A FIBONACCI THEME ON BALANCED BINARY TREES

Yasuichi Horibe
Shizuoka University, Hamamatsu, 432, Japan
(Submitted October 1990)

In this paper we show that, when a binary tree is in a certain critical balance, there emerge the Golden Ratio and the Fibonacci numbers.

The paper consists of two sections. In the first section we find some elementary balance properties of optimal binary trees with variously weighted leaves. In the second section, a basic inequality implied by the optimality of trees is in turn used to define what we mean by "balanced" for a binary tree with leaves all weighted 1. The Fibonacci tree is then shown to be a highest balanced tree.

1. Balance Properties of Optimal Binary Trees

Consider a binary tree that has n leaves (terminal nodes) with weights or probabilities $p_i > 0$, $p_1 + \cdots + p_n = 1$, assigned to leaves. It has, then, $n - 1$ internal nodes, where an internal node is a node that has two children. We define the weight of an internal node as the sum of all leaf weights of the subtree rooted at this node. Therefore, recursively, the weight of an internal node is the sum of the weights of its children. Clearly the root has weight 1.

A node is said to be at level k if the length of the path from the root to this node is k. The root is, hence, at level 0. Let ℓ_i be the level of the leaf weighted p_i. Then the average path length is defined by

$$L = \sum p_i \ell_i .$$

In this section we shall be concerned with a binary tree that is *optimal* in the sense that it has the minimum average path length for the given leaf weights.

The well-known Huffman algorithm [3] finds an optimal tree called the *Huffman tree*. The algorithm can be stated in the following recursion form: First, find the two least weights, say x and y, in the list $p_1, p_2, \ldots, p_n$, and replace these two by z $(= x + y)$. Then construct a Huffman tree for the new list of $n - 1$ weights, and then split, in this tree, a leaf of weight z into its children of weights x and y. Note, however, that not every optimal tree is a Huffman tree.

The original motivation for minimizing average path length was to minimize expected search time to leaves. Suppose that one person thinks of $z \in \{1, \ldots, n\}$ and you attempt, knowing $\text{Prob}\{z = i\} = p_i$, to determine what it is by asking questions that can be answered "yes" or "no." Then you may use a binary tree with leaves $1, \ldots, n$ of weights $p_1, \ldots, p_n$ as follows. You ask the first question at the root: "Does z belong to the left subtree of the root?" If the answer to this question is yes [no], then you go to the left [right] child of the root, say a, where you ask the second question: "Does z belong to the left subtree of a?" If the answer to this question is yes [no], then you go to the left [right] child of a, ... The average number of questions required to find z is given by the average path length of the tree.

Lemma 1: If w_{k-1} and w_k are weights of nodes at levels $k - 1$, k in an optimal tree, then $w_{k-1} \geq w_k$.

Proof: If the node of weight w_k exists in the subtree rooted at the node of weight w_{k-1}, then the assertion is obviously true. If not, consider exchanging the subtrees rooted at these nodes. Denote by L and L' the average path lengths of the trees before and after the exchange, respectively. Then we have $L' - L = w_{k-1} - w_k$, because leaves with total weight w_{k-1} have path length one longer under L', and leaves with total weight w_k have path length one longer under L. Since the tree before the exchange is optimal, we have $L \le L'$, hence $w_{k-1} \ge w_k$. □

We say that w_{k-1}, w_k, w_{k+1} is a *weight sequence* in a binary tree if w_k is the weight of an internal node at level k, w_{k-1} is the weight of its parent, and w_{k+1} is the weight of one of its children. Also, let $\overline{w}_k$ and $\overline{w}_{k+1}$ be the weights of the "brothers" of those nodes with weights w_k, w_{k+1}, respectively.

