

The Bidirectional Stack Algorithm

Vojin Šenk and Predrag Radivojac

University of Novi Sad, Faculty of Engineering
Trg Dositeja Obradovića 6, 21000 Novi Sad, Yugoslavia
Email: ram_senk@uns.ns.ac.yu

Abstract — A new procedure for sequential decoding of trellis codes, whose decoding effort is unaffected by single correctable bursts of errors of desired length (but not longer than the code constraint length) is simulated and compared with the classical stack algorithm. The results show an immense reduction in decoding effort.

I. INTRODUCTION

Two of the most powerful techniques for decoding trellis codes are the suboptimum tree search of sequential decoding, such as the stack (or ZJ) algorithm, and the optimal trellis search of Viterbi decoding (VA). Both the abovementioned techniques attempt to find the best path $\hat{\mathbf{I}} = [\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \dots, \hat{\mathbf{i}}_l, \dots, \hat{\mathbf{i}}_L]$, $\dim(\hat{\mathbf{i}}_l) = K$, through a graph (tree or trellis) in which the branches are assigned "branch metrics" μ_l . The branch metrics are cumulative so that, over a graph of length $L + \mathcal{M}$ branches (the transmission is assumed to be framed, i.e. it is terminated by a tail of \mathcal{M} branches, \mathcal{M} being the code memory), the objective of both techniques is to find the path $\hat{\mathbf{I}}$ for which the total metric is maximum over all possible transmitted paths. In this paper, we present a new bidirectional sequential decoding algorithm for decoding terminated trellis codes.

II. THE BIDIRECTIONAL STACK ALGORITHM

The bidirectional stack algorithm (BSA) is based on the notions of: 1) the *reverse trellis code*, obtained from the original one by time reversing; 2) the *tunnel*, the unique sequence $0 \leq T \leq \mathcal{M}$ branches long that connects two states in the trellis; 3) the *tentative decision*, the best so far sequence that connects the known initial and terminal trellis states; 4) a set of *discarding criteria* based on the tentative decision aimed to tell beforehand whether a partly explored path is likely to be a part of the finally decoded sequence or not.

BSA uses two stacks: F (forward) and B (backward, used for the reverse code).

Its steps are:

BSA1. Put the root node into F stack, and the terminal node into B stack, associating them zero metric. Make one of these stacks active (e.g. the F one);

BSA2. Eliminate the node with the largest metric (of length, say, l) from the active stack. Link it via a tunnel to all the eligible paths from the other stack whose lengths are $L - l + \mathcal{M} - T$. Store the best path into the tentative decision register. If there is already a path in the register, keep the better. Establish new discarding criteria and discard the paths from both stacks according to them. If both stacks are emptied in this way, the tentative decision is the decoder's final decision. Otherwise, evaluate the metrics of all the successors of the processed path, and eliminate all of them that do not conform with the discarding criteria;

BSA3. Sort the remaining successors into the active stack according to their metrics. Change the active stack and return to step BSA2.

After each tentative decision, two discarding criteria are established. The first one is based on the nonselection principle [1], that states that from two paths diverging from the same node, the ZJ algorithm keeps the one whose minimum Fano metric until the end node is maximal. The second one checks whether the sum of the accumulated path distance and the minimum such distance among the paths eligible for connecting with it in the future (i.e. of lengths shorter than $L - l + \mathcal{M} - T$) is greater than the accumulated distance of the tentative decision. This is a maximum likelihood criterion. Accumulated distance, d_l , and Fano metric, μ_l , are, after suitable scaling, tied via $\mu_l = l - A \cdot d_l$, $A \in \mathbf{R}^+$.

III. SIMULATION RESULTS

We have simulated the performance of the $\mathcal{M} = 12$ ODP code with $K = 1$, $R = 1/2$, generator polynomials $g_1 = 63374$ and $g_2 = 47244$ (in the usual octal form), $L = 200$ and the code rate equal to the cutoff rate of the binary symmetric channel. Only the frames correctly decodable using VA are used for simulating ZJ and BSA, in order to get a better concordance with the Pareto distribution. Fig. 1 presents obtained computational distributions for the ZJ algorithm, and BSA for $T = 12$ and $T = 0$ (the best distribution obtained). It is notable that BSA Pareto exponent for $T = 0$ is approximately doubled compared to the one pertaining to ZJA.

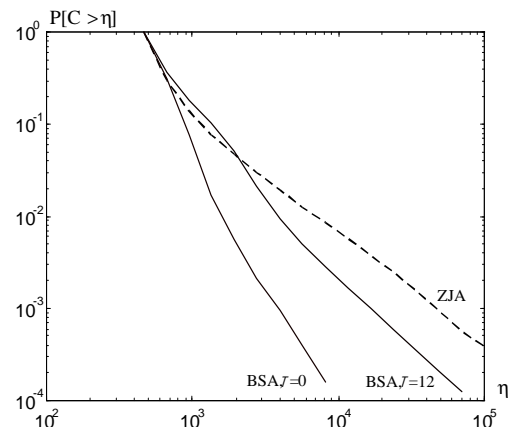


Figure 1. Distribution of the number of nodes generated until the final decision is made

REFERENCES

- [1] J. L. Massey, "Error Bounds for Tree Codes, Trellis Codes and Convolutional Codes with Encoding and Decoding Procedures", CISM Courses and Lectures No. 216, Coding and Complexity, 1975.