Step reduced K-best sphere decoding

Xinyu Mao, Yuxin Cheng, Lili Ma and Haige Xiang
School of Electrical and Computer Engineering
Peking University, Beijing, China, 100871
Email: (xymao, chengyx, malili and xianghg)@pku.edu.cn

***Abstract*—We propose an algorithm that reduces the complexity of the K-best sphere decoding (K-best SD) algorithm, which is a powerful parallel detection algorithm for multiple-input multiple-output systems (MIMO). By analyzing the probability of different nodes to be the final solution, the algorithm prunes some nodes during the tree search to reduce the complexity. Simulation results prove that compared with the K-best SD algorithm the proposed algorithm performance drops very little. Compared with the famous fixed-complexity sphere decoding (FSD) with the same complexity, the proposed algorithm has better performance.**

I. Introduction

Multiple-input multiple-output (MIMO) systems have been recognized as one of the major technologies to achieve high data rate services for current and future communication systems [1]. In MIMO detection, there exists a conflict between the performance and calculation complexity [2]. The maximum likelihood (ML) detection provides the optimal performance. But the computational complexity of ML grows exponentially with the number of the antennas and the modulation order [3]. The linear algorithms such as the zero forcing (ZF) and the minimum mean square error (MMSE) can be implemented in low computational complexity, but the performances they provide are not good enough. Sphere decoding (SD) [4] [5] and K-best sphere decoding (K-best SD) [6] achieve the optimal performance or close to the optimal performance with manageable complexity. The K-best SD algorithm, which can work in parallel with fixed complexity, attracts much attention and finds many applications [7]. Accompanied with the equalization and coding, the K-best SD, especially the soft-output K-best SD, can achieve very high performance with low complexity [8]. But to lower the calculation of K-best SD is still very important and many works have been done [9] [10] [11]. Among them, the most famous algorithm is the fixed-complexity sphere decoding (FSD) [12], which is welcomed for its capability of parallel computation, low complexity and good performance. But the effort to low the complexity has not stopped [13] [14].

In this paper, a step reduced K-best SD is proposed. We reduce the number of child nodes. The expanding numbers of nodes are not constant. Instead, they depend on the rank of the nodes metrics list. The smaller the node's metric is, the larger number child nodes it has.

II. MIMO systems

The MIMO systems with Nt transmit antennas and Nr receive antennas can be modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{s}$ is the $Nt\times 1$ transmitted symbol vector, $\mathbf{y}$ is the $Nr\times 1$ received symbol vector. $\mathbf{n}$ is the $Nr\times 1$ independent identically distributed (i.i.d.) Gaussian noise. H is the $Nr \times Nt$ channel matrix. The transmitted symbol is chosen from a constellation with $M = 2^m$ signal points, where m is the modulation order. The ML algorithm is to find a signal point which satisfies

$$\mathbf{s} = \arg\min_{\mathbf{s}\subset\aleph} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \tag{2}$$

where $\aleph$ is the set containing all the possible transmitted symbol vectors.

III. Tree search and detection algorithm

The calculation complexity of the ML algorithm is almost always too high to be executed. Many works have been done to simplify the complexity. Before further disposition, channel matrix in (1) should be transformed into a triangular matrix. The most common transformation is the QR decomposition. It is known that $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where $\mathbf{R}$ is a $Nr \times Nt$ upper triangle matrix, and $\mathbf{Q}$ is a $Nr \times Nr$ unitary matrix with $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$. Left multiplied by $\mathbf{Q^H}$, (1) is transformed into

$$\boldsymbol{\rho} = \mathbf{R}\mathbf{s} + \boldsymbol{\eta} \tag{3}$$

where $\boldsymbol{\rho} = \mathbf{Q}^H\mathbf{Y}$ and $\boldsymbol{\eta} = \mathbf{Q}^H\mathbf{n}$.

Then the ML algorithm can be expressed as

$$\hat{\mathbf{s}} = \arg\min_{\aleph} \|\boldsymbol{\rho} - \mathbf{R}\mathbf{s}\|^2 \tag{4}$$

Next, to simplify the expression, let $N = Nt$, (3) can be rewritten as

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_N \\ \rho_{N+1} \\ \vdots \\ \rho_{Nr} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1N} \\ 0 & r_{22} & \dots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{NN} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_N \\ \eta_{N+1} \\ \vdots \\ \eta_{Nr} \end{pmatrix} \tag{5}$$

So the solution of (4) can be searched layer by layer from the N^{th} to the 1^{st} like Fig. 1. In this figure, every circle represents a possible transmitted signal in (5). The circle is called a node and the whole figure is called a search tree.

