Adaptive Maximum-Likelihood Decoding Algorithms for Linear Block Codes

Hani Mehrpouyan

Abstract

This correspondence represents two new soft decision decoding algorithms that promise to reduce complexity and at the same time achieve the maximum likelihood decoding (MLD) performance. The first method is an Adaptive Two-Stage Maximum Likelihood Decoder [1] that first estimates a minimum sufficient set and performs decoding within the smaller set to reduce complexity and at the same time achieves MLD performance. The second scheme is an Iterative Reliability based decoder [2] that takes advantage of Adaptive Belief Propagation (ABP) [5] to update the reliabilities and then performs Order Statistics Decoding (OSD) or Box and Match Algorithm (BMA) to the new log likelihood ratios (LLRs). The updated reliability values reduce the number of errors in the most reliable positions (MPRs) therefore allowing for a smaller OSD or BMA to be used in the next step of decoding, thus reducing complexity and at the same time achieving close to MLD performance.

Keywords

Maximum Likelihood Decoding (MLD), Reliability Based Decoding (RBD), Two Stage Decoding (TS), Adaptive Belief Propagation (ABP), Order Statistics Decoding(OSD), Box and Match Decoding (BMA).

This work was supported in part by Shahram Yousefi. Hani Mehrpouyan is at the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada K7L 3N6 (email: 5hm@.queensu.ca).

I. Introduction

Hard-decision decoding has been deployed extensively in communication systems due to its simplicity. However the surge in processing power has renewed interest in soft-decision decoding algorithms that provide considerable performance enhancement over their counterparts.

Soft-decision Maximum Likelihood Decoding (MLD) algorithms provide performance gains at the cost of significant increase in complexity. The adaptive decoding algorithms provided in [1] and [2] claim to reduce complexity significantly and at the same time achieve the optimal performance of MLD. Reliability based algorithms [2] order the received symbols in blocks based on the assigned reliability values, dividing the code into reliable and unreliable sections, and therefore they reduce the number of computations required for MLD. The alternative approach proposed in [1] applies a two-stage (TS) coding structure, that first estimates a minimum sufficient set (MSS) of candidate codewords containing the optimal codeword and secondly performs optimal decoding within this smaller set. However it is important to note that both schemes suffer from exponential complexity under the worst case scenario.

II. Back Ground Information

A. OSD [9] reduces cost of MLD decoding at the expense of performance. The received sequence $\mathbf{r}$ is reordered based on the reliability values from the most reliable to the least reliable one. Because the first k symbols of $\mathbf{y}$ are the k most reliable independent symbols their hard decision should contain very few errors. Based on this concept, the algorithm generates a sequence of candidate codewords for testing by processing the k most reliable independent symbols of $\mathbf{y}$. The candidate codeword $\mathbf{v}^*$ with the least correlation discrepancy with $\mathbf{y}$ is the decoded codeword.

For $0 \leq i \leq k$, the OSD algorithm of order-i makes all possible changes to the l of the k most reliable binary digits for $0 \leq l \leq i$. Thus an OSD of order-i consists of $(i+1)$ processing phases and requires a total of

$$1 + \binom{k}{1} + ... + \binom{k}{i} \tag{1}$$

candidate codewords to make a decoding decision. The OSD-k is MLD, and requires 2^k operations for decoding. However $i = \lfloor d_{min}/4 \rfloor$ is practically sufficient to achieve the same error performance as MLD for bit error rates (BER) larger than 10^-6 [6].

B. In 1972 Chase devised three algorithms for decoding binary block codes [12]. Chase II and Chase III are the two schemes of interest in this paper and will be discussed very briefly here. In Chase III an error correcting algebraic decoder is used to generate a list of $\lfloor d_{min}/2 + 1 \rfloor$ candidate codewords from the least reliable positions (LRP). MLD decoding is performed on candidate codewords to determine the received codeword with the least discrepancy. On the other hand Chase II creates a set of candidate codewords by modifying the $\lfloor d_{min}/2 \rfloor$ LRPs. Therefore resulting in a larger set of codewords and higher complexity. In fact the complexity of Chase II grows exponentially with d_{min} [6]. Clearly Chase II's performance is closer to MLD compared to Chase III.

