Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT
ELSEVIER
Journal of Computational and Applied Mathematics 162 (2004) 51–55
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Trees and jumps and real roots

Werner Krandick

Department of Computer Science, Drexel University, 3200 Chestnut Street, Philadelphia, PA 19104, USA

Received 20 March 2002; received in revised form 11 December 2002

Abstract

When the nodes of a tree are visited in depth-first order there are occasional jumps from a deeper level of the tree to a higher level. On the set of all full binary trees with a given number of nodes there is about 1 jump for every 2 internal nodes, and the average jump distance is about 2 levels. These averages are close to averages for trees that arise in polynomial real root isolation.

MSC: 68R10; 68W40

Keywords: Algorithm analysis; Binary trees; Depth-first search; Catalan numbers

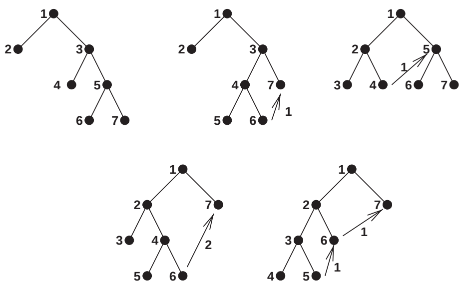
1. Motivation

When the nodes of a tree are visited in depth-first order there are occasional jumps from a deeper level of the tree to a higher level. No such jumps occur when the nodes are visited in breadth-first order. In certain applications, jumps are associated with an extra cost that depends on the jump distance. This is the case in an algorithm proposed by Rouillier and Zimmermann [2].

These authors propose a space-saving variation of the well-known Descartes method for polynomial real root isolation. Their method traverses a full binary tree in depth-first order. The nodes of the tree are associated with certain integral polynomials. Since the method uses exact computation, the polynomials at deeper levels of the tree tend to have longer coefficients than the polynomials at higher levels. Constructing a high polynomial from a deep polynomial thus will involve more expensive computations than constructing the same polynomial from its parent. The extra cost will be particularly high when the difference of the levels is large.

E-mail address: krandick@cs.drexel.edu (W. Krandick).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2003.08.018



n	b_n	j_n	d_n
0	1	0	0
1	1	0	0
2	2	1	1
3	5	5	6
4	14	21	28
5	42	84	120
6	132	330	495
7	429	1287	2002
8	1430	5005	8008
9	4862	19448	31824

Fig. 1. (Left:) The full binary trees with 3 internal nodes. The nodes of each tree are labeled in depth-first order. Arrows indicate jumps and are labeled with the jump distance. (Right:) If there are n internal nodes then there are b_n full binary trees with a total of j_n jumps and a total jump-distance d_n.

However, we show that, on the average, there is only about 1 jump for every 2 internal nodes, and the average jump distance is only about 2 levels when the average is taken over the set of all full binary trees with a given number of nodes. By experiment we show that these averages are close to averages observed in actual real root isolation.

2. Jumps in depth-first traversal of full binary trees

Definition 1. A binary tree is *full* if every internal node has exactly two children [1]. The *depth-first ordering* of a full binary tree is a total ordering of the nodes such that the first element is the root of the tree and the successor of any internal node is its left child and the successor of any leaf is the right sibling of the closest ancestor that is itself a left child. Any transition from a node at a deeper level to a node on a strictly higher level is called a *jump*; the *jump distance* is the (positive) difference of the levels.

The diagrams in Fig. 1 illustrate the definition. We compute the average number of jumps and the average jump distance on the set of full binary trees with n internal nodes, $n \geqslant 2$. The main result is Theorem 11.

Definition 2. Let b_n be the number of full binary trees with exactly n internal nodes, let j_n be the accumulated number of jumps in all those trees, and let d_n be the accumulated jump distance in all those jumps. In each tree there is a unique path that starts at the root and branches to the right at each node until it ends in a leaf; let r_n be the accumulated length of all those paths in the full binary trees with n internal nodes.

We will use r_n to derive formulas for j_n and d_n (Theorems 9 and 7).

Theorem 3. *For all* $n \geqslant 0$,

$$b_n = \frac{1}{n+1}\binom{2n}{n}.$$

Proof. The assertion clearly holds for $n = 0$; now let $n \geqslant 1$. The set of full binary trees with n internal nodes can be mapped to the set of binary trees with n nodes by stripping each full binary tree of its leaves. This mapping is a bijection, and so, b_n is equal to the number of binary trees with n nodes. That number is known to be the nth Catalan number, that is, the right-hand side of the asserted equation [1]. □

Remark 4. Theorem 3 can be proven directly by observing that, for $n \geqslant 1$,

$$b_n = \sum_{k=0}^{n-1} b_k b_{n-1-k}. \tag{1}$$

Indeed, since $n \geqslant 1$, the root is an internal node, so there is a left subtree with k internal nodes, $0 \leqslant k \leqslant n-1$, and a right subtree with $n-1-k$ internal nodes. For the left subtree there are b_k choices, for the right subtree b_{n-1-k}.

