On the distribution of the number of computations in any finite number of subtrees for the stack algorithm

Johannesson, Rolf; Zigangirov, Kamil

Published in:
IEEE Transactions on Information Theory

1985

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Correspondence

On the Distribution of the Number of Computations in Any Finite Number of Subtrees for the Stack Algorithm

ROLF JOHANNESSON AND KAMIL SH. ZIGANGIROV

Abstract—**Multitype branching processes have been employed to determine the stack algorithm computational distribution for one subtree. These results are extended here to the distribution of the number of computations in any finite number of subtrees. Starting from the computational distribution for K-1 subsequent subtrees, a recurrent equation for the distribution for K subsequent subtrees is determined.**

I. Introduction

It is well known [1]–[3] that the curse of sequential decoding is that its behavior is limited by a computational distribution (conditioned on correct decoding), which is asymptotically Pareto. Until now little attention has been paid to the problem of determining the computational distribution for a small number of computations. In our previously published papers [4], [5] we employed multitype branching processes to determine the stack algorithm computational distribution for the first incorrect subtree. The results in [5] were obtained for random tree codes, while in [4] we considered the class of binary, rate 1/2, complementary + random tree codes, a ficticious entity that is a reasonable model for constant convolutional codes that have column distance $d_0 = 2$.

In a complementary + random tree code the channel input symbols on the transmitted path are all zeros, and the channel input symbols on the incorrect branches stemming from nodes on the correct path are all ones. For all other branches, each channel symbol is chosen independently and according to a specified probability distribution. In this work we extend the results in [4] and determine the probability distribution for the number of computations in any finite number of subsequent subtrees.

II. Preliminaries

Let C_n, $n = 1,2,3,\cdots$, denote the number of computations made by the sequential decoder in order to decode the nth correct node. In Fig. 1 we show a partially explored code tree with $C_1 = 4$. For a tree of unbounded length the random variables $C_1, C_2, C_3, \cdots$ have the same distribution, but they are certainly not independent.

Let M_n be the cumulative metric for the first n branches of the correct path; i.e.,

$$M_n = \sum_{k=1}^{n} z_k, \tag{1}$$

where z_k is the branch metric for the kth branch. Let D_n, $n = 0,1,2,\cdots$, be the difference between the cumulative metric

Manuscript received July 15, 1983; revised June 18, 1984. This research was supported in part by the Royal Swedish Academy of Sciences, The Royal Swedish Academy of Engineering Sciences, and the USSR Academy of Sciences. This work was presented at the International Workshop "Convolutional Codes: Multi-User Communications", Sochi, USSR, May 31–June 5, 1983.

R. Johannesson is with the Department of Computer Engineering, University of Lund, P.O. Box 725, S-220 07 Lund, Sweden.

K. Sh. Zigangirov is with the Institute for Problems of Information Transmission, USSR Academy of Sciences, 19 Ermolovoi St., Moscow GSP-4, USSR.

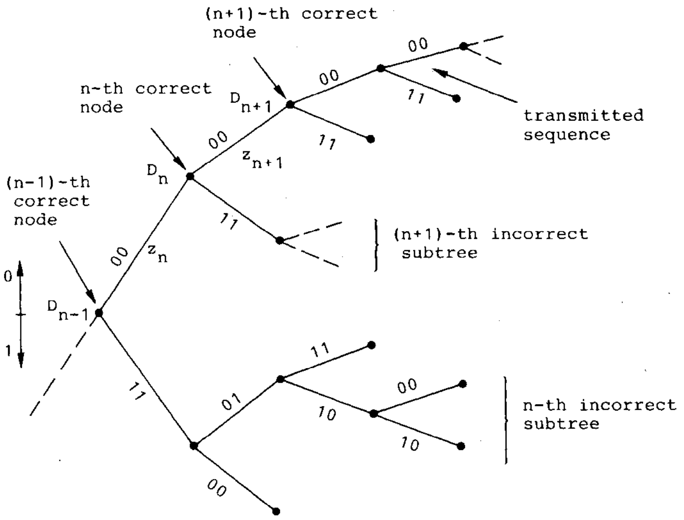

Fig. 1. Part of a complementary + random code tree partially explored by a sequential decoder.

and the smallest succeeding value; i.e.,

$$D_n = M_n - M_{\min, n}, \tag{2}$$

where

$$M_{\min, n} = \min\{M_n, M_{n+1}, \cdots\}. \tag{3}$$

The nonnegative random variables D_n all have the same well-known distribution [4], [6] since the metrics for the correct path stemming from the nth node have the same statistical character for all n.