C. Adaptive Belief Propagation (ABP)

Graphs of many traditional good codes such as the Reed-Solomon codes and BCH codes contains many short cycles thus making the above codes unsuitable for Belief Propagation BP. However in [10] a new adaptive scheme has been proposed where the graph of the code is iteratively updated to make it suitable for BP. For example upon receiving $C = \{c_1, c_2, ..., c_n\}$ the log likelihood ratios, (LLRs) $\mathbf{L}= \{L_1, L_2, ..., L_n\}$ are put into ascending order with $\{i_1, i_2, ..., i_n\}$ representing the bits corresponding to the sorted LLRs. Therefore, the i_1^{th} bit is the least reliable (LR) and i_N^{th} bit is the most reliable. We first reduce the i_1^{th} column of the parity check matrix of the code, H to the form $[100...0]^T$ by performing row operations. Then we proceed further and attempt to reduce the ith_2 column to the form $[0100...0]^T$. If this is not possible, we continue down the list and try to reduce the i_3^{th} column to the form $[010...0]^T$. Thus we proceed down the list $i_1, i_2, ..., i_N$ and reduce columns to be of weight one. Since H is of full-rank, it is possible to reduce exactly $(N - K)$ columns to be of weight 1, though these $(N - K)$ may not be the LR bits. This construction increases the likelihood that bits in error are moved to leaves on the graph. In other words, it decreases the probability that a bit in error participates in any loops. Thus error propagation is limited. Further the bit on the leaf receives extrinsic information from one check node only. Finally when the algorithm has stopped we will have a matrix of the form:

$$
\begin{array}{c}
\begin{bmatrix}
\cdots & 1 & \cdots & 0 & 0 & \cdots & \cdots & 0 & \cdots \\
\cdots & 0 & \cdots & 1 & 0 & \cdots & \cdots & 0 & \cdots \\
\cdots & 0 & \cdots & 0 & 1 & \cdots & \cdots & 0 & \cdots \\
\cdots & 0 & \cdots & 0 & 0 & \cdots & \cdots & 1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix} \\
\begin{array}{ccccccccc}
 & i_1 & & i_2 & i_3 & & i_4 & i_5 &
\end{array}
\end{array}
$$

Fig. 1. H after performing BP [11]

[10] & [11] provide a fairly detailed explanation of the ABP scheme.

D. Iterative Box Match Decoding (BMA)

BMA [13] is quite similar to the OSD algorithm described above. BMA of order-i has a performance similar to OSD of order-i. In addition to considering the codewords with i error patterns on the most reliable basis (MRB), BMA also considers error patterns of hamming weight $2i$ in the set of MRPs S. The algorithm is referred to as BMA$(i, S-k)$ and the $(S-k)$ values outside the MRB as the control band (CB). Therefore the average list size of BMA is:

$$\binom{k}{i} + 2^{-(S-k)}\binom{S}{2i} \tag{2}$$

A more detailed explanation of the BMA algorithm including performance analysis of BMA can be found in [13] & [14].

III. **Two Stage Maximum-Likelihood Decoding Proposed in [1]**

A. System structure

The TS decoder first estimates the MSS and then performs ordered algebraic decoding (OAD) within the MSS. The overall structure takes $\mathbf{r}$ the received vector, log likelihood vector L calculated based on 3, and the hard limited vector as inputs. Figure 2b illustrates the overall structure of the proposed block decoder for an [N, K] code.

$$\alpha_{\mathbf{i}} = \log\frac{Pr(r_i|c_i=1)}{Pr(r_i|c_i=0)} = \frac{2}{\sigma^2}.r_i, \qquad i = 1, ..., N \tag{3}$$

The LLR vector L is ordered from the least reliable to the most reliable values with vector α recording the position that each reliability value corresponds to in $\mathbf{r}$.