Theorem 5. *For all* $n \geqslant 0$,

$$r_n = b_{n+1} - b_n.$$

Proof. Induction on n. Start the induction step by observing that $r_n = \sum_{k=0}^{n-1}(r_k + b_k) \cdot b_{n-1-k}$, and use Eq. (1). □

Theorem 6. *For any full binary tree let* n *be the number of internal nodes*, d *the total distance jumped*, *and* r *the distance from the root of the rightmost node. Then*

$$d + r = n.$$

Proof. The assertion is true when $n = 0$. Let $n \geqslant 1$ in a full binary tree B. Let B_{L} and B_{R} be the left and right subtrees of B. Let n_{L} be the number of internal nodes in B_{L}, d_{L} the total distance jumped in B_{L}, and let r_{L} be the distance from the root of the rightmost node in B_{L}. Define n_{R}, d_{R}, and r_{R} analogously for B_{R}. Let d be the total distance jumped in B. Then $d = d_{\mathrm{L}} + r_{\mathrm{L}} + d_{\mathrm{R}} = n_{\mathrm{L}} + d_{\mathrm{R}}$, so $d + r = n_{\mathrm{L}} + d_{\mathrm{R}} + r = n_{\mathrm{L}} + d_{\mathrm{R}} + r_{\mathrm{R}} + 1 = n_{\mathrm{L}} + n_{\mathrm{R}} + 1 = n$. This completes a proof by induction. □

Theorem 7. *For all* $n \geqslant 0$,

$$d_n = nb_n - r_n.$$

Proof. Apply Theorem 6 to all the full binary trees with n internal nodes; sum all jump distances, all internal nodes, and all distances from the root of the rightmost nodes. □

Theorem 8. *For all* $n \geqslant 2$,

$$d_n = \binom{2n}{n-2}.$$

Proof. Let $n \geqslant 2$. By Theorem 7, $d_n = nb_n - r_n$, which, by Theorem 5, equals $(n+1)b_n - b_{n+1}$. Now apply Theorem 3 and manipulate factorial expressions. □

Theorem 9. *For all* $n \geqslant 1$,

$$j_n = 2d_{n-1} + r_{n-1}.$$

Proof. The assertion clearly holds for $n = 1$. Now let $n \geqslant 1$ and assume $j_{k+1} = 2d_k + r_k$ for all $k \in \{0, \ldots, n-1\}$. Then $d_n = \sum_{k=0}^{n-1}(d_k + r_k) \cdot b_{n-1-k} + b_k \cdot d_{n-1-k} = \sum_{k=0}^{n-1}(2d_k + r_k) \cdot b_{n-1-k} = \sum_{k=0}^{n-1} j_{k+1}b_{n-1-k} = \sum_{k=1}^{n} j_k b_{n-k} = \sum_{k=0}^{n} j_k b_{n-k}$. Use this equality together with Eq. (1) and Theorem 5 to perform the induction step $j_{n+1} = \sum_{k=0}^{n} b_k j_{n-k} + \sum_{k=0}^{n} j_k b_{n-k} + \sum_{k=1}^{n} b_k b_{n-k} = 2 \cdot (\sum_{k=0}^{n} j_k b_{n-k}) + b_{n+1} - b_n = 2d_n + r_n$. □

Theorem 10. *For all* $n \geqslant 2$,

$$j_n = \binom{2n-1}{n-2}.$$

Proof. Let $n \geqslant 2$. By Theorem 9, $j_n = 2d_{n-1} + r_{n-1}$, which, by Theorems 7 and 5, equals $(2n-1)b_{n-1} - b_n$. Now apply Theorem 3 and manipulate factorial expressions. □

Theorem 11. *On the set of the full binary trees with* $n \geqslant 2$ *internal nodes the average number of jumps (=number of jumps per internal node) is*

$$\frac{j_n}{nb_n} = \frac{1}{2} - \frac{1}{2n},$$

and the average jump distance is

$$\frac{d_n}{j_n} = 2 - \frac{4}{n+2}.$$

Proof. Theorems 3, 10, and 8. □

3. Jumps in real root isolation

We now consider trees that the Descartes method for real root isolation associates with polynomials [2]. We consider three sets of polynomials. The first set consists of 100 random polynomials for each of the degrees $10, 20, 30, \ldots, 300$; the coefficients are 20-bit integers that are generated uniformly at random. The second set consists of the Chebyshev polynomials of the first kind, degrees 2–100. The third set consists of the Mignotte-polynomials $x^m - 2(5x-1)^2$ for $3 \leqslant m \leqslant 50$. For each set of polynomials we consider the corresponding multi-set of trees with a given number n of internal nodes. The diagrams in Fig. 2 show, for each n, the average number of jumps and the average jump distance. For each set of polynomials the averages are very close to the averages in Theorem 11.

Our analytical and empirical results suggest that, in the algorithm of Rouillier and Zimmermann [2], the number of jumps, and also the jump distance, will usually be very small.

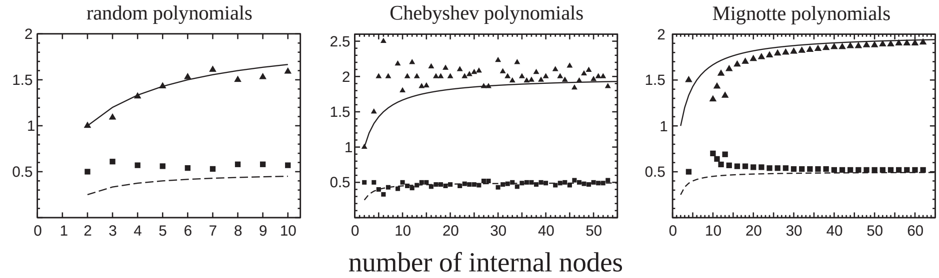


Fig. 2. In all three diagrams the continuous line shows the average jump distance for the set of all full binary trees, and the dashed line shows the corresponding average number of jumps. The triangles and squares denote empirical averages that are specific to the multi-sets of search trees for the respective sets of polynomials; triangles stand for the average jump distance, squares for the average number of jumps.

Acknowledgements

G.E. Collins simplified an earlier version of the manuscript by contributing Theorem 6.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, MIT Press, Cambridge, MA, McGraw-Hill, New York, 2001.

[2] F. Rouillier, P. Zimmermann, Efficient isolation of polynomial real roots, J. Comput. Appl. Math. 162 (2003) this issue.