

PROBABILISTIC LIST SPHERE DECODING FOR LDPC-CODED MIMO-OFDM SYSTEMS

Sungchung Park, Kwyro Lee
Korea Advanced Institute of
Science and Technology

Email: {chester,krlee}@eeinfo.kaist.ac.kr

Yuping Zhang, Keshab K. Parhi
University of Minnesota
Email: {zhan0480,parhi}@umn.edu

Jin Lee, Sin-Chong Park
Information and Communications
University

Email: {mygenie,separk}@icu.ac.kr

ABSTRACT

A probabilistic list sphere decoding (LSD) is proposed for LDPC-coded MIMO-OFDM systems. By confining the search to channel-adaptively chosen promising candidate groups, our proposed probabilistic LSD evaluates the more promising candidates earlier in the search, while retaining efficient implementation of the depth-first LSD. Simulation results show that, the proposed LSD significantly improves the error performance for constrained throughput, or increases the throughput for a fixed bit error rate. The proposed LSD can be implemented by simply adding special hardware blocks that perform preprocessing and control the tree traversal. The K-best search algorithm enables efficient implementation of the preprocessing.

1. INTRODUCTION

Nowadays, a lot of attention has been paid to low-density parity-check (LDPC) coded multi-input multi-output (MIMO) orthogonal frequency division multiplexing (OFDM) systems. For example, the IEEE Task-Group N (IEEE802.11n) proposed the use of LDPC-coded MIMO-OFDM transmission for the next-generation wireless local area network systems. In order to improve the error performance, list sphere decoding (LSD), which finds the N_{CD} closest candidates (usually, $N_{CD} > 1$), is required to provide reliable *a posteriori probabilities* for soft input LDPC decoding [1]. Among a variety of modifications of LSD, the depth-first LSD (DF-LSD) is particularly attractive in terms of both error performance and hardware complexity [2], [3]. Note that the DF-LSD inherently assumes unconstrained throughput, and, with tightly constrained throughput, the N_{CD} closest candidates cannot always be found, since the search may be terminated before evaluating all potential candidates [2], [3]. For a throughput of 20 Mv/s (mega-vectors per second) (e.g., IEEE802.11n), even the state-of-the-art LSD architectures ($N_{CD} = 1$) in [2] experience significant degradation in error performance. In [3], the authors proposed the use of parallel processing of multiple LSD blocks, which seems to be impractical because of increasing hardware complexity.

In this paper, the probabilistic sphere decoding (SD) ($N_{CD} = 1$) proposed in [4] is extended to generate soft information in LDPC-coded MIMO-OFDM systems ($N_{CD} >$

1). Our proposed LSD, the so-called probabilistic LSD (P-LSD), evaluates the more promising candidates earlier in the search, while retaining efficient implementation of the DF-LSD. Simulation results show that the proposed P-LSD significantly improves the error performance for constrained throughput, or increases the throughput for a fixed bit error rate (BER).

2. CALCULATION OF A POSTERIORI PROBABILITY

For N_T transmit and N_R receive antennas and M -ary quadrature amplitude modulation (QAM), the *a posteriori* L -value of the LDPC-coded bit x_i , $i = 1, \dots, N_T \log_2 M$, is given as [1],

$$L_i^{post} = \max_{x_i=1} \left(-\frac{1}{N_0} \|\underline{y} - H\underline{s}\|^2 + \sum_{j=1}^{N_T \log_2 M} L_j^{priori} \right) - \max_{x_i=0} \left(-\frac{1}{N_0} \|\underline{y} - H\underline{s}\|^2 + \sum_{j=1}^{N_T \log_2 M} L_j^{priori} \right) \quad (1)$$

where N_0 is the single-sided noise power spectral density, \underline{y} is the $N_R \times 1$ received signal vector, \underline{s} is the $N_T \times 1$ vector symbol whose element is drawn from the M -QAM constellation, H is the $N_R \times N_T$ channel coefficient matrix, and L_i^{priori} is the *a priori* L -value of x_i . The use of LSD maximizes the two terms in (1), approximately yet efficiently. To be specific, the maximization is considered over the N_{CD} candidates that minimize $\|\underline{y} - H\underline{s}\|^2$ [1].

