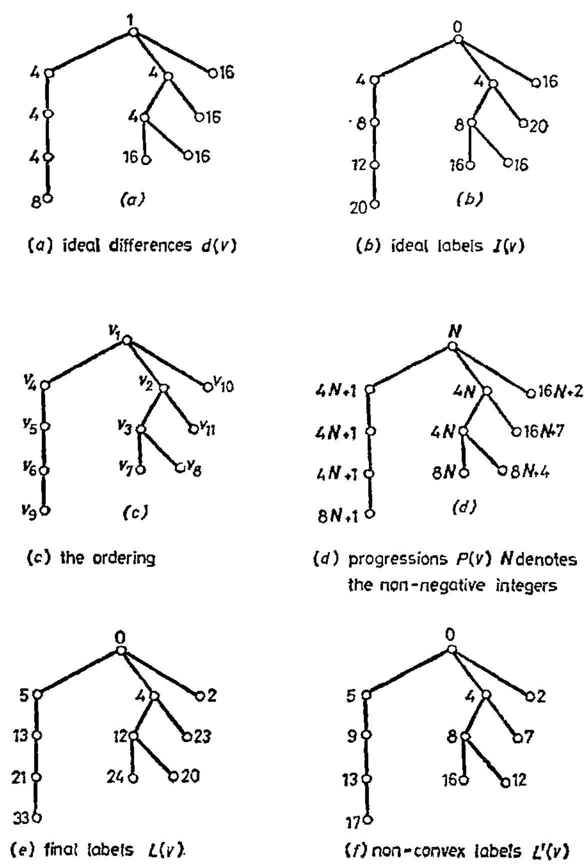

Fig. 1

For each vertex v of a rooted tree T with root r, let $w(v)$ be the number of vertices in the subtree of T which is rooted at v. If $v \neq r$ we use $F(v)$ to denote its parent, and $A(v)$ to denote the set of its proper ancestors. The first phase assigns ideal differences $d(v)$ and ideal labels $I(v)$ recursively as follows. For the root r of T, we define $d(r)=1$ and $I(r)=0$. For $v \neq r$, $d(v)$ is defined to be the power of 2 such that $(2w(r)-I(F(v)))/2 \leqq d(v)w(v) < 2w(r)-I(F(v))$. Intuitively $d(v)$ is the largest power of two that we could hope to use as the "unit difference" in the subtree rooted at v without causing the largest label in that subtree to be greater than or equal to $2w(r)$. $I(v)$ is defined by $I(v)=d(v)+I(F(v))$. Note that the function I is not usually a convex labelling since it is not necessarily one-to-one.

Figure 1 shows the assignments made by the phases of the algorithm for one particular tree. Parts (a) and (b) show the ideal differences and ideal labels.

We first prove some easy lemmas about ideal differences and labels.

Lemma 3.1. *For each vertex v we have $I(v) < 2w(r)$.*

Proof. This is obvious if $v=r$, so suppose $v \neq r$. By definition $d(v)w(v)+I(F(v)) < < 2w(r)$, and since $w(v) \geqq 1$ we have $I(v)=d(v)+I(F(v)) < 2w(r)$. ∎

Lemma 3.2. *If $c>0$ and L is a convex labelling of T such that $L(r)=0$ and for every other vertex v we have $L(v)-L(F(v)) \leqq cd(v)$, then every label is less than $2cw(r)$.*

Proof. From the definition of $I(v)$ it is easy to see that $L(v) \leqq cI(v)$ for all v, which implies $L(v) < 2cw(r)$ by Lemma 3.1. ∎

Lemma 3.3. *If $v \neq r$ then $d(F(v)) \leqq d(v)$.*

Proof. It suffices to show that $(d(F(v))/2)w(v) < (2w(r)-I(F(v)))/2$. We have

$$(d(F(v))/2)w(v) \leqq d(F(v))(w(F(v))-1)/2 < (2w(r)-I(F(F(v)))-d(F(v)))/2 = = (2w(r)-I(F(v)))/2$$

as desired. ∎

Lemma 3.4. *Suppose $d(v)=d(F(v))$. Then for each other vertex u with $F(u)=F(v)$ we must have $d(u)>d(v)$.*