Finally, let C_n^N be the total number of computations in the nth, $(n+1)$th,$\cdots$, Nth subtree

$$C_n^N = \sum_{k=n}^{N} C_k. \tag{4}$$

III. Recurrent Equation for Determining the Distribution of C_n^N

Let $F_n^N(r,i,k)$ be the conditional distribution for the random variable C_n^N on the condition that $D_n = i$ and $z_n = k$; i.e.,

$$F_n^N(r,i,k) = P\left[C_n^N = r | D_n = i, z_n = k\right]. \tag{5}$$

By dividing the number of computations in the $n, n+1,\cdots, N$th subtrees into two parts, C_n and C_{n+1}^N, we have

$$F_n^N(r,i,k) = \sum_{s=1}^{r-1} P[C_n = s | D_n = i, z_n = k] \cdot P\left[C_{n+1}^N = r - s | D_n = i, z_n = k\right]. \tag{6}$$

The conditional probability distribution for C_{n+1}^N can be expanded as

$$P\left[C_{n+1}^N = r - s | D_n = i, z_n = k\right] = \sum_{j=0}^{\infty} \sum_{l} P\left[C_{n+1}^N = r - s | D_{n+1} = j, z_{n+1} = l, D_n = i, z_n = k\right] \cdot P[D_{n+1} = j, z_{n+1} = l | D_n = i, z_n = k], \tag{7}$$

where the variable l is summed over all branch metric values.

The random variable C_{n+1}^N conditioned on D_{n+1} and z_{n+1} is independent of D_n and z_n. Furthermore, the random variables D_{n+1} and z_{n+1} conditioned on D_n are independent of z_n. Hence we obtain the recurrent equation for $F_n^N(r,i,k)$

$$F_n^N(r,i,k) = \sum_s \sum_j \sum_l f(s,i,k) \cdot F_{n+1}^N(r-s,j,l)\, g(j,l,i), \tag{8}$$

where $n < N$,

$$f(s,i,k) = F_N^N(s,i,k) \tag{9}$$

and

$$g(j,l,i) = P[D_{n+1} = j, z_{n+1} = l | D_n = i]. \tag{10}$$

The conditional probability distribution $g(j,l,i)$ is easily calculated as follows:

$$\begin{aligned} g(j,l,i) &= P[D_{n+1} = j,\, z_{n+1} = l | D_n = i] \\ &= P[D_n = i | D_{n+1} = j,\, z_{n+1} = l]\, P[D_{n+1} = j] \\ &\quad \cdot P[z_{n+1} = l] / P[D_n = i], \end{aligned} \tag{11}$$

where we have used the fact that the random variables D_{n+1} and z_{n+1} are statistically independent. Let us assume that we have the branch metric set $\{+1, -4, -9\}$, c.f. [4]; then

$$P[D_n = i | D_{n+1} = j,\, z_{n+1} = l] = \begin{cases} 1, & i = j = 0, l = 1 \\ 1, & i = j - l \geq 0 \\ 0, & \text{elsewhere.} \end{cases} \tag{12}$$