3. OVERVIEW OF LIST SPHERE DECODING

3.1. Conventional list sphere decoding

The sphere constraint with radius r is given as

$$d(\underline{s}) < r^2 \quad \text{with } d(\underline{s}) = \|\underline{\tilde{y}} - R\underline{s}\|^2, \quad \underline{\tilde{y}} = Q^H \underline{y} \quad (2)$$

where $d(\underline{s})$ is the distance, and the $N_R \times N_T$ orthonormal matrix Q and the $N_T \times N_T$ upper triangular matrix R are defined according to a QR decomposition of H . Due to the upper triangularity of R , the calculation of $d(\underline{s})$ can be performed recursively in the search tree: by describing each node in the i -th level by the *partial vector symbol*

$$\underline{s}_{(i)} = [s_i \ s_{i+1} \ \dots \ s_{N_T}]^T,$$

the corresponding partial distance $T_i(\underline{s}_{(i)})$ is recursively calculated as

$$T_i(\underline{s}_{(i)}) = T_{i+1}(\underline{s}_{(i+1)}) + \delta_i(\underline{s}_{(i)}) \quad \text{with} \quad \delta_i(\underline{s}_{(i)}) = \left| \tilde{y}_i - \sum_{j=i}^{N_T} r_{i,j} s_j \right|^2 \quad (3)$$

where $T_{N_T+1}(\underline{s}_{(N_T+1)})$ is set to zero and $d(\underline{s})$ is simply given by $T_1(\underline{s}_{(1)})$ [2].

In Fig. 1, three transmit and receive antennas ($N_T = N_R = 3$) and QPSK with Gray encoding ($M = 4$) are assumed. A shaded circle represents the child node whose $T_i(\underline{s}_{(i)})$ is minimized over the M possible child nodes for a particular parent node. For example, the vector symbol consisting of all shaded circles (corresponding to $[0 \ 1 \ 1]^T$) is nothing but the nulling and canceling (NC) solution. Note that only child nodes minimizing $T_i(\underline{s}_{(i)})$ are displayed at level $i = 1$, for simplicity. The numbers in the parentheses represent the instant when the node is evaluated (in clock cycles), assuming the one-node-per-cycle architecture [2]. For clearer illustration, tree pruning is not displayed in Fig. 1. The diagonal elements are assumed to be given as $r_{1,1}^2 = 3.5$, $r_{2,2}^2 = 2.0$ and $r_{3,3}^2 = 0.5$.

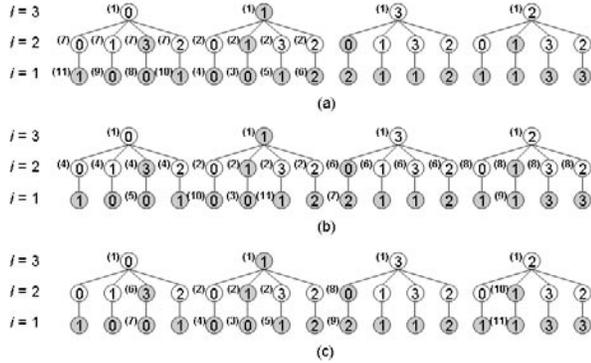


Fig. 1. Comparison of tree traversals
(a) DF-LSD [2], [3]
(b) probabilistic search algorithm [5]
(c) proposed P-LSD

Throughout this paper, the DF-LSD is assumed to adopt the radius reduction with infinite initial radius and the SE ordering, as in [2]. Interestingly, the radius reduction of the standard sphere decoding (SD) in [4] is modified as follows.