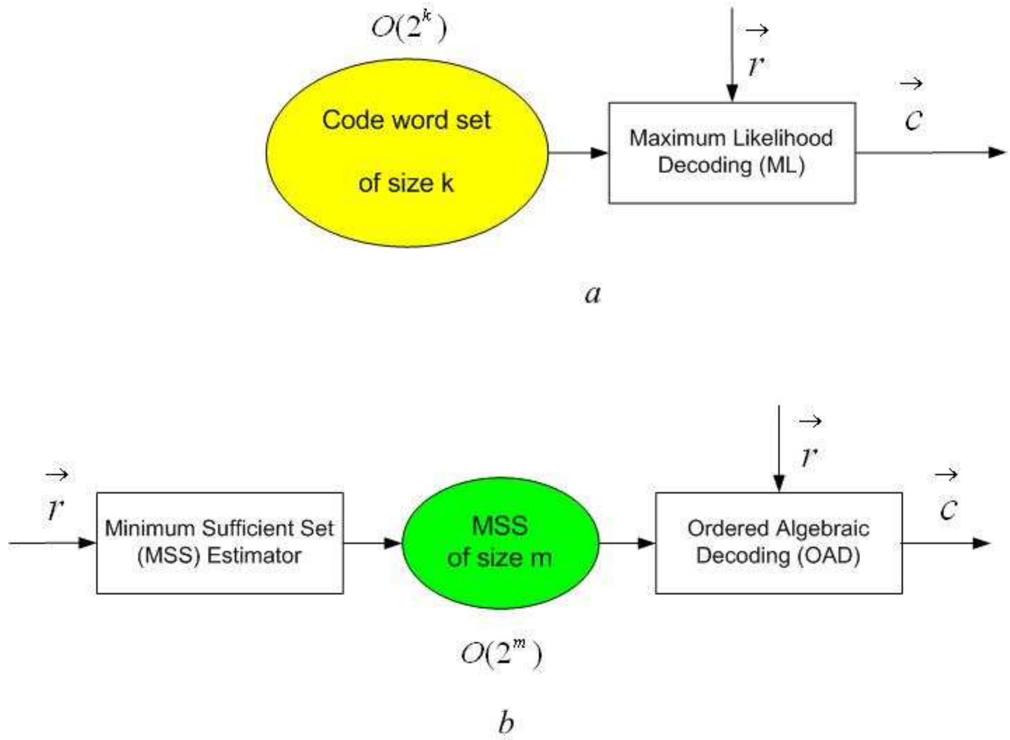


Fig. 2. Illustrating the decoding algorithm proposed in [1]

B. Adaptive TS Decoder

Figure 2 illustrates the advantage of the TS algorithm to conventional MLD. The ML decoder searches through all the available codewords to find the optimal one ($\leq O(2^K)$), however by estimating the MSS the TS method needs to only search through the smaller set, thus drastically reducing complexity. Further complexity reduction is achieved using Optimum Test Criterias (OTCs) [8] to terminate the search whenever the optimal codeword is found. Therefore the resulting complexity of the TS algorithm is less than $\leq O(2^m)$, where m is the size of the MSS.

C. Theoretical MSS and MSS Estimation

The MSS is an important part of complexity reduction proposed both in [1] & [4]. As mentioned in the previous section decoding complexity is upper bounded by the size of MSS, thus the theoretical average of m, $\bar{M}$, directly affects the decoding complexity and is of importance. Let E_m be the set of all vectors generated by applying all possible errors to the first m most unreliable positions of $\mathbf{y}$ and let S_m be the set of decoded codewords corresponding to E_m. Then for some m the set S_m contains the optimal codeword $\mathbf{c}_{opt}$. The MSS is the smallest such set containing $\mathbf{c}_{opt}$. According to proposition 1 of [1], (refer to the paper for the proof) MSS is the set S_m in which one error happens at position α_m and there are exactly τ errors outside the first M least reliable positions, where τ is the error correcting capability of code C. Treating M as a random variable subject to the noise effect, we denote $\Phi(M)$ as the probability of S_M being the MSS for a given M, and $\Phi(M, e)$ as the joint probability associated with the events when S_M is the MSS and there are exactly a total number of e errors. It follows immediately that $\bar{M} = \sum_{M=1}^{N} M.\Phi(M)$ and $\Phi(M) = \sum_{e=0}^{N} \Phi(M, e)$. When $e \leq \tau$, the hard-decision vector $\mathbf{y}$ can be directly mapped to the optimal codeword, resulting in $M = 0$. For $M \neq 0$, $\Phi(M)$ reduces to τ $\Phi(M) = \sum_{e=\tau+1}^{N} \Phi(M, e)$.

