A Modified Smart Candidate Adding Algorithm with Fixed-complexity for Soft-output MIMO Detection

Jen-Chieh Chen, Yu-Chieh Ho and Hung-Ta Pai, *Senior Member, IEEE*
Graduate Institute of Communication Engineering, National Taipei University

Abstract — **A multiple-input multiple-output (MIMO) system has been adopted in many modern wireless communication standards. The MIMO system can greatly improve the transmission capacity of the wireless communication. However, its computational complexity at the receiver also increases. The smart ordering and candidate adding (SOCA) algorithm that achieves near max-log optimal error-rate performance with low and fixed computational complexity was proposed recently. The proposed algorithm combines a smart-ordered QR decomposition with candidate adding and a parallel breadth-first search of the detection tree to achieve its desirable performance complexity trade-off.**

Nevertheless, it can provide soft-outputs counterhypotheses so it must try to find out not only the optimal solution but also the counterhypotheses of the optimal solution. In this study, we propose a method to reduce the complexity of searching for the counterhypotheses so generated. This technique can be reduce the search complexity while achieving a near-optimal performance.

I. Introduction

Today people are increasingly dependent on the convenience of wireless communications. The wireless communication is also a growing demand for data transmission. The use of multiple antennas at the transmitter and the receiver leads to a MIMO channel and the channel has already been an indispensable part of the wireless communication. It constitutes a basic structure in current and upcoming wireless communication standards such as IEEE 802.11n (Wi-Fi), 4G, 3GPP Long Term Evolution (LTE) [1], WiMAX and HSPA+. It can significantly increase the data rate and the reliability of a wireless communication link, without the additional transmit power or additional bandwidth. However, the transmission signals in the MIMO channel are mixed at the receiver. A practical receiver must separate them efficiently interfered other.

A MIMO detector is deployed to compute a posteriori probability (APP) for each of the coded bits, which is the conditional probability that each coded bit is 1 (or 0) given the observation of the channel output. The complexity of exact APP computation grows exponentially with the spectral efficiency. The SOCA algorithm [2] is proposed to achieve a desired performance-complexity trade-off. We propose an improved tree-search method based on the SOCA algorithm to reduce the computational complexity of the SOCA algorithm without sacrificing error-rate performance. The contribution in the SOCA algorithm is a list detection algorithm that closely approximate the max-log optimal counterhypothesis list with a low and fixed computational complexity.

The MIMO system can improve the decline in the quality caused by multi-path effects of communication transmission, and increase the transmission data rate without extra bandwidth. Its mentioned before, due to the presence of plurality of transmitters and receivers therefore received signals have overlapping phenomena. It is necessary to develop an algorithm to detect transmitted signals. The well-known detection methods include Maximum Likelihood (ML), Minimum Mean Square Error (MMSE), Zero Forcing (ZF), and Successive interference Cancelation (SIC) algorithms. Among them, the ML algorithm has optimal error-rate performance, but its complexity is very high and increases as the number of antennas and modulation symbols.

Soft-output MIMO detectors have been proposed based on a variety of principles, including Monte Carlo methods, semidefinite programming, interference cancellation, sphere projection, and tree search. Of particular interest is the smart candidate adding (SCA) approach, where a maximum a posteriori (MAP) estimate (or an approximation thereof) is supplemented by directed searches for counterhypotheses to this estimate. An improvement over the SCA approaches was proposed to find its list using a single pass through the detection tree, rather than using multiple searches. However, the SCA algorithm suffer from the problem of variable computational complexity and a potentially high worst case complexity.

The SOCA algorithm is a low- and fixed-computational complexity tree-based solution to the soft-output MIMO detection problem that never visits nodes in the detection tree more than once. That sacrifices max-log optimality for low and fixed computational complexity. The SOCA algorithm achieves a desirable performance- complexity profile by combining an intelligent ordering algorithm with SCA techniques and a parallel search of the detection tree.

The remaining of the paper are organized as follows. Section II introduces the system model. Section III gives the detail of the soft-output SOCA algorithm. Section IV present a modified of our proposed algorithm and its simulation result will be presented. Finally, we conclude in Section V.