- Step 1: Starting with infinite radius, find candidates and store them to the list, until N_{CD} candidates are found.
- Step 2: Reduce the radius to the *maximum* distance in the list.
- Step 3: Continue to find candidates with the radius. Whenever a candidate is found, update the list and go back to Step 3.

Note that, since the radius is chosen as the maximum distance in the list (not the minimum one, as in the standard

SD), the radius reduction of LSD tends to be slower, as N_{CD} increases. Fig. 2 compares the averages and probability density functions (at 15 dB) of the number of visited nodes N , showing that the impact of constrained N on the error performance is larger in LSD than in SD. Thus, the performance improvement by probabilistic search algorithm is expected to be more significant when $N_{CD} > 1$ than when $N_{CD} = 1$ (as in [4]).

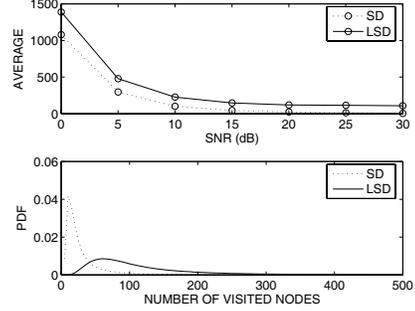


Fig. 2. The number of visited nodes ($N_{CD} = 16$):

As shown in Fig. 1 (a), in the DF-LSD, the potential candidates are evaluated in a predetermined order that exploits reuse of intermediate calculations with a memory size as small as $(M - 1)(N_T - 1)$. In terms of the probability of correct decision, however, the tree traversal in Fig. 1 (a) is not the best solution [5]. Since the number of visited parent nodes is constrained to guarantee fixed throughput [2], [3], it is desirable to evaluate the potential candidates in descending order of probability, in other words, evaluate the more promising candidates earlier (before termination). The probability decreases with

$$Q(\underline{s}) = \sum_{j=1}^{N_T} r_{j,j}^2 u_j^2 \quad (4)$$

where u_j is the distance of s_j from the constellation point minimizing $T_i(\underline{s}_{(i)})$ [5]. Since u_j is either 1.414 or 2 in the QPSK constellation, the probabilistic search algorithm in Fig. 1 (b) evaluates the potential candidates in the order of $[0 \ 1 \ 1]^T$ (i.e., the NC solution), $[0 \ 3 \ 0]^T$, $[2 \ 0 \ 3]^T$, $[1 \ 1 \ 2]^T$, $[0 \ 0 \ 1]^T$, $[1 \ 3 \ 1]^T$ and so on, i.e., in ascending order of $Q(\underline{s})$. Obviously, the probabilistic search algorithm in [5] is expected to improve the error performance of the constrained DF-LSD (i.e., DF-LSD with constrained throughput) by finding the more promising candidates earlier. However, it is not applicable to the DF-LSD, since the required memory size becomes prohibitively large (larger than $(M - 1)(N_T - 1)$) and, moreover, unpredictable, as shown in Fig. 2 (b). Furthermore, the memories are no more accessible in a simple *first-in last-out* manner, requiring random access memories instead of stack memories.

3.2. Proposed probabilistic list sphere decoding

In M -ary QAM, all M constellation points can be divided into a few *constellation groups*, based on the distance from