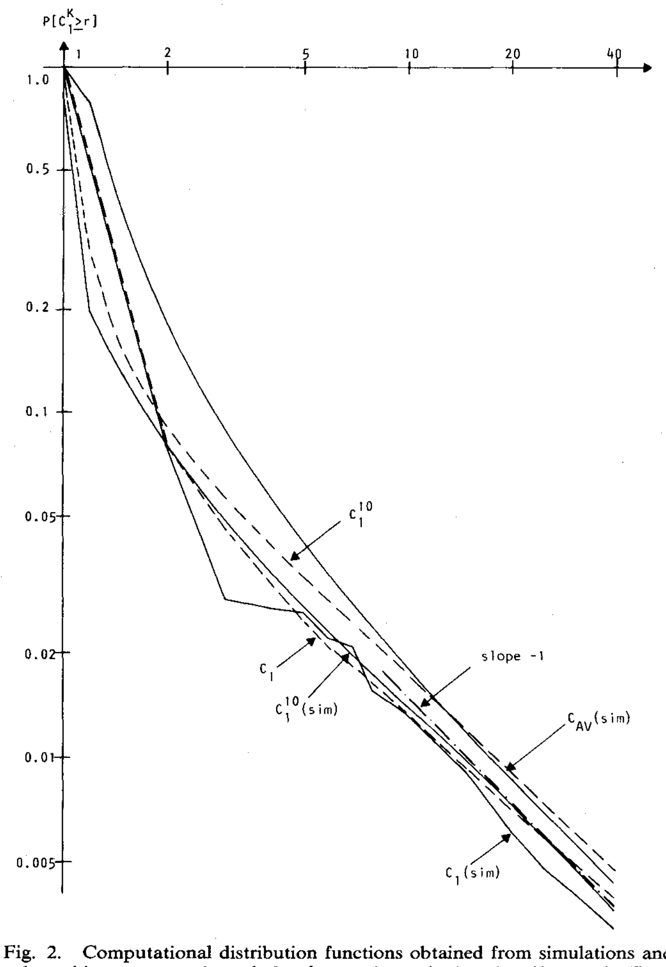

Fig. 2. Computational distribution functions obtained from simulations and branching processes bounds for the metric set $\{+1, -4, -9\}$, *viz.* the Fano metric.

Starting from the conditional probability distribution $f(s,i,k)$, which can be determined by the methods given in [4], we can use (8) and recursively obtain the conditional probability distribution $F_n^N(r,i,k)$ for any n, $0 < n < N$. Finally, we calculate the probability distribution for the number of computations in $K \geq 1$ subtrees as

$$P[C_1^K = r] = \sum_i \sum_k F_1^K(r,i,k)\, h(i,k), \tag{13}$$

where

$$h(i,k) = P[D_n = i, z_n = k] = P[D_n = i] P[z_n = k]. \tag{14}$$

IV. Discussion of Numerical Results and Simulations

The problem of resolving ties among the cumulative metrics is discussed in detail in [4]. When there was no obvious way to resolve ties the pessimistic policy "in case of ties extend an incorrect node first" was used to obtain an upper bound on the distribution function for C_1, and the optimistic policy "in case of ties extend the correct node first" was used to obtain a lower bound. Because of the close agreement between the simulated C_1-curves and the corresponding lower bounds in [4], we will use these lower bounds as our conditional distribution $f(s,i,k)$ in (8).

By combining the branching processes methods described in [4] and the recurrent equation (8), we evaluate theoretical lower bounds on the distribution function of the random variable C_1^{10}, $P[C_1^{10} \geq r]$ for the case where transmission takes place over a binary symmetric channel (BSC) with crossover probability $p = 0.045$. Since we are only considering rate $R = 1/2$ codes, this crossover probability corresponds to transmission at the computational cutoff rate R_0. The bounds are evaluated for the Fano