Theorem 1: If w_{k-1}, w_k, w_{k+1} is a weight sequence in an optimal tree, then

$$w_{k-1} \ge w_k + w_{k+1}.$$

Proof: By Lemma 1, we have $\overline{w}_k \ge w_{k+1}$. Hence,

$$w_{k-1} = w_k + \overline{w}_k \ge w_k + w_{k+1}. \quad \square$$

This inequality was implicit in [4] for Huffman trees and was explicitly stated in [1]. It was shown in [1] that it also holds in a weight-balanced tree if the node with weight w_{k+1} is internal or if the sequence of leaf weights forms a valley, i.e.,

$$p_1 \ge \cdots \ge p_j \le \cdots \le p_n \text{ for some } j,\ 1 \le j \le n.$$

Theorem 2: If w_{k-1}, w_k, w_{k+1} is a weight sequence in an optimal tree, then

$$w_k/w_{k-1} \le 2/3.$$

Proof: From Theorem 1, we have

$$(1) \quad \begin{aligned} w_{k-1} &\ge w_k + w_{k+1} \\ w_{k-1} &\ge w_k + \overline{w}_{k+1}. \end{aligned}$$

Putting $p = w_k/w_{k-1}$ and $q = w_{k+1}/w_k$, these inequalities, divided by w_{k-1}, can be written as

$$(2) \quad \begin{aligned} 1 &\ge p + pq \\ 1 &\ge p + p(1-q), \end{aligned}$$

which, added together, gives $p \le 2/3$. □

From this theorem, if the node with weight $\overline{w}_k$ is also internal, we have

$$1/3 \le w_k/w_{k-1} \le 2/3.$$

Otherwise, w_k/w_{k-1} can be arbitrarily small. These bounds 1/3 and 2/3 can be attained as seen from the Huffman tree for the leaf weights:

$$3^{-m-1},\ 3^{-m-1},\ 3^{-m-1},\ 3^{-m},\ 3^{-m},\ \ldots,\ 3^{-2},\ 3^{-2},\ 3^{-1},\ 3^{-1}.$$

The set of points (p, q), $0 < p < 1$, $0 < q < 1$, satisfying (2) above forms the region *ABCDO* in Figure 1. The figure may aid one to graphically understand the balance properties stated in the following. Here ψ is the Golden Section point of the unit interval:

$$\psi = (\sqrt{5} - 1)/2,\ 1 - \psi = \psi^2.$$

Now let us say that a is α-*balanced* for $1/2 \le \alpha \le 1$ if $a \in [1 - \alpha, \alpha]$. The next theorem states that a lack of ψ-balance involving siblings at one level is immediately restored at the next lower level.

Theorem 3: If w_{k-1}, w_k, w_{k+1} is a weight sequence in an optimal tree and if $w_k/w_{k-1} > \psi$, then

(a) w_{k+1}/w_k is ψ-balanced,

(b) $(w_k/w_{k-1}) + (w_{k+1}/w_k) < 2\psi$.

Proof: From the first inequality of (2), we have

$$q \le 1/p - 1 < 1/\psi - 1 = \psi,$$

and from the second,

$$q \ge 2 - 1/p > 2 - 1/\psi = 1 - \psi.$$

Therefore, q is ψ-balanced. For (b) use $p + q = p + 1/p - 1$ obtained from (2). The function $p + 1/p - 1$ is monotonically decreasing for $p < 1$; hence, $p + q$ is less than $\psi + 1/\psi - 1 = 2\psi$ by the assumption $p > \psi$. □

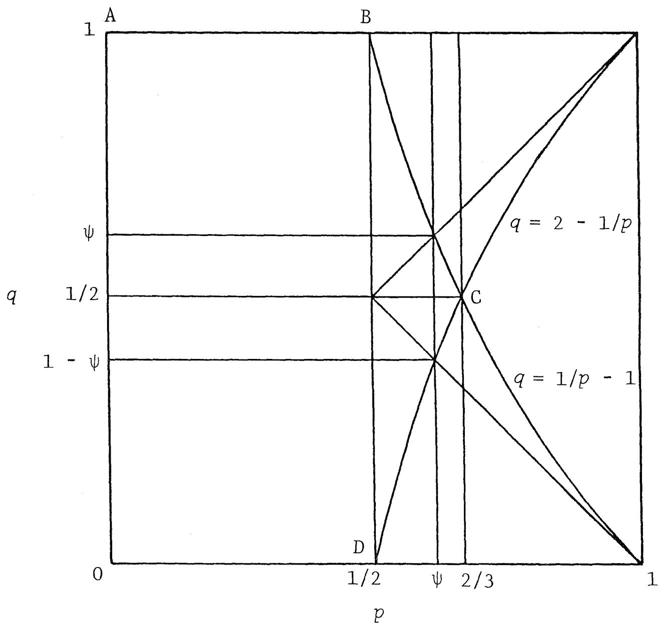

Figure 1

Notice that (1) is equivalent to the following "uncle ≥ nephew" condition:

$$\begin{aligned} \bar{w}_k &\ge w_{k+1}, \\ \bar{w}_k &\ge \bar{w}_{k+1}. \end{aligned} \tag{3}$$

Hence, the worse the balance of p (approaching 2/3), the better the balance of q (approaching 1/2). And the critical point for turning back to a better balance may be defined by the number

$$\alpha^* = \inf\{\alpha : p > \alpha \Rightarrow 1 - \alpha \le q \le \alpha\}.$$

Theorem 4: $\alpha^* = \psi$.