a reference point. If the reference point itself is considered as a group ($G0$), there are 3 groups ($G0 \sim G2$) in QPSK, 10 groups ($G0 \sim G9$) in 16-QAM, 33 groups ($G0 \sim G32$) in 64-QAM, as shown in Fig. 3. Note that the squared distances of constellation groups, u_j^2 in (4) are given as $G0$: 0, $G1$: 2, $G2$: 4, $G3$: 8, $G4$: 10 and $G6$: 18. Subsequently, M^{N_T} candidates can be divided into a few *candidate groups*, based on the constellation groups of each level, with the reference point of the i -th level being the constellation point minimizing $T_i(\underline{s}_{(i)})$. From [5], it is easily understood that all candidates belonging to a particular candidate group have the same probability of correct decision. Note that, in Fig. 1 (b), the probabilistic search algorithm evaluates the candidate groups in the order of $G0$ - $G0$ - $G0$ ($[0 \ 1 \ 1]^T$), $G0$ - $G0$ - $G1$ ($[0 \ 3 \ 0]^T$, $[2 \ 0 \ 3]^T$), $G0$ - $G0$ - $G2$ ($[1 \ 1 \ 2]^T$) and so on.

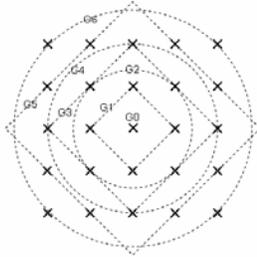


Fig. 3. Constellation groups of M -QAM

Our proposed P-LSD can be seen as a special case of the DF-LSD whose search is confined to a few promising candidate groups. By confining the search to the candidate groups with high probability, it ignores the less promising candidates and evaluates the more promising ones earlier. Therefore, it improves the error performance of the DF-LSD, while retaining efficient implementation. Fig. 1 (c) shows the proposed P-LSD with candidate groups of $G0$ - $G0$ - $G0$, $G0$ - $G0$ - $G1$, $G0$ - $G0$ - $G2$ and $G0$ - $G1$ - $G0$. Note that the required memory size is usually even smaller than $(M - 1)(N_T - 1)$ and the most promising six candidates are evaluated within 11 cycles, just as in Fig. 1 (b), even though the candidate groups are evaluated in a slightly different order (i.e., $G0$ - $G0$ - $G0$ ($[0 \ 1 \ 1]^T$), $G0$ - $G1$ - $G0$ ($[0 \ 0 \ 1]^T$, $[1 \ 3 \ 1]^T$), $G0$ - $G0$ - $G1$ ($[0 \ 3 \ 0]^T$, $[2 \ 0 \ 3]^T$), $G0$ - $G0$ - $G2$ ($[1 \ 1 \ 2]^T$)).

4. HARDWARE ARCHITECTURE

Our proposed P-LSD can be implemented by simply adding special hardware blocks to the DF-LSD blocks shown in [2] or [3]. These additional hardware blocks play two roles: selection of candidate groups as a preprocessing step and control of tree traversal that confines the search to the selected candidate groups.

In order to select K promising candidate groups, $Q(\underline{s})$ needs to be evaluated, based on the channel realization (i.e., $r_{i,j}$). Note that constellation groups at $i = 1$ does not need to be concerned in the selection, since our proposed P-LSD evaluates all possible child nodes within one cycle (referred

to as the *one-node-per-cycle architecture* in [2]). Consequently, denoting the number of constellation groups under consideration by L , the K candidate groups minimizing $Q(\underline{s})$ should be found among all possible $L^{(N_T-1)}$ candidate groups. Thus the selection of the K candidate groups can be seen as a tree search with $(N_T - 1)$ levels and L nodes. Obviously, L should be chosen as the minimum of K and M , since any candidate group containing constellation groups larger than GK (or GM) never minimizes $Q(\underline{s})$. Fig. 4 shows the tree (before tree pruning), assuming four antennas ($N_T = N_R = 4$), 64-QAM ($M = 64$) and 10 candidate groups ($K = 10$) (thus $L = 10$). Here the well-known K -best algorithm is applied to the selection of K candidate groups. Interestingly, since $Q(\underline{s})$ is not jointly evaluated over antennas as shown in (4), the K -best algorithm is quite simple, and, moreover, always guarantees the optimality of the K candidate groups, as opposed to the general K -best algorithms.

