Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 369 (2006) 442 – 447
Theoretical Computer Science
www.elsevier.com/locate/tcs

Note

Cost distribution of the Chang–Roberts leader election algorithm and related problems

Wei-Mei Chen*, 1

Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan

Received 23 November 2005; received in revised form 30 June 2006; accepted 10 August 2006

Communicated by H. Prodinger

Abstract

A detailed probabilistic analysis is proposed of the total number of messages of the Chang–Roberts leader election algorithm. The cost is shown to be closely related to the total path length in random recursive trees, the total left-path length in increasing binary trees and the major cost of an in situ permutation algorithm.

Keywords: Leader election; Extrema finding; Maxima finding; Recursive trees; Increasing binary trees; In situ permutation; Left-to-right minima

1. Introduction

The leader election (or extrema finding, or maxima finding) problem in a distributed system is to find exactly one processor as a coordinator or an initiator for performing some special tasks. It is a fundamental problem for achieving fault tolerance and reducing resource utilization under distributed networks and is studied extensively in different computation models; see [14,20] for more information.

LeLann [13] proposed the first leader election algorithm on unidirectional rings with message complexity $O(n^2)$ where n is the number of processors. Later, Chang and Roberts [2] suggested an improved solution with expected message complexity $O(n \log n)$. Then several $O(n \log n)$ algorithms were presented either for unidirectional rings [4,6,17] or bidirectional rings [7] to reduce the constant in the $O(n \log n)$ upper bound.

The Chang–Roberts algorithm is simple but prototypical for the researches of the leader election problem on rings [9,14,20]. The algorithm identifies the leader by passing $O(n \log n)$ messages on average in a unidirectional ring, on the assumption that each of the processors holds a distinct identity and the number of processors, n, is unknown. For the reader's convenience, the decentralized algorithm is described as follows.

Input: All processors $P_1, P_2, \ldots, P_n$ form a unidirectional ring $\mathcal{R}$.
Output: The processor with the minimum identity declares itself the leader.

* Tel.: +886 2 27376379; fax: +886 2 27376424.
E-mail address: wmchen@mail.ntust.edu.tw.
1 Partially supported by National Science Council under the Grant NSC92-2213-E-036-018 and NSC93-2213-E-036-002.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.08.019

```
Elect-Leader(R)
    for all P_i do
        m_i := the identity of P_i
        send m_i to the next processor
        t_i := ∞
        while (t_i ≠ m_i)      // Only the leader terminates the algorithm //
            receive t_i from the previous processor
            if t_i < m_i, then m_i := t_i; send m_i
End.
```

At the end, the leader informs all the processors about its identity.

In the above algorithm, every processor starts by passing its own identity to its next neighbor along a prescribed direction. Whenever a processor receives an identity from its previous neighbor, it either sends the message along the direction of the ring if the identity received is the smallest number among all it has ever seen, discards the message if the identity received is greater than its own, or declares itself the leader if the identity received matches its own. Clearly, the major cost of the algorithm is the number of messages sent by all the processors.

The expected cost of the Chang–Roberts algorithm was derived in the original paper [2]. The aim of this paper is to refine their analysis, to give more insight into the algorithm and to prove the bijections between the cost and a few other combinatorial quantities already known in the literature. More precisely, we first give a deeper probabilistic analysis of the cost, including the expected value, variance and the limiting distribution. We then show that there are bijections between the cost of the Chang–Roberts algorithm and three quantities in different structures: the total path length in random recursive trees, the total left-path length in increasing binary trees, and the major cost of the in situ permutation algorithm (see [11,12]).

2. Analysis

Assume that all $n!$ possible permutations of n identities are equally likely. Let X_n denote the number of messages used by the Chang–Roberts algorithm on a ring of size n.

Theorem 1. *The number X_n of messages sent by the Chang–Roberts leader election algorithm on a ring of size n satisfies the following.*

(i) *Mean*

$$\mathbb{E}(X_n) = nH_n = n \log n + \gamma n + \mathrm{O}(1).$$

(ii) *Variance*

$$\mathbb{V}(X_n) = \left(2 - \frac{\pi^2}{6}\right) n^2 + \mathrm{O}(n \log n).$$

(iii) *Convergence in distribution*

$$\frac{X_n - \mathbb{E}(X_n)}{n} \xrightarrow{\mathrm{d}} X,$$

where $\gamma = 0.57721\ldots$ is Euler's constant, H_n is the nth harmonic number, $X \stackrel{\mathrm{d}}{=} UX + (1-U)X^* + U \log U + (1-U)\log(1-U) + U$, U is a uniform $(0,1)$ random variable, $X \stackrel{\mathrm{d}}{=} X^*$, and X, X^* and U are mutually independent.