Proof: To determine the critical point, we set $q = p = \alpha^*$ and assume the equality $q = 1/p - 1$, i.e., $\alpha^* = 1/\alpha^* - 1$. □

What about the upper bound on w_k? Is there a bound in terms of k? Letting $F_{n+1} = F_n + F_{n-1}$ $(n \ge 1)$, $F_0 = 0$, $F_1 = 1$, be the Fibonacci numbers, we have

Theorem 5: If $w_0, \ldots, w_k, w_{k+1}$ are the weights on a path from the root in an optimal tree, then $w_k \le 2/F_{k+3}$.

Proof: By Theorem 2, we have

$$w_{k-1} \geq (3/2)w_k = (F_4/F_3)w_k.$$

Using the basic subadditivity relation of Theorem 1, we have, recursively,

$$w_{k-2} \geq w_{k-1} + w_k \geq (F_4/F_3)w_k + w_k = (F_5/F_3)w_k,$$
$$w_{k-3} \geq w_{k-2} + w_{k-1} \geq (F_5/F_3)w_k + (F_4/F_3)w_k = (F_6/F_3)w_k,$$
$$\dots$$
$$1 = w_0 \geq w_1 + w_2 \geq (F_{k+2}/F_3)w_k + (F_{k+1}/F_3)w_k = (F_{k+3}/F_3)w_k,$$

completing the proof. □

The bound $2/F_{k+3}$ can also be attained. This is seen from one Huffman tree (there may be many) for the leaf weights that are the following divided by F_{k+3} (see Figure 2):

$$F_1,\ F_2,\ F_2,\ F_3,\ F_4,\ \dots,\ F_k,\ F_{k+1}.$$

The internal node at level i has weight

$$w_i = F_{k+3-i}/F_{k+3},\quad 0 \leq i \leq k.$$

We have $w_k/w_{k-1} = F_3/F_4 = 2/3$, and all the inequalities in the proof of Theorem 5 become equalities, and $w_k = 2/F_{k+3}$. Furthermore, we see that

$$w_i/w_{i-1} = F_{k+3-i}/F_{k+4-i}$$

approaches ψ for each i when k becomes large. This Huffman tree is a tree where the restoration of the ψ-balance is occurring "most" frequently, because, from the well-known identity $F_{n-1} - \psi F_n = (-\psi)^n$, the ratio F_{n-1}/F_n becomes larger or smaller than ψ, alternately.

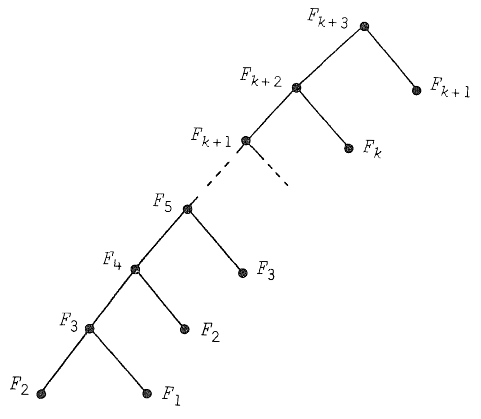

Figure 2

2. Fibonacci Tree as a Highest Balanced Tree

In the binary tree we consider here in this section, the weight of a node is defined as the number of leaves of the subtree rooted at the node. Hence, the leaf weights are all one, the weight of the root is just the total number of leaves.

When can we say that a binary tree is generally "balanced"? One natural definition may come from the inequality of Theorem 1. Since this relation is equivalent to (3) given in the previous section, let us say that a binary tree is *balanced* if it satisfies the following condition.

Balance Condition: The weight of every node is greater than or equal to the weight of each of its two "nephews" (if they exist).