II. System Model

In the MIMO system, multiple antennas can be used at both the transmitter and receiver to improve communication performance. It is one of several forms of smart antenna technology. The input **u** is a vector of independent and identically distributed (i.i.d.) information bits that is encoded and interleaved. The coded bit stream is partitioned into blocks **c** of ωN_t bits. Each block is mapped onto a vector whose N_t component symbols are taken from a QAM alphabet of size $q = |K| = 2^{\omega}$, where ω is the number of bits per symbol. We define $Z = K^{N_t}$ as the set of all possible symbol vectors $\mathbf{x} \in Z$, one for each binary vector $\mathbf{c} \in \{\pm 1\}^{\omega N_t}$, as determined by the mapping from coded bits to transmitted symbols. The vector of symbols **x** is transmitted through an $N_r \times N_t$ MIMO channel whose equivalent complex baseband model is

$$Y \equiv \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{N_r} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_t} \\ h_{22} & h_{22} & \cdots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{N_r2} & \cdots & h_{N_rN_t} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{N_r} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ n_{N_r} \end{bmatrix}$$
$$\equiv \mathbf{H}_{N_r t}\mathrm{x} + \mathrm{n}\ , \quad (1)$$

Corresponding author:popo782@gmail.com
DOI:10.6159/IJSE.2014.(4-1).46

where y_j denotes the symbol received at the j antenna, **y** is the received signal vector, n_j denotes the additive white Gaussian noise (AWGN) at the j receive antenna, that the components of **n** are zero-mean, circularly symmetric, i.i.d. complex Gaussian random variables with variance N_o, and h represents the complex channel gain between the *j*th receive and the *i*th transmit antennas. We assume the channel is Rayleigh-faded. The signal-to-noise ratio (SNR) at any receive antenna isSNR $= E_b/N_o$.

We focus on a MIMO detector at a receiver. Numbers of researchers have proposed many practical algorithms to reduce the complexity of the detector. The SOCA has got much attention and been esteemed as one of the most acceptable detectors implemented in hardware owing to its low and fixed complexity.

III. SOCA ALGORITHM

Soft-output MIMO detection requires the computation of LLRs, denoted as $LLR(\cdot)$, for all bits in the label b. In order to reduce the corresponding computational complexity, we employ the *max-log approximation*

$$\mathrm{LLR}\left(c_j|y\right) = \min_{\tilde{x}\in Z_j^{(-1)}} \| y - H\tilde{x} \|^2 - \min_{\tilde{x}\in Z_j^{(1)}} \| y - H\tilde{x} \|^2 \qquad (2)$$

The aim of a soft-output detector is to calculate or approximate the a posteriori probability (APP) for each of the code bits c_j in a given signaling interval, where $j \in \{1, \dots, \omega N_t\}$ is the bit index. An exact solution to (2) need only consider a list containing the MAP candidate along with at most ωN_t counterhypotheses. List detection is the process of finding a list of candidates $L \subseteq Z$. From this list, (2) is approximated as

$$\mathrm{LLR}\left(c_j|y\right) = \min_{\tilde{x}\in L\cap Z_j^{(-1)}} \| y - H\tilde{x} \|^2 - \min_{\tilde{x}\in L\cap Z_j^{(1)}} \| y - H\tilde{x} \|^2 \qquad (3)$$

The only difference between (2) and (3) is the insertion of the list L, whose size is typically much less than that of Z. The list size$l = |L|$ plays a critical role in the overall complexity and performance. The root node in the tree at layer 0 has q children. Each child node (now parent nodes) has q children. This continues until there are qN_t leaf nodes at layer N_t. The detection process can hence be interpreted as a search for leaf nodes in a tree, corresponding to elements of a list L. In other words, the SOCA algorithm finds L using a standard detection tree.

The SOCA algorithm is mainly divided into two stages, a preprocessing stage and a core processing stage. The preprocessing stage is used to determine the mapping between layers in the detection tree and the transmitted vector of information symbols. The core processing finds the list L, the output of the SOCA algorithm. Because the preprocessing can be considered as a performance enhancement, we begin by describing the core processing.