Here the symbol $\stackrel{\mathrm{d}}{=}$ denotes equivalence in distribution, and the symbol $\xrightarrow{\mathrm{d}}$ denotes convergence in distribution. Note that the standard deviation is linear, which is not very far away from the mean, especially for moderate values of n. Also the limiting distribution is not normal, intuitively due to the large standard deviation.

Consider a ring of n processors, say $P_1, P_2, \ldots, P_n$. Without loss of the generality, suppose that all the processors proceed along the clockwise direction and the identities are labeled by the set $\{1, 2, \ldots, n\}$. For convenience, let $[\alpha_1\ \ \alpha_2\ \ \ldots\ \ \alpha_n]$ denote the ring of size n with identities $\alpha_1, \ldots, \alpha_n$, where we list the identities in the counter-clockwise direction with $\alpha_n = 1$.

Lemma 1. *Let $F_n(y)$ be the probability generating function of X_n. Then $F_n(y)$ satisfies the recurrence*

$$F_n(y) = \frac{1}{n-1}\sum_{k=1}^{n-1} F_k(y)F_{n-k}(y)y^k \quad (n\geqslant 2), \tag{1}$$

with $F_1(y) = y$.

Proof. Observe that, with the exception of the first pass, every processor sends a message to its neighbor when it receives the smallest number among all that has been seen. For the ring $[\alpha_1\ \ \alpha_2\ \ \ldots\ \ 1]$, the number of messages sent by the processor with identity α_i is exactly the number of left-to-right minima of the permutation $\alpha_i \ldots \alpha_{n-1}1$, namely the number of indices j such that $\alpha_j = \min\{\alpha_k | i \leqslant k \leqslant j\}$. Thus, the number of messages sent on the ring $[\alpha_1\ \ \alpha_2\ \ \ldots\ \ 1]$ is equal to the total number of left-to-right minima for all the suffixes of the permutations $\alpha_1\alpha_2\ldots\alpha_{n-1}1$ (i.e. $\alpha_1\alpha_2\ldots\alpha_{n-1}1, \alpha_2\alpha_3\ldots\alpha_{n-1}1, \ldots, \alpha_{n-1}1$, and 1).

Let $S_{i,j}$ be the total number of left-to-right minima of all the suffixes of the permutations $\alpha_i\alpha_{i+1}\ldots\alpha_j$, where α_j is the minimum of the set $\{\alpha_i, \alpha_{i+1}, \ldots, \alpha_j\}$. If the minimum of the set $\{\alpha_1, \alpha_2, \ldots, \alpha_{n-1}\}$ is α_k, then $S_{1,n} = S_{1,k} + k + S_{k+1,n}$. This leads to the equality in distribution $X_n \overset{\mathrm{d}}{=} X_k + k + X^*_{n-k}$, with X^*_{n-k} distributed as X_{n-k} and independent of X_k. $\square$

The recurrence (1) with a different initial condition appeared in the study of random recursive trees [18]. A recursive tree (or increasing tree) is a labeled rooted tree in which the sequence of labels along any path starting at the root is increasing [1]. Let Y_n denote the total path length of a random recursive tree of n nodes. By random recursive trees, we assume that all $(n-1)!$ recursive trees of n labels are equally likely. Then the probability generating function $G_n(y) := \mathbb{E}(y^{Y_n})$ can be expressed as follows:

$$G_n(y) = \frac{1}{n-1}\sum_{k=1}^{n-1} G_k(y)G_{n-k}(y)y^k \quad (n\geqslant 2),$$

with $G_1(y) = 1$ [3,15]. By mathematical induction, we have $F_n(y) = y^n G_n(y)$ for $n\geqslant 1$. Thus X_n has the same distribution as $Y_n + n$.