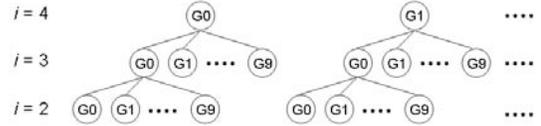


Fig. 4. Tree for selecting promising candidate groups ($L = 10$)

The preprocessing hardware blocks for selecting candidate groups are shown in Fig. 5. Constellation groups are evaluated at the i -th level in the order of $G0, G1, \dots, GL$, until the corresponding

$$Q(\underline{s}_{(i)}) = \sum_{j=i}^{N_T} r_{j,j}^2 u_j^2$$

exceeds the maximum value in the list of candidate groups. The hardware blocks in Fig. 5 recursively calculate $Q(\underline{s}_{(i)})$ from $i = N_T$ to $i = 2$ and store the intermediate calculations onto the memory of the list. After these recursive calculations, the list finally contains the desired K candidate groups. Even though the calculation of $Q(\underline{s}_{(i)})$ ("calccost" in Fig. 5) include several multipliers in Fig. 6, the hardware area is only a small fraction (about 6%) of the total area. (Note that K (and thus L) are usually less than 10, as will be shown in Section 6.) Moreover, these additional hardware blocks are only activated once for a specific channel

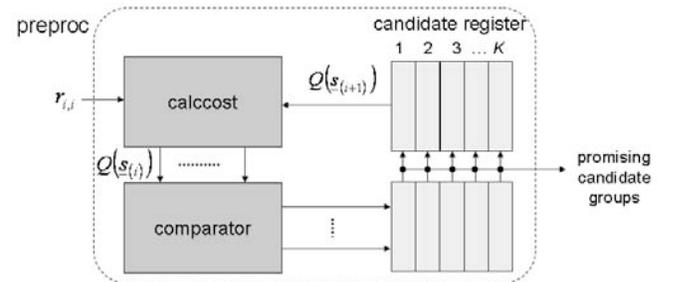


Fig. 5. Architecture for selecting promising candidate groups

realization (as a preprocessing step), implying that the corresponding power consumption becomes negligible.

Once the K candidates are selected, the tree traversal can be easily controlled so that only the selected candidate groups are evaluated: it is easy to know the constellation group of a particular child node by the help of the SE ordering, since $T_i(s_{(i)})$ increases with the size of constellation group. Additionally, as shown in Fig. 5, the constellation groups of the partial vector symbols should be stored onto the memory along with intermediate calculations.

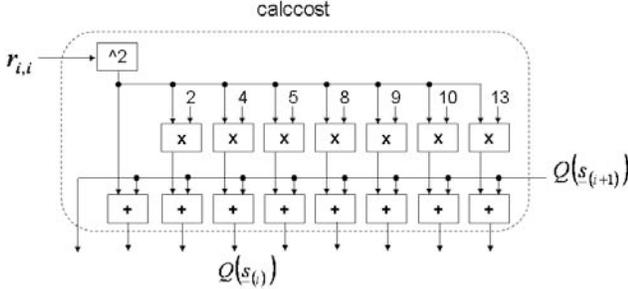
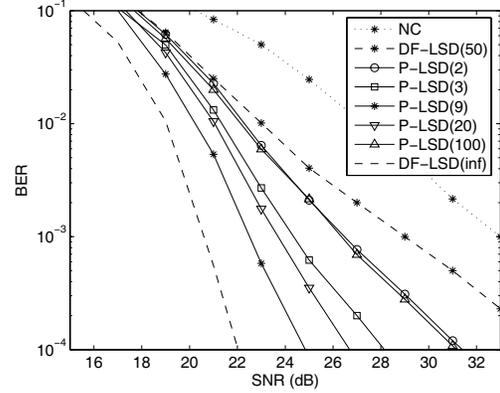