After a QR-decomposition of the channel matrix, H=QR, squared Euclidean distance for a candidate $\tilde{x}$ is

$$\begin{aligned} J(\tilde{x}) &= \| y - QR\tilde{x} \|^2 \\ &= \| Q^H (y - QR\tilde{x}) \|^2 \\ &= \| \tilde{y} - R\tilde{x} \|^2 \\ &= \sum_{i=1}^{N_t} \left| \tilde{y}_i - \sum_{j=1}^{i} R_{i,j}\, \tilde{x}_j \right|^2 \end{aligned} \qquad (4)$$

where Q is an orthogonal matrix of dimension $N_r \times N_t$ and R is an upper-triangular matrix of $N_r \times N_t$, and $\tilde{y} = Q^H y$. Notably, R has real and positive diagonal entries. The cost function (4) can be interpreted as the sum of branch metrics, one for each branch in a path from the root to a leaf node, where the metric for a branch in the *j*th stage of the detection tree is

$$\left| \tilde{y}_i - \sum_{j=1}^{i} R_{i,j}\, \tilde{x}_j \right|^2 \qquad (5)$$

Here, the SOCA algorithm is used with a smart-ordered QR decomposition[2], and we will not discuss it.

The foundation of the SOCA algorithm is a simple breadth-first strategy for searching the tree that is closely related to the M algorithm [3]. It moves through the tree one layer at a time and discards all but a subset of bi surviving child nodes at layer i. The candidate node with the best metric is identified as the partial MAP (PMAP) node which is the best surviving node from the set of current candidate nodes at layer i. Assume that $\hat{c}^{\mathrm{PMAP}}$ is the last ω bits corresponding to the node with the best metric, the ω sibling nodes of the PMAP node by simply flipping each of the last $\hat{c}^{\mathrm{PMAP}}$ in turn are added to the candidate set. The bit-flipping method has the advantages of low complexity, but there is also two disadvantages as follow 1) The counterhypotheses may not be the ones with the best metrics. 2) A counterhypothesis for the bit may already be represented in the candidate set. As a result, these counterhypotheses may be immediately pruned after being added. That is, no performance gain is obtained, but the computational complexity is increased in the SOCA algorithm.

The number of branch metric computations for the SOCA algorithm as

$$\begin{aligned} u &= b_1 + \sum_{i=1}^{N_t} b_1 + \omega(i-1) \\ &= N_t\left(b_1 + \frac{\omega(N_t-1)}{2}\right) \end{aligned} \qquad (6)$$

Usually, $\mathbf{b} = [b_1\ 1 \dots 1]$, where $b_2 = b_3 = \cdots = b_{N_t}$ The list size for the SOCA algorithm is given by

$$l = b_1 + \omega(N_t - 1) \qquad (7)$$

IV. PROPOSED SCHEME AND SIMULATIONS

A. A Modified Smart Candidate Adding Algorithm:

The candidate adding for the SOCA algorithm is flip each of best node in turn. Although the bit-flipping method has the advantages of low complexity and ensure the presence of the counterhypotheses, these counterhypotheses may be immediately pruned after being added. As a result, the performance gain is obtained by adding these counterhypotheses but the computational complexity is increased. We are going to solve this problem.

We focus on the improvement of the candidate adding and propose a screening method to select surviving nodes from the set of current candidate nodes for the next layer. The improved candidate-adding method is summarized as follows.

Step 1. When the SOCA algorithm moves to the *i*th layer,$i = 2,3, \dots, N_t$, identify the candidate node with the best metric as the maximum a posteriori node and the best surviving nodes b_i form the set of current candidate nodes.

Step 2. Add the ω sibling nodes to the candidate set, each of that has only one bit different from the last ω bits of the partia maximum a posteriori node.

Step 3. Prune m worst nodes from the set of current candidate nodes, where m = ω is the worst metric of the best surviving nodes.

Pruning of the proposed method can reduce the nodes of tree traversal, and then reduce the computational complexity. When $\mathbf{b} = [b_1\ 1 \ldots\ 1]$and m =ω, the number of branch computation is

$$u = 2 \times b_1 + \sum_{i=2}^{N_t-1} b_1 + \omega(i-1)$$
$$= N_t\left(b_1 + \frac{\omega(N_t-1)}{2}\right) - m \times (N_t - 1) \qquad (8)$$

In addition, the number of branch metric computations will be modified to

$$l = b_1 + \omega(N_t - 1) - m \qquad (9)$$

Although the computational complexity can be reduced that **m** nodes are pruned, but if $m >$ ω will result in performance degradation. Therefore, we recommend the conditions $m \leq$ **ω** will achieve the best trade-off.