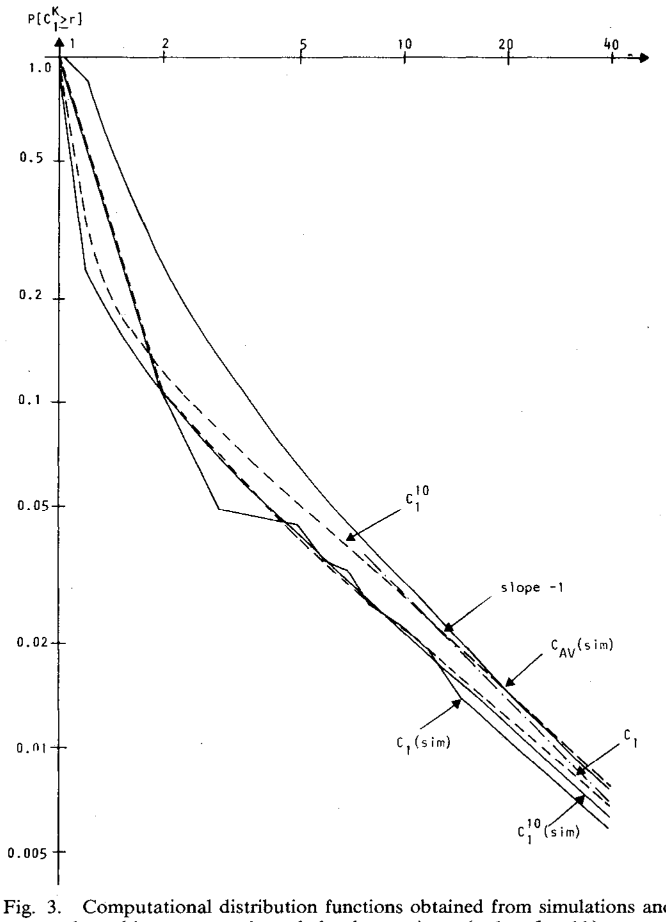

Fig. 3. Computational distribution functions obtained from simulations and branching processes bounds for the metric set $\{+1, -5, -11\}$.

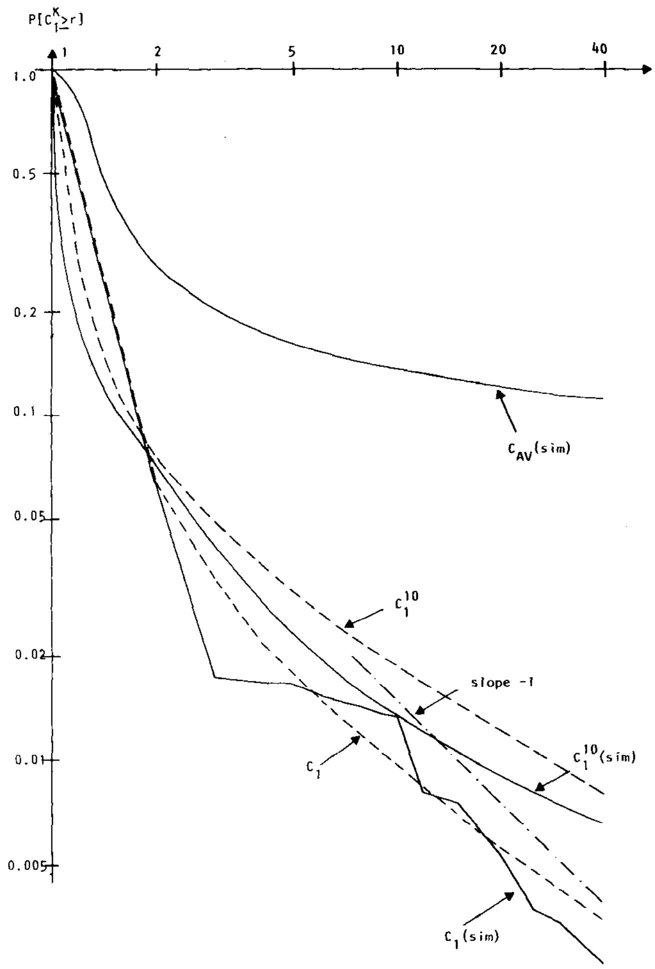

Fig. 4. Computational distribution functions obtained from simulations and branching processes bounds for the metric set $\{+1, -3, -7\}$. This set is pessimistic for the simulated channel.

metric set $\{+1, -4, -9\}$ and for the "unmatched" sets $\{+1, -5, -11\}$ and $\{+1, -3, -7\}$. The bounds are shown in Figs. 2–4 together with lower bounds on the distribution function of the random variable C_1, $P[C_1 \geq r]$ and simulations of $P[C_1 \geq r]$, $P[C_1^{10} \geq r]$; and $P[C_{AV} \geq r]$ for the fixed nonsystematic optimum distance profile (ODP) convolutional code with memory length $M = 23$ and $d_\infty = 25$ [7]. As many as 100 000 frames consisting of 500 information symbols augmented by a tail of $M = 23$ zeros were transmitted at rate $R = R_0$ and decoded with different metrics.