Proof. Suppose $d(u) \leqq d(v)$. Then $d(F(v))w(F(v)) \geqq d(F(v))(w(v)+w(u)+1) \geqq d(v)w(v)+d(u)w(u)+d(F(v)) \geqq (2w(r)-I(F(v)))/2+(2w(r)-I(F(v)))/2+d(F(v)) = =2w(r)-(I(F(v))-d(F(v)))=2w(r)-I(F(F(v)))$ which contradicts the definition of $d(F(v))$. ∎

We now describe the second phase of the labelling algorithm. Let $n=w(r)$ and let $v_1, \ldots, v_n$ be an ordering of the vertices of T such that $d(v_1) \leqq d(v_2) \leqq \ldots \ldots \leqq d(v_n)$, and such that if $v_j=F(v_i)$ then $j<i$. Lemma 3.3 implies that such an ordering always exists. Figure 1(c) shows the ordering we will use for our example. The vertices are assigned progressions in the order $v_1, \ldots, v_n$. The key point in the assignment of progressions is a type of "locality principle". Whenever possible the progression $P(v_i)$ will be a subprogression of $P(F(v_i))$, and in general $P(v_i)$ will be a subprogression of the progression of the closest possible of its ancestors. To describe this precisely we must introduce some more notation. For each $i>1$, we will specify an integer $g(i)<i$ such that $v_{g(i)}$ is an ancestor of v_i. $P(v_i)$ will be a subprogression of $P(v_{g(i)})$ with period $d(v_i)$. For $1 \leqq i \leqq j \leqq n$ we define $R(i,j)=P(v_i) \setminus \bigcup$

$\cup\{P(v_k)\colon k\leqq j$ and $g(k)=i\}$. We think of $R(i,j)$ as the remaining part of $P(v_i)$ immediately after v_j has been assigned its progression. Finally, for $1\leqq i\leqq j\leqq n$ we define $c(i,j)=\sum\{1/d(v_k)\colon g(k)\leqq i,\ i<k\leqq j,\ v_i\in A(v_k)\}$.

Lemma 3.5. *There is an assignment of $g(i)$ and $P(v_i)$ with the following properties:*

(a) *If $R(i,j)\neq\emptyset$, then $R(i,j)$ is the disjoint union of a finite number of progressions, say $P_1,\dots,P_s$, such that their periods $d_1,\dots,d_s$ are powers of 2 with* $\max\{d_1,\dots,d_s\}\leqq d(v_j)$.

(b) *If $1<i<i'$ and $g(i)=g(i')$ then $P(v_i)\cap P(v_{i'})=\emptyset$.*

(c) *If $g(j)<i<j$ and $v_i\in A(v_j)$ then $R(i,j-1)=\emptyset$.*

(d) *If $i<i'$ and $P(v_i)\cap P(v_{i'})\neq\emptyset$ then $v_i\in A(v_{i'})$.*

(e) *If $i\leqq j$ then $c(i,j)\leqq 2(w(v_i)-1)/(2w(r)-I(v_i))$.*

(f) *$R(1,j)\neq\emptyset$ for $1\leqq j\leqq n$.*

Proof. We begin by setting $P(v_1)$, the progression of the root, to be the non-negative integers. It is easy to verify that the properties above hold whenever $i',j\leqq 1$. Suppose $k\geqq 2$ and that $g(i)$, $P(v_i)$ have been assigned for $1\leqq i\leqq k-1$ in such a way that 3.5(a)—3.5(f) are satisfied whenever $i',j\leqq k-1$. We now show how to assign $g(k)$ and $P(v_k)$ so that 3.5(a)—3.5(f) are satisfied whenever i', $j\leqq k$. We define $g(k)$ to be the maximal i such that $v_i\in A(v_k)$ and $R(i,k-1)\neq\emptyset$. In other words, $g(k)$ is the first vertex v_i on the path from v_k to the root with $R(i,k-1)\neq\emptyset$. By 3.5(f) $g(k)$ is well-define since $v_1\in A(v_k)$ and $R(1,k-1)\neq\emptyset$. By 3.5(a) there exists $s\geqq 1$ such that $R(g(k),k-1)$ is the disjoint union $P_1\cup\dots\cup P_s$, where each P_i is a progression with period $d_i\leqq d(v_{k-1})$. Now $d_s\leqq d(v_k)$ since $d(v_{k-1})\leqq d(v_k)$, so P_s is the disjoint union $Q_1\cup\dots\cup Q_t$ of subprogressions with period $d(v_k)$ where $t=d(v_k)/d_s$. We define $P(v_k)=Q_t$.