B. Simulation Results:

We employ a block interleaver and a code polynomial [133 171] of a convolutional code with constraint length 7 in 1/2 code rate and the information block size (including tail bits) is 360 and 512. All the codes are transmitted over the Additive White Gaussian Noise (AWGN) channel with slow-fading. Performance is measured in terms of the in E_b/N_0 (dB) required to achieve a frame error rate (FER) of 10^{-2}. The target FER of 10^{-2} was selected because it is common to design systems for this error rate [4].After calculation, a probability of the counterhypothesis was not found is approximately 13%.

The 4×4 MIMO systems with 16-QAM and 64-QAM are deployed to demonstrate the reduction of the computational complexity. We combine our method with the SOQR decomposition[2]. The improvement is shown in Figures 1, 2, 3, 4, 5. The results with/without modification are summarized in Table 1, 2 and 3.

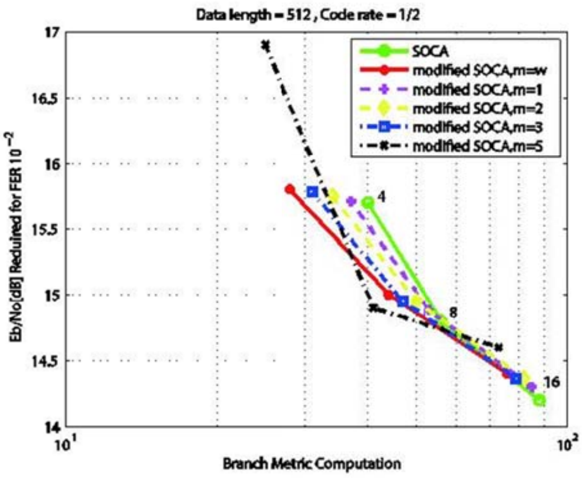

Fig. 1 Performance versus complexity for soft-output 4 × 4 MIMO detection schemes using 16-QAM transmission in slow Rayleigh fading.

TABLE Ⅰ

Comparison in 4×4 16-QAM (average number of visited nodes)

b_1	Without modification	With modification	complexity reduction
16	88	76	13.63%
8	56	44	21.42%
4	28	28	30.00%

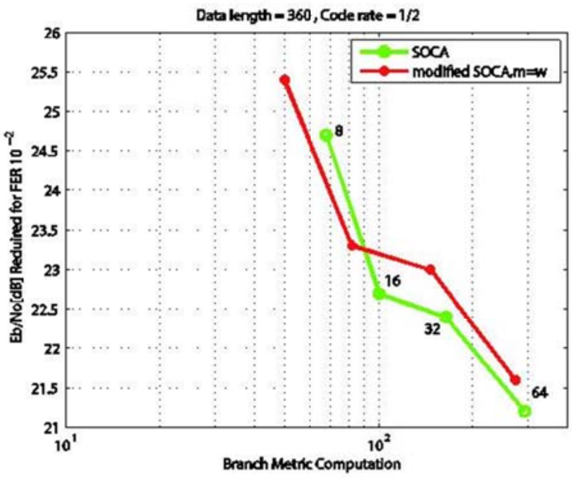

Fig. 2 Performance versus complexity for soft-output 4 × 4 MIMO detection schemes using 64-QAM transmission in slow Rayleigh fading.