A binary tree in this weight model is thus balanced if and only if the two subtrees at the children of the root are balanced and the weights of the children of the root are larger than or equal to the weights of their nephews. Also, this condition need only be checked for the child of the root with smaller weight.

There are other balance conditions that can be enforced in constructing trees, some applicable from the top down. The λ-*weight-balancing* described in [2] is such a method. Given $1/2 \le \lambda < 1$, to construct a binary tree with n leaves by λ-weight-balancing, we find the integer m such that

$$m - (1 - \lambda) < \lambda n \le m + \lambda,$$

let m and $n - m$ be the weights of the children of the root (i.e., the number of leaves assigned to each subtree), and proceed similarly to construct the two subtrees. The partition $m : (n - m)$ of n is a discrete version of the cut $\lambda : (1 - \lambda)$ of the unit interval. Notice that the λ-weight-balancing can be considered as a method to build a binary tree having a self-similar structure. We will show that the tree with n leaves built by this method is a balanced tree for every n if and only if $1/2 \le \lambda \le \psi$. First, we review a few things about the Fibonacci trees (see [2]).

The Fibonacci tree of order k, denoted by T_k, is a binary tree that has F_k leaves, and is constructed as follows: T_1 and T_2 are simply the roots only, and for $k \ge 3$ the left subtree of T_k is T_{k-1} and the right subtree is T_{k-2}. Let us denote by $T(n)$ the tree with n leaves constructed by ψ-weight-balancing. We may call $T(n)$ "the extended Fibonacci tree," for it has been shown in Theorem 5 of [2] that

$$T(F_k) = T_k.$$

We also have

Theorem 5 [2]: If $n = F_k + r$, where $0 \le r < F_{k-1}$, then the height of $T(n)$ is $k - 2$. [From $F_k \sim (1/\sqrt{5})\psi^{-k}$, we have $k - 2 \sim (\log n)/(-\log \psi)$.]

Theorem 6: If the tree with n leaves constructed by λ-weight-balancing is a balanced tree for every n, then we have $\lambda \le \psi$.

Proof: Suppose $\lambda = \psi + \varepsilon$, $\varepsilon > 0$. Let m_1 and $n - m_1$ be the weights of the children of the root, and let m_2 and $m_1 - m_2$ be the weights of the children of the node with weight m_1. The balancing rule implies $\lambda n \le m_1 + \lambda$ and $\lambda m_1 \le m_2 + \lambda$. Although the node with weight m_2 is a nephew of the node with weight $n - m_1$, we have m_2 greater than $n - m_1$, if n is taken large, as shown below:

$$\begin{aligned} m_2 - (n - m_1) &\ge \lambda m_1 - \lambda - n + m_1 \\ &\ge \lambda(\lambda n - \lambda) - \lambda - n + (\lambda n - \lambda) \\ &= \varepsilon(\sqrt{5} + \varepsilon)n - (\lambda^2 + 2\lambda), \end{aligned}$$

where we used $\lambda = \psi + \varepsilon$ and $\psi = (\sqrt{5} - 1)/2$. □

Approximately speaking, the above proof is like this: The bipartition of n by the ratio $\lambda : (1 - \lambda)$ makes children with weights λn and $(1 - \lambda)n$. And the partition of λn by the same ratio produces the node with weight $\lambda(\lambda n)$, which is a nephew of the node of weight $(1 - \lambda)n$. The balance condition requires $\lambda(\lambda n) \le (1 - \lambda)n$; hence, $\lambda^2 \le 1 - \lambda$, and we have $\lambda \le \psi$.

Next, we show

Theorem 7: The tree with n leaves constructed by ψ-weight-balancing is a balanced tree.

Proof: We prove by induction on n that $T(n)$ is balanced. Trivially, $T(2)$ is a balanced tree. Let us represent n (≥ 3) in the following form:

$$n = F_k + r,\ 0 \leq r < F_{k-1}.$$

Then $k \geq 4$. Let m_1 and $n - m_1$ be the weights of the children of the root, so that $m_1 - (1 - \psi) < \psi n \leq m_1 + \psi$. As noted in [2],

$$m_1 = F_{k-1} + s, \text{ where } s = \lceil \psi r - \psi - (-\psi)^k \rceil \text{ and } 0 \leq s < F_{k-2}.$$
$$(\lceil x \rceil = \text{the least integer} \geq x)$$