Lemma 2. *For $n\geqslant 1$,*

$$X_n \overset{\mathrm{d}}{=} Y_n + n.$$

By considering $L_n(y) := F_{n+1}(y)$, we see that (1) is equivalent to

$$L_n(y) = \frac{1}{n}\sum_{k=0}^{n-1} L_k(y)L_{n-1-k}(y)y^k \quad (n\geqslant 1),$$

with $L_0(y) = y$. This recurrence, with the different initial condition $L_0(y) = 1$, enumerates the total path length in random increasing binary trees and the cost, called the "parameter a", of the in situ permutation algorithm (see [11,12,16]). Let Z_n be the total left path length of a random increasing binary tree. And the relation between X_n and Z_n can be expressed as below.

Lemma 3. *For $n\geqslant 1$,*

$$X_{n+1} \overset{\mathrm{d}}{=} Z_n + 2n + 1.$$

Proof. By induction we have, for $n \geqslant 1$,

$$y^{2n+1}L_n(y) = \frac{1}{n}\sum_{k=1}^{n}(y^{2(k-1)+1}L_{k-1}(y))(y^{2(n-k)+1}L_{n-k}(y))y^k = F_{n+1}(y).$$

This completes the proof. □

Note that X_n is also equi-distributed with (by symmetry of permutations) n plus the sum of ranks of the records (or left-to-right maxima) in a sequence of independent and identically distributed (i.i.d.) sequence of random variables with a common continuous distribution.

Proof of Theorem 1 (*sketch*). Denote the mean of X_n by $M_n := \mathbb{E}(X_n) = F_n'(1)$. Then, by (1) we have $M_1 = 1$ and M_n obeys the full-history recurrence relation

$$M_n = \frac{2}{n-1}\sum_{k=1}^{n-1} M_k + \frac{n}{2} \quad \text{for } n \geqslant 2.$$

By iterating the difference $(n-1)M_n - (n-2)M_{n-1}$, we deduce that $M_n = nH_n$, where H_n is the nth harmonic number. The variance of X_n, $\mathbb{V}(X_n)$, can be similarly computed by the relation $\mathbb{V}(X_n) = F_n''(1) + M_n - M_n^2$.

The convergence in distribution follows from that for recursive trees proved in [15,3]; see also [8] for different proofs. □

Note that the bivariate generating function $F(z, y) := \sum_n F_{n+1}(y)z^n$ satisfies the differential equation

$$\frac{\partial}{\partial z}F(z, y) = yF(yz, y)F(z, y),$$

with the initial condition $F(0, y) = y$. Then, by taking successive derivatives with respect to y and then substituting $y = 1$, the mean and all higher factorial moments can be derived by solving the associated differential equations. For more details, see [3].

3. Bijections

There exists a well-known bijection between permutations and increasing binary trees [19, pp. 23–25]. In the proof of Lemma 1, we see that the cost used by the Chang–Roberts algorithm on the ring $[\alpha_1\ \alpha_2\ \ldots\ \alpha_{n-1}\ 1]$ is exactly the total number of left-to-right minima of all the suffixes of the permutation $\alpha_1\alpha_2\ldots\alpha_{n-1}1$. In the following we give the constructive bijections between the cost of the Chang–Roberts algorithm and the cost of the in situ permutation algorithm, the left total path length of increasing binary trees and the total path length of recursive trees, respectively. Fig. 1 shows the close relationships among the related problems.

3.1. The cost of the in situ permutation algorithm

The in situ permutation algorithm [11,12,16] is to rearrange n items according to a given permutation $x_1x_2\ldots x_n$ using a bounded amount of auxiliary memory. A cycle leader of a permutation is the smallest number in its own cycle. The critical operation for solving the in situ permutation problem is to locate all cycle leaders of the permutation. Once a leader is detected in a cycle, the remaining job is to carry out the desired permutation. The major cost for finding all cycle leaders can be expressed as $n + a$, where $a \triangleq |\{(i, j) : 1 \leqslant i < j \leqslant n, x_i = \min\{x_i, \ldots, x_j\}\}|$; see [5,10,12,16] for further information. Essentially, given a permutation, computing the parameter a is similar to counting the number of right-to-left minima of all prefixes of the permutation. See Fig. 2 for an example where the bijective correspondence is constructed between the cost of Chang–Roberts algorithm for the ring $[\alpha_1\ \alpha_2\ \ldots\ \alpha_{n-1}\ 1]$ and the parameter a of the in situ permutation algorithm for the permutation $\beta_{n-1}\beta_{n-2}\ldots\beta_1$ where $\beta_i = \alpha_i - 1$ for $1 \leqslant i \leqslant n-1$.