We now show that 3.5(a) is satisfied when $j=k$. First note that if $i\neq k$ and $i\neq g(k)$, then $R(i,k)=R(i,k-1)$. Thus it suffices to verify 3.5(a) for $R(g(k),k)$ and $R(k,k)$. If $R(g(k),k)\neq\emptyset$, then $R(g(k),k)$ is the disjoint union $P_1\cup\dots\cup P_{s-1}\cup Q_1\cup\dots\cup Q_{t-1}$ and the period of each progression is at most $d(v_k)$. Finally $R(k,k)=P(v_k)$ which obviously satisfies 3.5(a).

It is obvious that 3.5(b) is satisfied for $i'=k$ since $P(v_k)\subset R(g(k),k-1)=P(v_{g(k)})\setminus\cup\{P(v_i)\colon i\leqq k-1$ and $g(i)=g(k)\}$, and 3.5(c) is satisfied for $j=k$ by the definition of $g(k)$.

We now prove that 3.5(d) is satisfied for $i'=k$. Suppose $i<k$ and $P(v_i)\cap P(v_k)\neq\emptyset$. If $i=g(k)$ we are done since $v_{g(k)}\in A(v_k)$. In any case we have $P(v_i)\cap P(v_{g(k)})\neq\emptyset$ since $P(v_k)\subset P(v_{g(k)})$. Thus if $i<g(k)$, since 3.5(d) holds for $i'=g(k)$ because $g(k)\leqq k-1$, we have $v_i\in A(v_{g(k)})$ which implies $v_i\in A(v_k)$ as desired. Hence we may assume $g(k)<i$ which implies that $v_{g(k)}\in A(v_i)$ by 3.5(d) since $i\leqq k-1$. We will prove that in fact this case cannot occur. Consider the sequence $g(i)$, $g(g(i))$, $\dots$, $g^j(i)$. Since $g(s)$ is defined for each $s>1$ and $g(s)<s$, obviously there is some j such that $g^j(i)=1$. Let m be minimal such that $g^m(i)\leqq g(k)$, and let $t=g^{m-1}(i)$, where we use the convention that $g^0(i)=i$. Since $t<k$, $R(g(k),t-1)\neq\emptyset$ and hence by 3.5(c) we must have $g(t)=g^m(i)=g(k)$. This shows that $P(v_t)\cap P(v_k)=\emptyset$. Now it it easy to see that $P(v_i)\subset P(v_t)$ since either $t=i$ or $t=g^{m-1}(i)$ where $m-1\geqq 1$. Thus $P(v_i)\cap P(v_k)=\emptyset$, a contradiction.

Next we show that 3.5(e) is satisfied for $j=k$ and $1\leqq i\leqq k$. Let $n(i)=|\{h\colon h\leqq k$ and $v_i\in A(v_h)\}|$. We will prove 3.5(e) by induction on $n(i)$. If $n(i)=0$, $c(i,k)=0$ and since by Lemma 3.1 we have $I(v_i)<2w(r)$ for all i, the statement clearly holds. Thus suppose $n(i)>0$. For $1\leqq m\leqq k$ we define $c'(m,k)=$

$=\sum\{1/d(v_j)\colon g(j)<m;\ m<j\leqq k,\ v_m\in A(v_j)\}$. By 3.5(c) it is easy to see that $c'(m,k)=\max\{0, c(m,k)-1/d(v_m)\}$. Let $S=\{m\colon F(v_m)=v_i$ and $m\leqq k\}$. Then

$$c(i,k)=\Sigma\{1/d(v_m)+c'(m,k)\colon m\in S\}=\Sigma\{\max\{1/d(v_m), c(m,k)\}\colon m\in S\}.$$