Lemma 2 of [8] is used for estimating the MSS. The MSS is estimated based on the idea that some certain codewords within the set of all possible codewords do not meet certain optimality conditions, therefore they cannot be the optimal codeword. Thus the MSS can be estimated by identifying these codewords and removing them from the set of candidate codewords for decoding.

The input sequence to the estimator consists of the hard decision sequence $\mathbf{y}$ and an error sequence $\mathbf{e}= (e_1, e_2, ...e_n)$. The set of error sequence that is enough to perform MLD is of interest here and will be discussed. Let C_x, be the set of codewords which are more likely than $x = (x_1, x_2, ...x_n) \in C$, where x is generated by the encoder. Then C_x, is enough to perform MLD, however, we cannot generate C_x, exactly without referring to all codewords. Thus we define, instead, the set of codewords L_i, as follows: We select all sequences as the estimated error sequences es which have any combination of ls, which are located in the i positions with the lowest value of $|\alpha_i|$. Then , define a set L_i, to be the set of codewords that are outputs of the algebraic decoder when $\mathbf{y} + \mathbf{e}$ are inputs. To generate L_i the error sequence is estimated and then provided as an input to the algebraic decoder, therefore:

$$L_0 \subseteq L_1 ... \subseteq L_N \tag{4}$$

where L_N represents the set of all possible codewords. Then lemma 2 of [8] states:

$$l(x,y) < \sum_{i=1}^{d-\lfloor \frac{m_0+m}{2} \rfloor - \tau - 1} |\alpha_{s_x^{(i)}}| + \sum_{i=1}^{\tau+1} |\alpha_{u^{j+i}}, \tag{5}$$

then $C_x \subseteq L_j$. S_x is the set of x_is that have the same values as $y_i \in \mathbf{y}$ (if $x_i = y_i$ then $x_i \in S_x$). This powerful result ensures that the most optimum codeword belongs to L_j as long as condition 5 is satisfied and thus the search for the optimum codeword becomes $O(2^j = m)$ complex where $j \ll N$ at high SNRs.

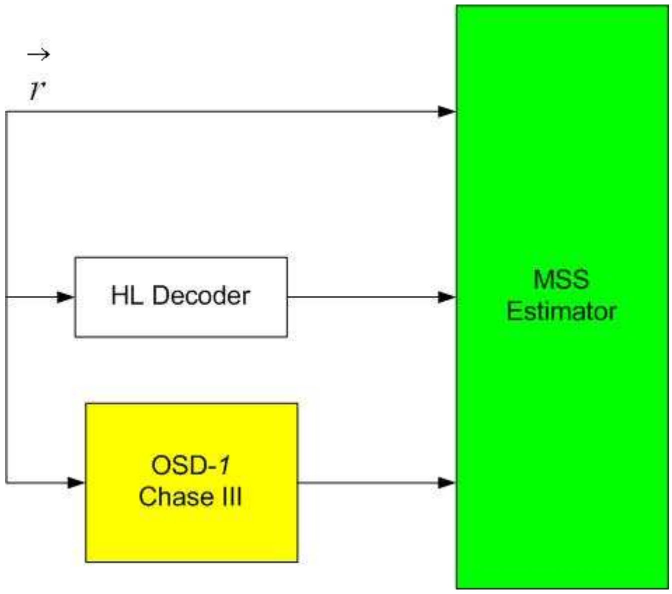


Fig. 3. The MSS estimator and the inputs based on [8] & [1]

Figure 3 represents the inputs to the MSS estimator. As illustrated the estimator requires a suboptimal codeword to estimate the MSS. [?] uses a combination of OSD and Chase II (This scheme is called complementary decoding since Chase II takes advantage of the LRPs and OSD uses the MRPs to come up with the most optimal codeword. The resultant codewords from the two algorithms are then compared and the codeword with the largest correlation is chosen as x_{est}). However as described before the complexity of

OSD and Chase II increases exponentially with the order of the OSD and the size of d_{min} respectively, therefore [1] proposes the use of OSD_1 and Chase III to come up with x_{est}.