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

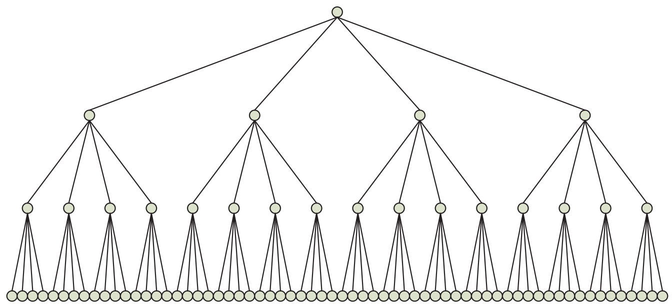

Fig. 1. ML algorithm and tree search

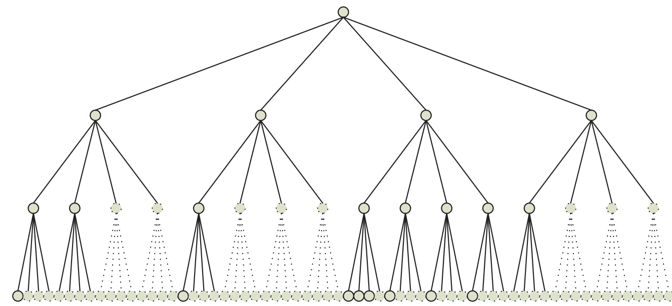

Fig. 2. K-best SD algorithm

A. Euclidean distance and the partial Euclidean distance

The Euclidean distance (ED) between transmit and receive symbol is defined as

$$\Phi = \left(\sum_{j=1}^{N} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2 \right)^{\frac{1}{2}} \tag{6}$$

Define the partial Euclidean distance (PED) as

$$\Phi_i = \left(\sum_{j=i}^{N} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2 \right)^{\frac{1}{2}} \tag{7}$$

The solution of (4) is to find the $\hat{\mathbf{s}}$ with the smallest ED.

B. The K-best SD and FSD algorithm

K-best SD visits all child nodes of the kept nodes and keeps a fixed number of nodes for each layer in the search tree no matter how many nodes are visited. At the end of each layer search, all visited nodes should be ranked in increasing order according to their PEDs. Only the first K nodes will be kept, like Fig. 2. The solid line points to the visited node, the solid node means the kept node and the dotted node means the abandon node. As a result, the number of visited nodes increases linearly with the layer number.

The FSD algorithm tries to lower the complexity of the K-best SD algorithm. During the tree search, when more transmitted signals in the vector are worked out, the interference in the received signals becomes less. It is easier to search nodes near the bottom of the tree than to search nodes near the root. The number of expanded child nodes can be shrunk because some transmitted signals are known. Define ns_i as the number of child nodes per parent node in layer i. In the FSD algorithm the number of child nodes should satisfy

$$ns_N \geq ns_{N-1} \geq ... \geq ns_1 \tag{8}$$

The nodes in the previously visited layer are more important than those in the later visited layer. To improve the performance, the child nodes number in the previously visited layer is set as large as possible. If we put the child nodes number of all layers together in a vector $\mathbf{ns} = \begin{pmatrix} ns_1 & ... & ns_{N-1} & ns_N \end{pmatrix}$, the child node number vector of FSD is often set as $\mathbf{ns} = \begin{pmatrix} 1 & ... & 1 & M & ... & M \end{pmatrix}$, like Fig. 3.

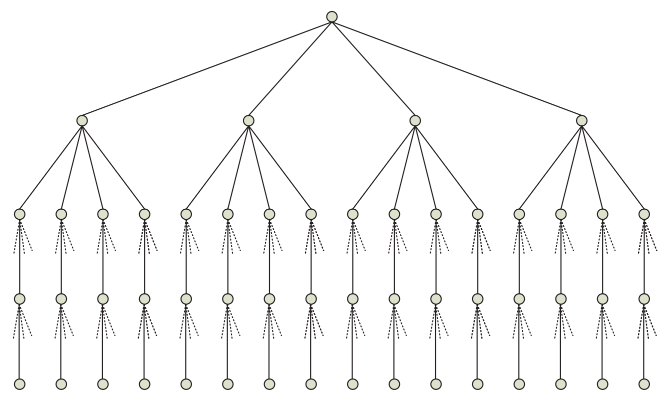

Fig. 3. fixed-complexity sphere decoding algorithm

IV. Step Reduced K-best Sphere Decoding

Since some visited nodes will not be kept, it is reasonable to deduce that some of the nodes visits can be omitted. The aim of detection algorithm is to find a vector $\hat{\mathbf{s}}$ which satisfies (4), that is $\hat{\mathbf{s}}$ has the smallest ED, where ED of $\hat{\mathbf{s}}$ is expressed as (6). Consider that $N-i$ layers have been detected and i layers need to be detected. The square of Φ can be written as