Furthermore, let m_2 and $m_1 - m_2$ be the weights of the children of the node with weight m_1, then, similarly, we have

$$m_2 = F_{k-2} + \lceil \psi s - \psi - (-\psi)^{k-1} \rceil.$$

The left and right subtrees of $T(n)$ are $T(m_1)$ and $T(n - m_1)$, which are balanced by the induction hypothesis. Since, clearly, $m_1 \geq n - m_1$ and $m_2 \geq m_1 - m_2$, we only need to show $n - m_1 \geq m_2$ or $n \geq m_1 + m_2$.

$$\begin{aligned} m_1 + m_2 &= (F_{k-1} + s) + (F_{k-2} + \lceil \psi s - \psi - (-\psi)^{k-1} \rceil) \\ &= F_k + \lceil s + \psi s - \psi - (-\psi)^{k-1} \rceil \\ &\leq F_k + \lceil (1 + \psi)(\psi r - \psi - (-\psi)^k + 1) - \psi - (-\psi)^{k-1} \rceil \\ &= F_k + \lceil r + (\psi^2 + \psi - 1)(r - 1 + (-\psi)^{k-1}) \rceil \\ &= F_k + r \\ &= n, \end{aligned}$$

completing the proof. □

Remark: If we use the rule

"$m - 1/2 < \psi n \leq m + 1/2$"

instead of the rule

"$m - (1 - \psi) < \psi n \leq m + \psi$,"

it will construct an unbalanced tree, when $n = 9$, for example. If we use the rule

"$m < \psi n \leq m + 1$,"

the tree built will not become T_8 when $n = F_8$, for example.

Now what can we say about the height of a balanced tree?

Lemma 2: Denote by n_h the minimum number of leaves that a balanced tree of height h can have. Then we have $n_h = F_{h+2}$.

Proof: Induction on h. It is immediate that $n_0 = 1 = F_2$ and $n_1 = 2 = F_3$. Consider a balanced tree of height $h \geq 2$ with n_h leaves. Let a, a' be the weights of the children of the root. We may assume that the subtree rooted at the node with weight a has height $h - 1$. We may further assume that, letting b be the weight of a child of the node with weight a, the subtree rooted at the node with weight b has height $h - 2$. Since any subtree of a balanced tree is itself a balanced tree, we have

$$a \geq n_{h-1},\quad b \geq n_{h-2}.$$

By the induction hypothesis, we have

$$n_{h-1} = F_{h+1},\quad n_{h-2} = F_h.$$

Hence,

$$a \geq F_{h+1},\quad b \geq F_h.$$

The balance condition implies

$$a' \geq b.$$

Consequently, we have

$$n_h = a + a' \geq F_{h+1} + F_h = F_{h+2}.$$

On the other hand, consider the Fibonacci tree T_{h+2}. This is a balanced tree by Theorem 7, and has height h by Theorem 5, and has F_{h+2} leaves. Hence, from the minimality of n_h, we have

$$n_h \leq F_{h+2}.$$

In conclusion, we have $n_h = F_{h+2}$. This completes the proof. □

Theorem 8: In the class of all the balanced trees with n leaves, the extended Fibonacci tree $T(n)$ is a highest one.

Proof: Let $n = F_k + r$, $0 \leq r < F_{k-1}$, and let h be the height of an arbitrary balanced tree with n leaves. From Lemma 2, we have $n \geq F_{h+2}$; hence,

$$F_{h+2} \leq F_k + r < F_k + F_{k-1} = F_{k+1}.$$

This implies $h + 2 \leq k$ or $h \leq k - 2$. However, $k - 2$ is the height of $T(n)$ by Theorem 5. □

Acknowledgment

The author is indebted to the referee for many helpful comments.

References

1. Y. Horibe. "Weight Sequences and Individual Path Length in a Balanced Binary Tree." *J. Combin. Inform. System Sci.* *4* (1979):19-22.
2. Y. Horibe. "An Entropy View of Fibonacci Trees." *Fibonacci Quarterly 20* (1982):168-78.
3. D. A. Huffman. "A Method for Construction of Minimum-Redundancy Codes." *Proc. I.R.E.* *40* (1952):1098-1101.
4. G. O. H. Katona & T. O. H. Nemetz. "Huffman Codes and Self-Information." *IEEE Trans. on Information Theory* *IT-22* (1976):337-40.

AMS Classification numbers: 05C, 94A.

*****