It is important to note that the closer x_{est} to the optimal codeword the closer the MSS_{est} to MSS. Therefore the method proposed in [4] will result in a better estimation of the MSS compared to [1], on the other hand at high SNRs the difference is insignificant and therefore the complexity gain is more advantageous. Figure 4 illustrates the size of the estimated MSS compared to the theoretical results. It is clear that at high SNRs all algorithm will converge in terms of performance (the method proposed in [4] is not included in the graph but is expected to have performance similar to TS-ML scheme).

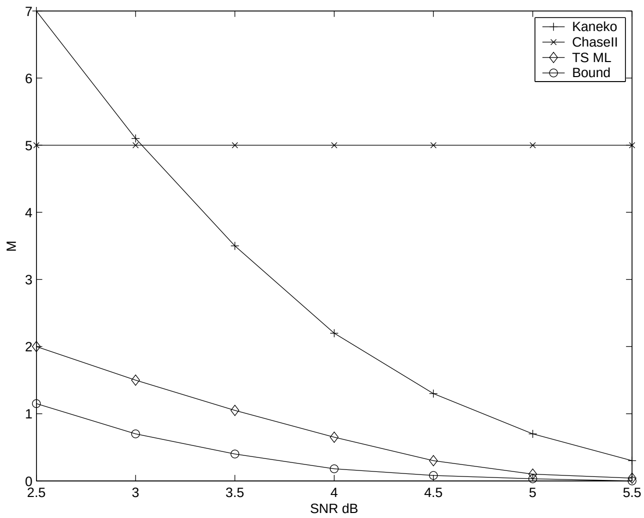


Fig. 4. complexity of ($\bar{M}$) versus SNR [1]

D. Ordered Algebraic Decoding

The second stage of decoding attempts to find the codeword with highest correlation within the smaller MSS, passed on from the previous stage. [1] uses and order_i algebraic decoder (OAD_i). The MSS is created by applying the error patterns to the M most unreliable positions of the the hard-decision vector $\mathbf{y}$. However the complexity could be further reduced by applying the errors patterns to only the first i unreliable positions, therefore resulting in the OAD_i decoder. Then among the new set the codeword that has the minimum Euclidean distance to $\mathbf{r}$ is chosen as the output.

Figure 5 illustrates the structure of the order_i algebraic decoder and its relation with

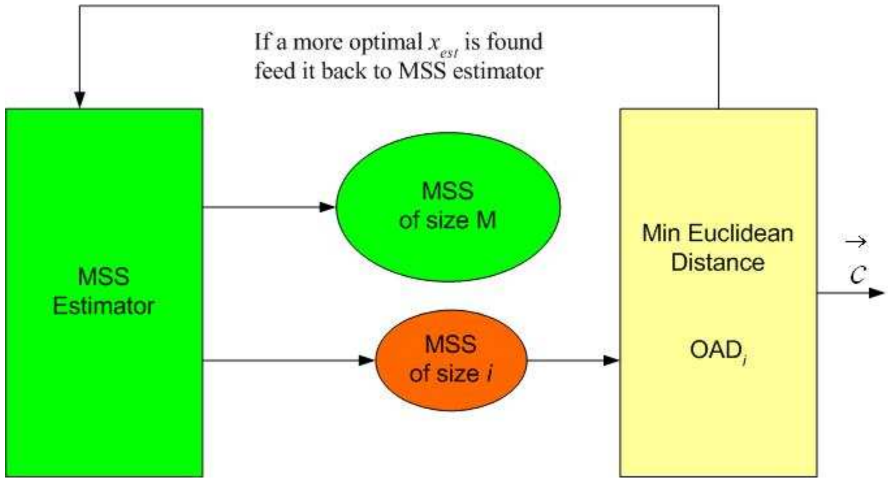
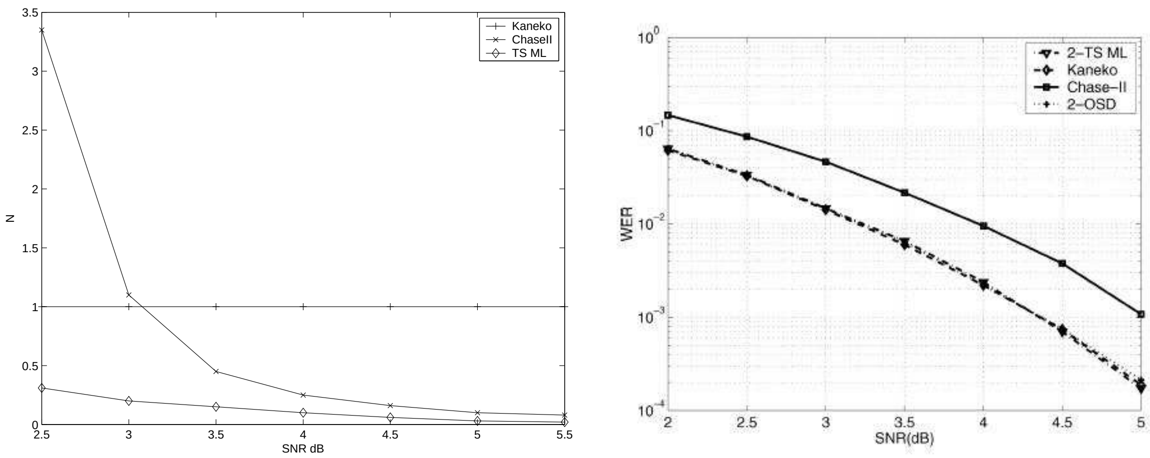