We claim that for each $m\in S$ we have

$$\max\{1/d(v_m), c(m,k)\}\leqq 2w(v_m)/(2w(r)-I(v_i)).$$

First consider $1/d(v_m)$. By definition we have $(2w(r)-I(v_i))/2\leqq d(v_m)w(v_m)$ and rearranging this gives $1/d(v_m)\leqq 2w(v_m)/(2w(r)-I(v_i))$ as desired. Now considering $c(m,k)$, by the inductive hypothesis we have

$$c(m,k)\leqq 2(w(v_m)-1)/(2w(r)-I(v_m))=2Hw(v_m)/(2w(r)-I(v_i))$$

where

$$H=(2w(r)-I(v_i))(w(v_m)-1)/(w(v_m)(2w(r)-I(v_m))).$$

Thus it suffices to show that $H\leqq 1$. Let $X=(2w(r)-I(v_i))w(v_m)$. Now

$$H=(X-(2w(r)-I(v_i)))/(X-d(v_m)w(v_m)),$$

so $H\leqq 1$ if $d(v_m)w(v_m)\leqq 2w(r)-I(v_i)$ which holds by the definition of $d(v_m)$. Thus $\max\{1/d(v_m), c(m,k)\}\leqq 2w(v_m)/(2w(r)-I(v_i))$. Finally, since $\sum\{w(v_m)\colon m\in S\}\leqq w(v_i)-1$, this yields $c(i,k)\leqq 2(w(v_i)-1)/(2w(r)-I(v_i))$.

We conclude the proof by showing that 3.5(f) is satisfied for $j=k$. From the preceding paragraph we have $c(1,k)\leqq 2(w(r)-1)/2w(r)<1$. Let $J=\{i\colon g(i)=1, 1<i\leqq k\}$, and let $c(1,k)=b/d$ where $0\leqq b<d$ and d is a power of 2 such that $d(v_i)\leqq d$ for each i in J. Then by 3.5(b) we have $P(v_i)\cap P(v_i')$ for i, i' distinct elements of J, so it is easy to see that $\cup\{P(v_i)\colon i\in J\}$ is the disjoint union of b progressions $P_1, \ldots, P_b$ of period d. Now by the definition of progression, since $b<d$ at least one of the numbers $0, 1, \ldots, d-1$ is not in any of the P_i and hence $R(1,k)\neq\emptyset$. ∎

Figure 1(d) shows the assignment of progressions for our example.

We are now ready to show how the progressions $P(v)$ can be used to recursively define a convex labelling. Let $L(r)=0$, and suppose we have defined $L(F(v))$. Then $L(v)=\min\{x\in P(v)\colon x\geqq L(F(v))+2d(F(v))\}$. Figure 1(e) gives the labelling thus obtained for our example.

Lemma 3.6. $L(v)-L(F(v))\leqq 2d(v)$.

Proof. There are two cases to consider, $d(v)>d(F(v))$ and $d(v)=d(F(v))$. (We cannot have $d(v)<d(F(v))$ by Lemma 3.3.) First suppose that $d(v)>d(F(v))$. Note that since $d(v)$ and $d(F(v))$ are powers of 2 we have $d(v)\geqq 2d(F(v))$. Since $P(v)$ has period $d(v)$, obviously $L(v)-(L(F(v))+2d(F(v)))<d(v)$, so $L(v)-L(F(v))<$ $<d(v)+2d(F(v))\leqq 2d(v)$. Now suppose $d(v)=d(F(v))$. Let $F(v)=v_i$ and $v=v_j$. By lemmas 3.3 and 3.4 and the choice of order $v_1, \ldots, v_n$ we see that $i<j$, and that for any m with $i<m<j$, v_i is not in $A(v_m)$. Thus $R(i,j-1)=P(v_i)$, and hence $P(v)=P(v_j)=P(v_i)=P(F(v))$. Since $L(F(v))\in P(F(v))$, clearly

$$L(F(v))+2d(F(v))\in P(F(v))=P(v),\quad\text{so}\quad L(v)=L(F(v))+2d(F(v))=$$
$$=L(F(v))+2d(v).\ ∎$$

Lemma 3.7. *If* $u\neq v$ *then* $L(u)\neq L(v)$.