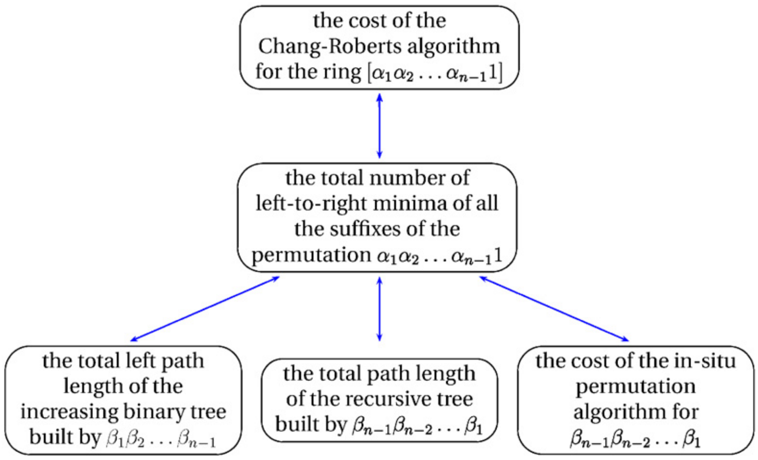

Fig. 1. The bijective mappings between the number of messages sent by the Chang–Roberts algorithm on rings and the related problems, respectively (where $\beta_i = \alpha_i - 1$ for $1 \leqslant i \leqslant n-1$).

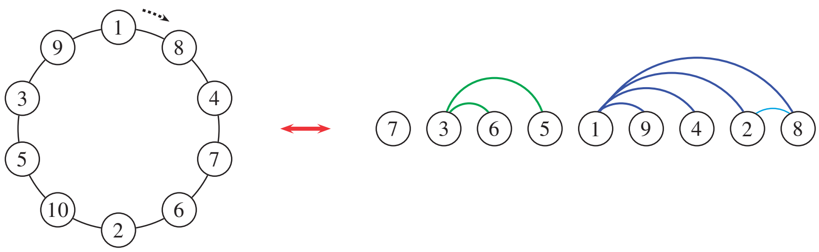

Fig. 2. The number of messages sent by the Chang–Roberts algorithm on the ring [9 3 5 10 2 6 7 4 8 1] is 26 and the parameter a of the in situ permutation algorithm for the permutation 736519428 is 7.

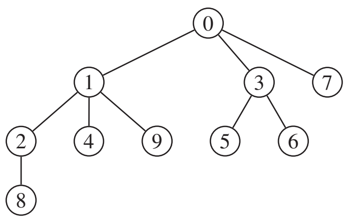

Fig. 3. The total path length of the recursive tree built by the permutation 736519428 is 16.

3.2. The total path length of recursive trees

By [19, pp. 25], given a permutation of $\{1, 2, \ldots, n\}$, we can construct a recursive tree with $n+1$ nodes with label set $\{0, 1, \ldots, n\}$ by scanning the permutation from right to left: defining the parent of node i to be the rightmost number j with $j < i$ and the parent of all left-to-right minima to be the children of the node 0. Fig. 3 shows an example where the bijective correspondence is constructed between the cost of Chang–Roberts algorithm for the ring $[\alpha_1\ \alpha_2\ \ldots\ \alpha_{n-1}\ 1]$ and the total path length of the recursive tree built by the permutation $\beta_{n-1}\beta_{n-2}\ldots\beta_1$ where $\beta_i = \alpha_i - 1$ for $1 \leqslant i \leqslant n-1$.

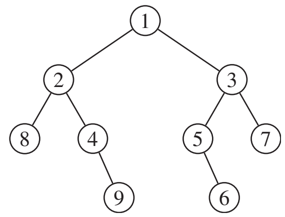

Fig. 4. The total left path length of the binary increasing tree of the sequence 824915637 is 7.