Fig. 5. OAD$_i$ algorithm used in finding the optimum codeword within the MSS

the MSS estimator.



(a) **(b)**

Fig. 6. Performance comparison for BCH(31,16) [1]

The OTC for the optimal codeword [8] is built into the decoder to ensure termination when the optimal codeword is found and the MSS is also updated whenever a more optimal codeword has been found. Figure6 (a) compares the overall complexity of the TS with Chase II and Kaneko [8] and figure6 (b) illustrates the performance analysis in terms of word error rate between the different codes.

IV. Iterative Reliability-Based Decoding Proposed in [2]

The iterative scheme proposed consists of two different decoding approaches. The first method combines ABP and OSD to determine the optimal codeword, on the other hand the second algorithm takes advantage of ABP, BMA, and iterative information set reduction (IISR) [13] for determining the optimum codeword. Figure 7 Illustrates the approach of both schemes.

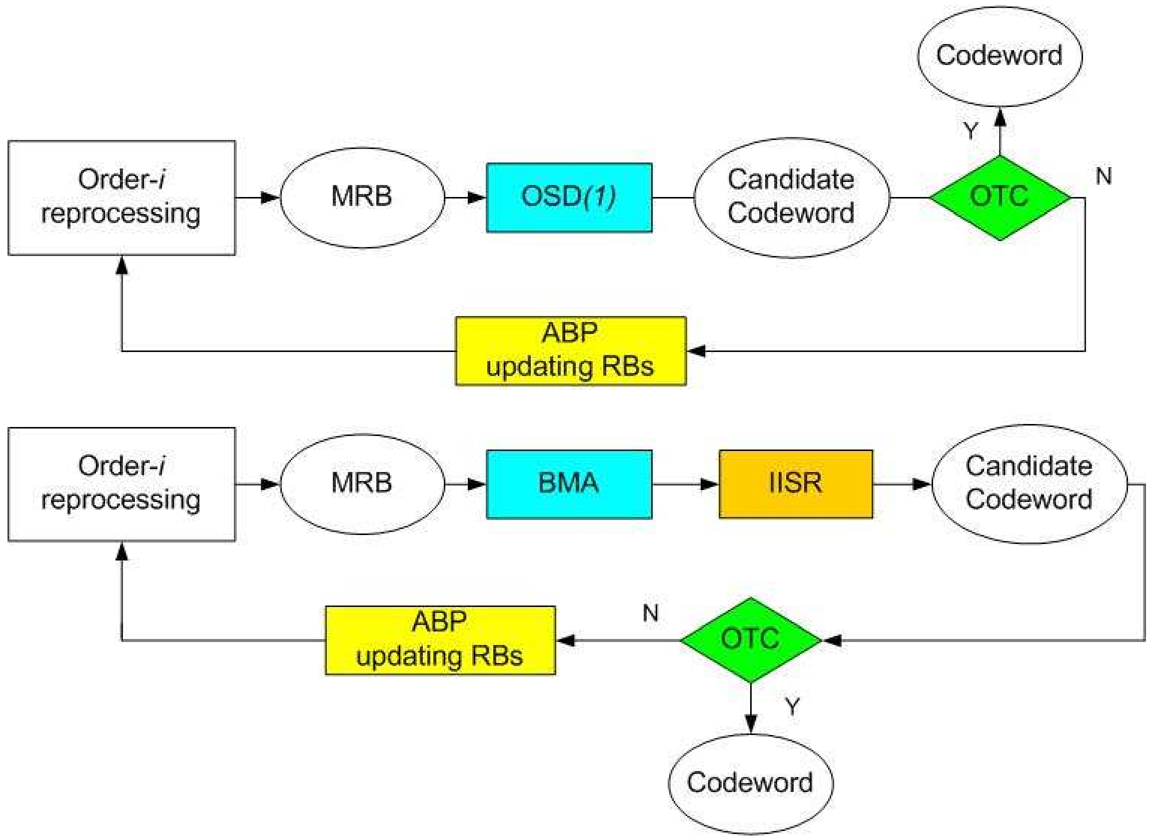


Fig. 7. OAD_i algorithm used in finding the optimum codeword within the MSS

A. ABP combined with OSD(1)

At every iteration of the algorithm the following steps are executed:

1. Using the log likelihood ratios (LLRs) a new MRB is generated. OSD(1) is performed to come up with the optimum codeword within the MRB and OTCs are used to determine if the optimal codeword has been found.