$$\begin{aligned} &\Phi^2 \\ &= \sum_{j=1}^{i-1} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2 + \sum_{j=i}^{N} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2 \\ &= \bar{\Phi}_i^2 + \Phi_i^2 \end{aligned} \tag{9}$$

where $\Phi_i^2 = \sum_{j=i}^{N} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2$ and $\bar{\Phi}_i^2 = \sum_{j=1}^{i-1} \left(\rho_j - \sum_{k=j}^{N} r_{jk} \hat{s}_k \right)^2$. For the nodes in i^{th} layer, Φ_i is known, and $\bar{\Phi}_i$ is unknown. Define $\Phi_i(a)$ as the PED of the node that ranks a in layer i, where $a = 1...K$. The PEDs of the nodes are ranked from the smallest to the largest, as follows

$$\Phi_i^2(1) \leq \Phi_i^2(2) \leq ... \leq \Phi_i^2(K) \tag{10}$$

Define the ED according to the rank in layer i as $\underset{i}{\Phi}(a)$, we have

$$\underset{i}{\Phi}^2(a) = \bar{\Phi}_i^2(a) + \Phi_i^2(a) \tag{11}$$

When the i^{th} layer is being searched, there is no PED information from layer $i-1$ to layer 1. The unknown branch metrics can be deemed as Gaussian noise, as in (7), we can get

$$E\left(\bar{\Phi}_i^2(a_1)\right) = E\left(\bar{\Phi}_i^2(a_2)\right) \quad when \quad a_1 \neq a_2 \tag{12}$$

Consider (10), (11) and (12), we have

$$E\left(\Phi_i^2(a_1)\right) < E\left(\Phi_i^2(a_2)\right) \quad when \quad a_1 < a_2 \tag{13}$$

(13) shows that the smaller the node's PED, the more probability it is the ML solution. So given (13), we conclude that if a node ranks before another node in the same layer, it is more likely to be the final solution. If a certain node has less opportunity to be the final solution, the number of its child nodes can be set smaller to reduce the calculation complexity. So we propose that the number of child nodes to be visited for different nodes is based on the nodes rank in PED list.

There are many kinds of schemes to set the number of child nodes of different groups. Here we propose a directly-thinking scheme. Specifically, in a MIMO system, the constellation size is M for each layer, the kept nodes number is K and we assumed that $M \leq K$, which is almost always satisfied. K nodes in each layer are divided into M groups according to their PED ranks. Denote $\mathbf{ng} = \left(\begin{array}{cccc} ng_M & ng_{M-1} & \ldots & ng_1 \end{array}\right)$ as the number of nodes in each group, the element of $\mathbf{ng}$ can be expressed separately. We only consider $\mathrm{mod}(K, M) = 0$ here. The conclusion can be extended to $\mathrm{mod}(K, M) \neq 0$.

$$ng_i = K/M \tag{14}$$

Denote $\mathbf{nng} = \left(\begin{array}{cccc} nng_M & nng_{M-1} & \ldots & nng_1 \end{array}\right)$ as the vector of child nodes number in every group. It is

$$\begin{array}{l} \mathbf{nng} = \left(\begin{array}{cccc} M & M-1 & \ldots & 1 \end{array}\right) \\ or \quad nng_i = i \end{array} \tag{15}$$

The nodes in the M^{th} group are those nodes those have the first K/M smallest PEDs. The number of child nodes of the node in this group is the maximum possible number M. Then this number is reduced group by group. It becomes $M-1$, $M-2$ for the subsequent groups. It is 1 in the last group, as in Fig. 4. In Fig. 4 between layer $N-1$ and $N-2$, the node pointed by the solid line means the visited nodes. The node pointed by the dotted line means the node that is not visited in the proposed algorithm and is visited in the K-best SD. The number of nodes pointed by dotted lines is the reduced number in the proposed algorithm compared with the K-best SD.

There is almost no additional calculation in the proposed algorithm although it is necessary to classify the nodes. The major calculation complexity comes from the calculations related to the nodes. The calculation complexity is almost proportional to the number of visited nodes. The proposed algorithm drops the number of visited nodes of the original K-best SD. When K and M are large enough, it is one half of the total number of the visited nodes.

