



Doctoral Thesis

Multitree search decoding of linear codes

Author(s):

Ostojic, Maja

Publication Date:

2010

Permanent Link:

<https://doi.org/10.3929/ethz-a-006391879> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

DISS. ETH Nr. 19108

Multitree Search Decoding of Linear Codes

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences ETH Zurich

presented by
MAJA OSTOJIC

dipl. El.-Ing. ETH
born August 6, 1976
citizen of Lenzburg AG

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Daniel J. Costello, Jr., co-examiner

2010

Abstract

Tree search algorithms have a long history in computer science. In the coding literature, tree search algorithms have traditionally been used for decoding convolutional codes. Convolutional codes are linear codes with a special structure. Classic tree search decoders (most notably the sequential decoder) search one code tree in which the bits are ordered sequentially.

We propose a multitree search decoder for arbitrary linear codes. We develop several algorithms for constructing code trees from the parity check matrix of a linear code. We propose algorithms that generate code trees, in which the bits appear in random order. We also show how to generate code trees specially designed to decode a given sequence received from a noisy channel. In such code trees, the ordering of the bits depends on the received sequence. Multiple code trees can be generated for each code and sequence. Specialized code trees for low-density parity check codes are also presented.

The different code trees are explored with a new search algorithm. The algorithm is similar to the M-algorithm for convolutional codes; it explores a code tree with limits on the breadth of the explored subtree. An evaluation function is used to decide which node to expand at each depth. We present an evaluation function for general linear codes and an improved evaluation function for low-density parity check codes. Both are optimized for the proposed search algorithm.

The proposed multitree search decoder achieves near optimal performance for short block codes.

For longer block lengths, we propose to use tree search decoding to

improve the standard sum-product decoder for low-density parity check codes. When the sum-product decoder fails to find a codeword, a tree search is used to decode a subset of bits. The channel messages for these bits are then replaced by the decisions found in the tree search in an additional sum-product decoding attempt. This can be repeated multiple times for different subsets of bits. The resulting decoder significantly outperforms the sum-product decoder.

Keywords: Linear codes, low-density parity check codes, tree search, branch and bound, informed search, depth-first search, A^* -search, best-first search, convolutional codes, sequential decoding.

Kurzfassung

Algorithmen zur Baumsuche haben eine lange Geschichte in der Informatik. In der Kodierungsliteratur wurden Baumsuchalgorithmen traditionell zur Dekodierung von Faltungscodes verwendet. Faltungscodes sind lineare Codes mit besonderer Struktur. Diese klassischen Dekodierer (insbesondere der sequentielle Dekodierer) durchsuchen einen Baum, in welchem die Bits sequentiell geordnet sind.

Wir stellen einen Dekodierer für beliebige lineare Codes vor, welcher mehrere Bäume durchsucht. Wir entwickeln verschiedene Algorithmen, welche aus einer Kontrollmatrix Bäume erzeugen, die einen Code mit zufälliger Bitreihenfolge darstellen. Ebenso zeigen wir, wie man Bäume erzeugen kann, die speziell zur Dekodierung einer gegebenen, über einen verrauschten Kanal gesendeten, Sequenz dienen. In diesen Bäumen hängt die Bitreihenfolge von der empfangenen Sequenz ab. Für jeden Code und jede Sequenz kann eine Vielzahl von Bäumen generiert werden. Speziell für Low-Density-Parity-Check-Codes entwickelte Suchbäume werden auch vorgestellt.

Die verschiedenen Bäume werden mit einem neuen Algorithmus durchsucht. Der Algorithmus ist mit dem M-Algorithmus für Faltungscodes verwandt; die Suchbreite ist beschränkt. Eine Schätzfunktion wird verwendet, um in jeder Baumtiefe zu entscheiden, welcher Knoten expandiert werden soll. Wir präsentieren eine Schätzfunktion für allgemeine lineare Codes und eine verbesserte Schätzfunktion für Low-Density-Parity-Check-Codes. Beide Schätzfunktionen sind für den vorgestellten Suchalgorithmus optimiert.

Der vorgestellte Dekodierer erreicht fast optimale Fehlerraten bei Co-

des von kurzer Länge.

Für längere Codes schlagen wir vor, Baumsuche zur Verbesserung des Summe-Produkt-Algorithmus für Low-Density-Parity-Check-Codes zu verwenden. Wenn der Summe-Produkt-Algorithmus kein Codewort findet, wird ein Teil der Bits mittels Baumsuche dekodiert. Die Kanal-messages für diese Bits werden durch die von der Baumsuche gefundenen Entscheidungen ersetzt und ein weiterer Dekodierversuch mit dem Summe-Produkt-Algorithmus wird gestartet. Dies kann mehrfach und mit verschiedenen Teilmengen von Bits durchgeführt werden. Der daraus resultierende Dekodierer übertrifft den Summe-Produkt-Dekodierer deutlich.

Stichworte: Lineare Codes, Low-Density-Parity-Check-Codes, Suchalgorithmen in Bäumen, Branch-and-Bound, Informierte Suche, A^* -Suche, Best-First-Search, Faltungscodes, Sequentieller Dekodierer.