2. If the previous step does not result in the optimal codeword, then ABP is performed to update the LLRs and step 1 is repeated.

B. ABP combined with BMA(1) and IISR

The proceeding algorithm is repeated in this section with BMA(1) and IISR replacing OSD(1). IISR is very much similar to the extension provided by BMA to OSD. IISR of BMA consists at the end of the conventional BMA of selecting the $(f - S)$ MRPs outside the the MRB and CB and assuming they are error free. a new MRB is constructed with these $(f - S)$ MRPs automatically included in it. This new BMA scheme is called IT-BMA$(i, S - k, f - S)$.

It is expected that second approach using the ABP-IT-BMA algorithm have a higher complexity with respect to the ABP-OSD algorithm, on other hand performance gain is expected using ABP-IT-BMA compared 5to ABP-OSD due to the fact that more codewords are taken into considerations during the decoding process. The simulation result provided by [2] support this intuition. The results are presented in figure 8 & table I.

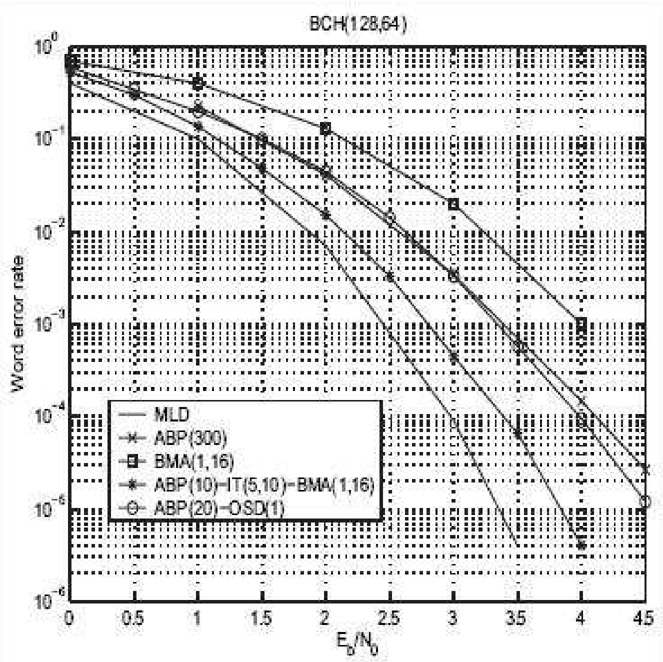


Fig. 8. WER of ABP-OSD and ABP-IT-BMA for a (128,64) eBCH code [2]

V. **Comparing the TS model with the Iterative ABP scheme**

Comparing the two models proposed in section III and IV of this report is a difficult task. Both methods attempt to reduce the complexity of MLD but the TS model reduces