3.3. The total left path length of increasing binary trees

Given a permutation, we can construct an increasing binary tree by the divide-and-conquer approach introduced in [19, pp. 23–24]: picking the smallest number as the root, splitting the permutation into the left and right parts by removing the smallest number, and finally recursively performing the same steps in the remaining two parts for the left and right subtrees, respectively. Fig. 4 demonstrates a specific example. Thus, we describe a constructive bijection between the cost of Chang–Roberts algorithm for the ring $[\alpha_1\ \ \alpha_2\ \ \ldots\ \ \alpha_{n-1}\ \ 1]$ and the total left path length of the increasing binary tree built by the sequence $\beta_1\beta_2\ldots\beta_{n-1}$ where $\beta_i = \alpha_i - 1$ for $1 \leqslant i \leqslant n-1$.

References

[1] F. Bergeron, P. Flajolet, B. Salvy, Varieties of increasing trees, in: CAAP ’92 (Rennes, 1992), Lecture Notes in Computer Science, Vol. 581, Springer, Berlin, 1992, pp. 24–48.
[2] E. Chang, R. Roberts, An improved algorithm for decentralized extrema-finding in circular configurations of processes, Comm. ACM 22 (5) (1979) 281–283.
[3] R.P. Dobrow, J.A. Fill, Total path length for random recursive trees, Combinatorics, Probab. Comput. 8 (4) (1999) 317–333.
[4] D. Dolev, M. Klawe, M. Rodeh, An $O(N \log N)$ undirectional distributed algorithm for extrema-finding in a circle, J. Algorithms 3 (3) (1982) 245–260.
[5] F.E. Fich, J.I. Munro, P.V. Poblete, Permuting in place, SIAM J. Comput. 24 (2) (1995) 266–278.
[6] L. Higham, T.M. Przytycka, A simple, efficient algorithm for maximum finding on rings, Inform. Process. Lett. 58 (6) (1996) 319–324.
[7] D.S. Hirschberg, J.B. Sinclair, Decentralized extrema-finding in circular configurations of processors, Comm. ACM 23 (1980) 627–628.
[8] H.-K. Hwang, R. Neininger, Phase change of limit laws in the quicksort recurrence under varying toll functions, SIAM J. Comput. 31 (6) (2002) 1687–1722.
[9] A. Itai, M. Rodeh, Symmetry breaking in distributed networks, Inform. and Comput. 88 (1) (1990) 60–87.
[10] J. Keller, A heuristic to accelerate in situ permutation algorithms, Inform. Process. Lett. 81 (3) (2002) 119–125.
[11] P. Kirschenhofer, H. Prodinger, R.F. Tichy, A contribution to the analysis of in situ permutation, Glas. Mat. 22 (42) (1987) 269–278.
[12] D.E. Knuth, Mathematical analysis of algorithms, in: Information Processing 71 (Proc. IFIP Congress, Ljubljana, 1971), Foundations and Systems, Vol. 1, North-Holland, Amsterdam, 1972, pp. 19–27.
[13] G. LeLann, Distributed systems—towards a formal approach, in: Information Processing 77, North-Holland, Amsterdam, 1977, pp. 155–160.
[14] N. Lynch, Distributed Algorithms, Morgan Kaufmann, Los Altos, CA, 1997.
[15] H.M. Mahmoud, Limiting distributions for path lengths in recursive trees, Probab. Eng. Inform. Sci. 5 (1) (1991) 53–59.
[16] A. Panholzer, H. Prodinger, M. Riedel, Permuting in place: analysis of two stopping rules, J. Algorithms 51 (2) (2004) 170–184.
[17] G.L. Peterson, An $O(n \log n)$ unidirectional algorithm for the circular extrema problem, ACM Trans. Programming Lang. Sys. 4 (4) (1982) 758–762.
[18] R.T. Smythe, H.M. Mahmoud, A survey of recursive trees, Theoret. Probab. Math. Statist. 51 (1995) 1–27.
[19] R.P. Stanley, Enumerative Combinatorics, Vol. 1, Cambridge University Press, Cambridge, 1997.
[20] G. Tel, Introduction to Distributed Algorithms, Cambridge University Press, Cambridge, 2000.