Since our fixed code has a better distance profile than a random code, the simulated curves are slightly below the corresponding calculated values. We also notice the close agreement between all curves for the Fano metric set $\{+1, -4, -9\}$ and the well-known Pareto asymptote (slope -1).

Finally, it is interesting to notice the wide gap between the C_1 and C_{AV} curves for the metric set $\{+1, -3, -7\}$. When estimating the crossover probability it does not pay to be pessimistic!

References

[1] I. M. Jacobs and E. R. Berlekamp, "A lower bound to the distribution of computation for sequential decoding," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 167–174, Apr. 1967.

[2] J. E. Savage, "The distribution of sequential decoding computation time," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 145–157, Apr. 1966.

[3] F. Jelinek, "An upper bound on moments of sequential decoding effort," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 140–149, Jan. 1969.

[4] R. Johannesson, "On the distribution of computation for sequential decoding using the stack algorithm," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 323–331, May 1979.

[5] N. D. Vvdenskaja and K. Sh. Zigangirov, "On the computation time distribution of the sequential decoding," *Probl. Peredach. Inform.*, vol. 5, pp. 78–80, 1969.

[6] J. L. Massey *et al.*, "Certain infinite Markov chains and sequential decoding," *Discrete Math.*, vol. 3, pp. 163–175, Sept. 1972.

[7] R. Johannesson and E. Paaske, "Further results on binary convolutional codes with an optimum distance profile," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 264–238, Mar. 1978.

A New Class of Check-Digit Methods for Arbitrary Number Systems

H. PETER GUMM

Abstract—**For arbitrary number systems we present a new check-digit method that detects all single-digit errors and all transpositions of adjacent digits using a single check digit from the given number system. In previous methods at least one type of transpositional error had to remain undetected. The key to this method lies in using the dihedral groups together with appropriate transformations in the important cases, where the numbers are represented in base $2r$ with r odd.**

I. Introduction

Empirical studies have shown [9], [10] that the most common typing errors that occur when data are entered on a keyboard are

- single-digit errors (one digit wrong)
- format errors (one digit inserted or left out)
- transpositions (interchanging of two adjacent digits).

To detect such errors, the original string of data is supplied with one check digit, where digit now means a "digit" in the chosen number system, i.e., the check digit is numerical for numerical data and may be alphanumeric for alphanumeric data. Various check-digit methods have been designed for the decimal number system. Each method is able to detect all "single-error mistakes," but they fail to detect all transposition errors. At least one (and often not more than one) erroneous transposition is undetectable with those methods, see [1]–[3].[1]

The format errors principally cannot all be detected by only one check digit, and every method that detects single-digit errors will automatically detect about 90 percent of all format errors. The use of two check digits as proposed in [3] and [11] is not advised since the previously mentioned studies have also shown that the absolute number of errors that occur roughly doubles when the number of digits increases by two. Thus generally specifying and checking for a fixed format seems appropriate to detect format errors. We therefore concentrate on methods to detect single-digit errors and transposition errors.

Clearly, for numbers to the base 2 such a check digit method is impossible since the numbers 00, 01, 10 would have to be supplied with mutually different check digits from the set $\{0, 1\}$.

If a number n has the digits $d_k, d_{k-1}, \cdots, d_1$ in base r, i.e., $n = \Sigma d_i r^{i-1}$, then using $p = -\Sigma d_i$ (in modulo r arithmetic) for a check digit will detect every single-digit error. The secured number then has the digit representation $d_k, d_{k-1}, \cdots, d_1, p$, and

Manuscript received August 26, 1983; revised January 16, 1984.

The author is with the Gesellschaft für Strahlen und Umweltforschung München-Neuherberg, Institut für Medizinische Informatik und Systemforschung, D-8042 Oberschleissheim, Federal Republic of Germany.

[1] After this paper was written, I was informed that J. Verhoeff in his thesis "Error correcting decimal codes," (*Mathematical Centre Tracts*, no. 29, Amsterdam: Mathematisch Centrum, 1969) has presented a different check-digit method. Our method, nevertheless, appears to be more general and simpler from a conceptual as well as an implementational standpoint.

0018-9448/85/0100-0102$01.00 ©1985 IEEE