TABLE I

NUMBER OF CANDIDATES PROCESSED FOR THE BCH(128, 64, 22) CODE (100 WORD ERRORS RECORDED) [2]

SNR	3dB		
Algorithm	ABP(20)-OSD(1)	BMA(1)	ABP(10)-IT(5,10)-BMA(1,16)
Max. for Simulations	1344	399	4314
Average	606	171	1917

complexity when achieving ML performance on the other hand the IT-ABP model only achieves near MLD performance. It is also important to note that the overall complexity of the IT-ABP scheme is user defined. By reducing the number of iterations and also using a lower order OSD or BMA scheme the overall complexity of the decoder can be considerably reduced. However the TS algorithm's complexity is dictated by the size of the MSS. The TS model also achieves achieve MLD performance compared to the IT-ABP model, therefore it is expected to have a higher worst case complexity. It is important to note that the results provided in [1] & [2] may not be compared in terms of complexity since the two papers use two different version of the BCH code to simulate the algorithms.

VI. **Summary**

In this paper I provided an overview of two soft decision decoding schemes that use suboptimal decoding tools to achieve MLD or near MLD performance at reduced complexity. The worst case complexity is still exponential however using the OTCs and MSS the overall complexity is reduced quite substantially in a majority of scenarios.

References

[1] Xianren Wu, Hamid R. Sadjadpour, and Zhi Tian, "A New Adaptive Two-Stage Maximum-Likelihood Decoding Algorithm for Linear Block Codes," *IEEE Transactions on Communications Theory*, VOL. 53, NO. 6, June 2005.

[2] Aditi Kothiyal, Oscar Y. Takeshita, Wenyi Jin, and Marc Fossorier, "Iterative Reliability Based Decoding of Linear Block Codes with Adaptive Belief Propagation," *BIEEE Communication Letters*, VOL. 9, NO. 12, December 2005.

[3] Marc P.C. Fossorier and Shu Lin, "Error Performance Analysis for Reliability-Based Decoding Algorithms," *IEEE Transactions on Information Theory*, VOL. 43, NO. 5, September 1997.

[4] Marc P. C. Fossorier and Shu Lin, "Complementary Reliability-Based Decodings of Binary Block Codes," *IEEE Transactions on Information Theory*, VOL. 43, NO. 5, September 1997.

[5] Jing Jiang and Krishna R. Narayanan, "Iterative Soft Decision Decoding of Reed Solomon Codes Based on Adaptive Parity Check Matrices," *ISIT 2004, Chicago, USA,*, June 27 July 2, 2004.

[6] Shu Lin and Daniel J. Costello, Jr., "Error Control Coding ," *Second Edition*, 2004.

[7] Frank R. Kschischang, Brendan J. frey, and Hans-Andrea Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, July 1998.

[8] Toshimitsu Kaneko, Toshihisa Nishijima, Hiroshige Inazumi, and Shigeichi Hirasawa, "An Efficient Maximum-Likelihood-Decoding Algorithm for Linear Block Codes with Algebraic Decoder," *IEEE Transactions on Information Theory*, VOL. 40, NO. 2, MARCH 1994.

[9] Marc P. C. Fossorier and Shu Lin, "Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics," *IEEE Transactions on Information Theory*, VOL. 41, NO. 5, SEPTEMBER 1995.

[10] Jing Jiang and Krishna R. Narayanan, "Iterative Soft Decision Decoding of Reed Solomon Codes Based on Adaptive Parity Check Matrices," *ISIT 2004*, Chicago, USA, June 27 July 2, 2004.

[11] Aditi Kothiyal and Oscar Y. Takeshita, "A Comparison of Adaptive Belief Propagation and the Best Graph Algorithm for the Decoding of Linear Block Codes," 4-9 Sept. 2005 Page(s):724 - 728 .

[12] DAVID CHASE, "A Class of Algorithms for Decoding Block Codes With Channel Measurement Information," *IEEE Transactions on Information Theory*, VOL. IT-18, NO. 1, JANUARY 1972.

[13] M. Fossorier, "Reliability-based soft-decision decoding with iterative information set reduction," *IEEE Transactions on Information Theory*, VOL. 48, PP. 3101- 3106, Dec. 2002.

[14] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Transactions on Information Theory*, VOL. 50, PP. 796- 810, May 2004.