Consequently if $u \in A(v)$ then $L(u) < L(v)$. Thus we may assume that u is not in $A(v)$ and, symmetrically, v is not in $A(u)$. Assume without loss of generality that
Proof. For every nonroot vertex w, since $d(F(w)) > 0$, we have $L(F(w)) < L(w)$.
$u = v_{i'}$ and $v = v_i$ with $i < i'$. Now by 3.5(d) we see that $P(u) \cap P(v) \neq \emptyset$ and since $L(u) \in P(u)$ and $L(v) \in P(v)$ this shows that $L(u) \neq L(v)$. ▮

Theorem 3.8. *L is a convex labelling with norm less than $4n$.*

Proof. It is obvious from the definition that $L(v)$ is a nonnegative integer for each v, and L is one-to-one by Lemma 3.7. Thus to prove that L is a convex labelling it suffices to show that if $F(v) \neq r$ then $L(F(v)) - L(F(F(v))) \leqq L(v) - L(F(v))$. Now by Lemma 3.6 we have $L(F(v)) - L(F(F(v))) \leqq 2d(F(v))$, and by the definition of $L(v)$ we have $2d(F(v)) \leqq L(v) - L(F(v))$. Finally $\|L\| < 4n$ follows immediately from 3.2 and 3.6. ▮

Remark 3.9. *The algorithm can be implemented to run in $O(n \log n)$ time.*

Proof. It is easy to check that the only phase of the algorithm which requires more than linear running time is the actual assignment of the $g(i)$ and the progressions $P(v_i)$ in Lemma 3.5. At first glance, straightforward implementations of this phase seem to require $O(n^2)$ time. However, with a little extra thought it is possible to give an implementation using $O(n \log n)$ time. We sketch such an implementation below.

During this phase we maintain a tree T' so that immediately before the assignment of $g(j)$ and $P(v_j)$, T' contains the vertices v_i with $R(i, j-1) \neq \emptyset$ and the vertices $v_j, \ldots, v_n$. T' is initialized to be the original tree T. After the assignment of $P(v_{j-1})$, if that assignment caused $R(g(j-1), j-1)$ to be $\emptyset$, the children of $v_{g(j-1)}$ in T' become children of the parent of $v_{g(j-1)}$ in T'. The vertex $v_{g(j-1)}$ is then removed from from T'. It is easy to see that with this initialization and updating procedure, T' contains the desired set of vertices before each assignment. Moreover, immediately before we assign $P(v_j)$, the father of v_j in T' is $g(j)$. Thus ignoring the updating of T', each assignment of $g(j)$ and $P(v_j)$ can be done in constant time. Thus it suffices to show that the total time needed for updating T' is $O(n \log n)$. We do this by showing that each vertex has $O(\log n)$ different parents in T' during the entire phase.

First note that there are only $O(\log n)$ possible values for the ideal difference $d(v)$ of a vertex. Thus it is enough to prove that for each vertex w, any two of its parents in T' have different ideal differences. If u and v are different parents in T' of w, with u the earlier parent, then v must be an ancestor of u so $d(v) \leqq d(u)$. Suppose $d(v) = d(u)$. By lemmas 3.3 and 3.4 the vertex v must have exactly one child v' with $d(v') = d(v)$. Now by the properties of the ordering, u cannot strictly precede v' in $v_1, \ldots, v_n$. Moreover, immediately after the assignment of $P(v')$, the vertex v must be removed from T'. Because u does not strictly precede v', the vertex u is still in T' at this point, which makes it impossible for v to be a parent of w in T' at a later time than u. Thus we must have $d(v) < d(u)$. ▮

4. Open problems

We conjecture that for rooted trees our lower bound is tight, i.e.

Conjecture 4.1. *Every rooted tree with n vertices has a convex labelling with norm less than $2n$.*

It is possible that with judicious choices of the progressions $P(v)$, a convex labelling with norm less than $2n$ could be obtained in the following way. Define $L'(r)=0$, and $L'(v)=\min\{x\in P(v)\colon x>L'(F(v))\}$. Clearly $L'(v)-L'(F(v))\leqq d(v)$, and the arguments in the proof of 3.7 show that L' is one-to-one. However in order for L' to be a convex labelling we also need $L'(v)-L'(F(v))\geqq L'(F(v))-L'(F(F(v)))$ whenever $F(v)\neq r$. This would certainly hold if the following conjecture is true.