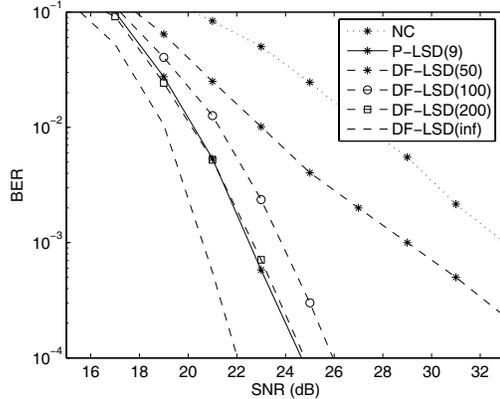
Fig. 6. Details of “calccost” in Fig. 5

6. SIMULATION RESULTS

The error performance of our proposed P-LSD is evaluated for LDPC-coded MIMO-OFDM systems, relying on Monte-Carlo simulations. Here four antennas ($N_T = N_R = 4$), 64-QAM with Gray encoding ($M = 64$), 64-pt FFT, and (1944,972) irregular LDPC codes (proposed in IEEE802.11n) are assumed. The channel is assumed to be frequency-selective (with delay spread of 100 ns) and spatially uncorrelated. The list is assumed to contain the 16 closest candidates ($N_{CD} = 16$) and the belief propagation algorithm with 10 iterations is used for the LDPC decoding. In Fig. 7 (a), the number of visited nodes N is constrained to be 50 with $K = 2, 3, 9, 20$, and 100. For K smaller than 9 (a specific value), the bit error rate (BER) of our proposed P-LSD, P-LSD(2) or P-LSD(3), decreases with K , since the larger K becomes, the more candidates (other than the NC solution) can be evaluated. On the other hand, for K larger than 9, the BER of P-LSD(20) or P-LSD(100) increases with K , since the larger number of less promising candidates are evaluated. Fig 7 (a) shows that our proposed P-LSD, P-LSD(9), outperforms the DF-LSD(50) by an SNR gain of 7 dB at a BER of 10^{-3} . Apart from improvement in error performance, our proposed P-LSD achieves higher throughput than the DF-LSD at a given BER. Fig. 7 (b) shows that the error performance of the DF-LSD with 200 visited nodes (0.005 vectors per cycle), DF-LSD(200), is achievable with our proposed P-LSD with only 50 visited nodes (0.02 vectors per cycle) and 9 candidate groups ($K = 9$), P-LSD(9), implying a throughput increase of 300%. The performance of DF-LSD with unconstrained throughput, DF-LSD(inf), is given in the figures, as the theoretical limit.



(a)



(b)

Fig. 7. Error performance of proposed P-LSD

7. CONCLUSIONS

The probabilistic search algorithm was applied to the state-of-the-art DF-LSD. Our proposed P-LSD evaluates the more promising candidates earlier by simply selecting a few promising candidate groups, while retaining hardware efficiency of the DF-LSD. With a tolerable hardware overhead, the proposed P-LSD provides an SNR gain of 7 dB or a throughput increase of 300%.

REFERENCES

- [1] B. M. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel”, *IEEE Trans. Commun.*, vol. COM-51, pp. 389-399, Mar. 2003.
- [2] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Boelskei, “VLSI implementation of MIMO detection using the sphere decoding algorithm”, *IEEE J. Solid-State Circuits*, vol. 40, pp. 1566-1577, July 2005.
- [3] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, “Silicon complexity for maximum likelihood MIMO detection using spherical decoding”, *IEEE J. Solid-State Circuits*, vol. 39, pp. 1544-1552, Sep. 2004.
- [4] S. Park, K. Lee and S. -C. Park, “Efficient probabilistic sphere decoding architecture”, *Proc. of Int. Symp. on Circuits and Systems*, Kos, Greece, May 21-24, 2006, to be presented.
- [5] W. Zhao and G. B. Giannakis, “Reduced complexity closest point decoding algorithms for random lattices”, *IEEE Trans. Wireless Commun.*, to be published.