TABLE Ⅱ

Comparison in 4×4 64-QAM (average number of visited nodes)

b_1	Without modification	With modification	complexity reduction
64	292	274	6.16%
32	164	146	10.97%
16	100	82	18.00%
8	68	50	26.47%

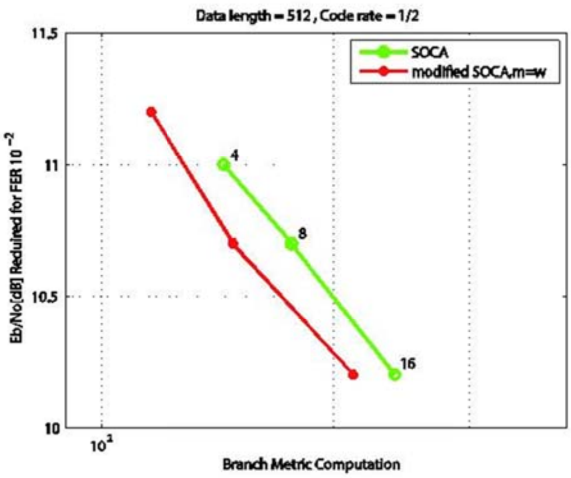

Fig. 3 Performance versus complexity for soft-output 8 × 8 MIMO detection schemes using 16-QAM transmission in slow Rayleigh fading.

TABLE Ⅲ

Comparison in 8×8 16-QAM (average number of visited nodes)

b_1	Without modification	With modification	complexity reduction
16	270	212	11.66%
8	176	128	15.90%
4	144	116	19.44%

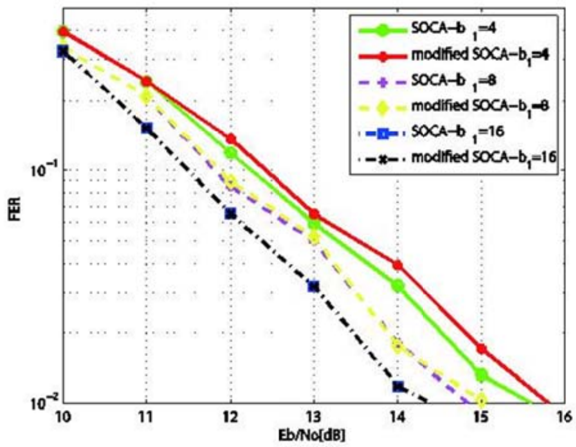

Fig. 4 Performance result for soft-output 4×4 MIMO detection schemes using 16-QAM transmission in slow Rayleigh fading.

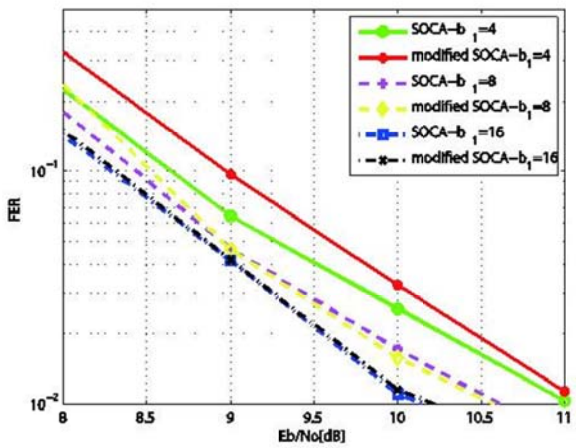

Fig. 5 Performance result for soft-output 8×8 MIMO detection schemes using 16-QAM transmission in slow Rayleigh fading.

V. CONCLUSION

We have presented and modified the SOCA algorithm to further reduce the computational complexity with fixed-complexity for soft-output MIMO detection. Compared to the SOCA algorithm, the proposed method can be up to 30% complexity decrease in a 4 × 4 MIMO system, and 19.44% complexity decrease in the 8×8. Specifically, the reduction in computational complexity is fixed and determined by the number of antennas of the MIMO system, and numbers of the best surviving nodes b_i will directly a affect the overall computational complexity when m = ω. We have found a more efficient MIMO detection algorithm as indicated in this thesis.

REFERENCES

[1] A. Leon-Garcia, *LTE*. Addison-Wesley, 1994.

[2] J. R. B. David L. Milliner, Ernesto Zimmermann and G. Fettweis, "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE Trans. Signal Process.*, vol. 3, no. 6, pp. 1016-1025, Dec. 2009.

[3] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis, ",*IEEE Trans. Commun*., Feb. 1984.

[4] C. Studer, A. Burg, and H. B olcskei, "Soft-output sphere decoding: algorithm and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 4827{4842, Feb. 2008.