Conjecture 4.2. *For every tree T, there is an assignment of progressions $P(v)$ to the vertices of T such that the function L' defined by $L'(r)=0$, and $L'(v)=\min\{x\in P(v)\colon x>L'(F(v))\}$ for $v\neq r$, satisfies $L'(v)-L'(F(v))\geqq d(F(v))$.*

Figure 1(f) shows the labelling L' obtained from the progressions given in Figure 1(d). L' is not a convex labelling as can be seen by examining the path v_1, v_2, v_{11}. However, merely changing $P(v_{11})$ to $16N+3$ solves this problem. Unfortunately it is not hard to find examples requiring a much more complicated rearrangement of progressions in order to obtain an L' which is convex.

The question for unrooted trees seems even more difficult. It would be interesting to see an algorithm which takes advantage of the flexibility available in placing the root.

References

[1] G. S. Bloom, A chronology of the Ringel—Kotzig conjecture and the continuing quest to call all trees graceful, *in: Topics in Graph Theory, Annals of the New York Academy of Sciences,* **328** (F. Harary, ed. and, New York Academy of Sciences, 1979, 32—51.

[2] G. J. Chang, D. F. Hsu and D. G. Rogers, Additive variations on a graceful theme: some results on harmonious and other related graphs, *Proc. Twelfth Southeastern Conf. on Combinatorics, Graph Theory, and Computing,* Vol. 1, Congr. Numer. **32** (1981) 181—197.

[3] F. R. K. Chung, Some problems and results in labelings of graphs, *in: The Theory and Applications of Graphs,* (G. Chartrand, ed.), John Wiley, New York, 1981, 255—263.

[4] F. R. K. Chung, On the cutwidth and the topological bandwidth of a tree, *SIAM J. Alg. Disc. Meth.* **6** (1985), 268—277.

[5] M.-J. Chung, F. S. Makedon, I. H. Sudborough, and J. Turner, Polynomial time algorithms for the min cut problem on degree restricted trees, *SIAM J. Comput.,* **14** (1985), 158—177.

[6] M. Gardner, *Wheels, Life, and Other Mathematical Amusements,* W. H. Freeman and Co., 1983, 152—165.

[7] M. R. Garey, R. L. Graham, D. S. Johnson and D. E. Knuth, Complexity results for bandwidth minimization, *SIAM J. Appl. Math.,* **34** (1978), 477—495.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman and Co., 1979, p. 200.

[9] S. Golomb, How to number a graph, *in: Graph Theory and Computing,* (R. C. Read, ed.), Academic Press, N. Y. (1972), 23—37.

[10] R. L. Graham and N. J. A. Sloane, On additive bases and harmonious graphs,

[11] T. Lengauer, Upper and lower bounds on the complexity of the min cut linear arrangement problem on trees, *SIAM J. Alg. Disc. Meth.,* **3** (1982) 99—113.

[12] M. Maheo and M. Thuillier, On *d*-graceful graphs, *Ars. Combin.* **13** (1982), 181—192.

[13] D. F. Rall and P. J. Slater, Convex labellings of trees, *preprint.*

[14] J. Saxe, Dynamic programming algorithms for recognizing small bandwidth graphs in polynomial time, *SIAM J. Alg. Disc. Meth.,* **1** (1980) 363—369.

[15] Y. Shiloach, A minimum linear arrangement algorithm for undirected trees, *SIAM J. Comput.,* **8** (1979), 1—32.

[16] M. Yannakakis, A polynomial algorithm for the min-cut linear arrangement of trees, *JACM* **32** (1985), 950—988.

Marshall Bern, Alice Wong

Computer Science Division
U. C. Berkeley
Berkeley, CA 94720, USA

Maria Klawe

IBM Almaden Research Center
K53/802, 650 Haury Rd.
San Jose, CA 95120, USA