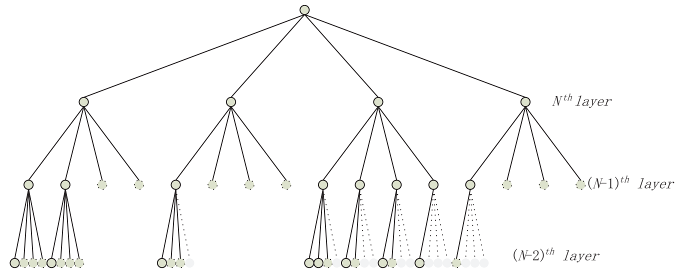

Fig. 4. step reduced K-best sphere decoding algorithm

The proposed algorithm reduces the visited nodes number of the K-best SD algorithm while the number of kept node does not change. If the visited nodes number kept unchanged, the K-best SD algorithm has the smaller K number than the proposed algorithm. The kept nodes number of the K-best SD algorithm is the main factor that determines the performance, so that the K-best SD with the smaller K number is meant to have the worse performance. The performance of the proposed algorithm drops not too much, and it promotes the performance of K-best SD algorithm with the same number of visited nodes. A couple of numerical examples will be given out in the next section to prove that.

V. Simulation Results

The simulation results about the performance of the proposed algorithm are shown here. A 4×4 MIMO system and an 8×8 MIMO system are considered in Fig. 5 and Fig. 6. The K-best algorithm is denoted as KSD, the step reduced K-best algorithm is written as SRKSD, and the FSD algorithm is shown as FSD in both figures. We execute the search in real field instead of complex field, so that the number of tree layer is twice the number of antennas.

Fig. 5 shows a 4×4 MIMO system with 16QAM, 64QAM and 256QAM. The breadth of the original K-best SD is $K = 5$ for 16QAM, $K = 12$ for 64QAM, $K = 26$ for 256QAM. The path vector of the FSD is $\mathbf{ns} = (1, 1, 1, 1, 1, 1, M, M)$, where M is the size of the constellation. The visited number of one layer of FSD is 16, 64 or 256. The visited number of one layer of the proposed algorithm is 14, 64 or 251, which is equal to or smaller than that of the FSD. That means the complexity of FSD is no less than that of the proposed algorithm. From fig. 5, it can be found that the performance of the proposed algorithm is almost the same as the K-best SD and a little better than that of the FSD.

An 8×8 MIMO system with 16QAM, 64QAM and 256QAM is considered in Fig. 6. The breadth of the original K-best SD is $K = 25$ for 16QAM, $K = 112$ for 64QAM, $K = 480$ for 256QAM. The path vector of the FSD is $\mathbf{ns} = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, M, M, M)$. The number of visited node of the FSD is 64, 512 and 4096. It can be found that the performance of the proposed algorithm has almost the same performance as that of the K-best SD which has the same breadth and almost the twice complexity, and better than that of the FSD algorithm. Specifically, when BER=10^{-3}, the

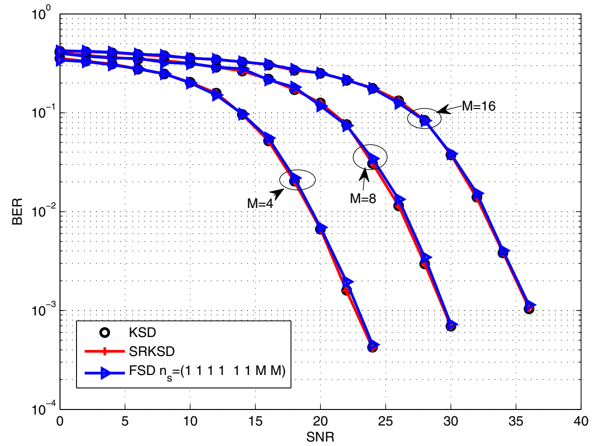

Fig. 5. performance compare in a 4-by-4 MIMO system

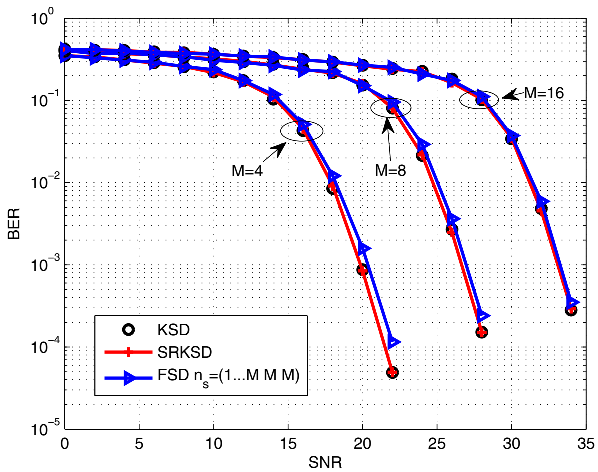

Fig. 6. performance compare in an 8-by-8 MIMO system

performance of the FSD algorithm is about 20.5dB, 27.1dB or 33.0dB. The performance of the proposed algorithm and K-best SD algorithm is about 19.8dB, 26.4dB or 32.6dB for 16QAM, 64QAM or 256QAM. The proposed algorithm gains 0.7dB, 0.5dB or 0.4dB compared to the FSD with even more calculation complexity.

VI. Conclusion

In this paper, a new efficient simplified K-best SD has been proposed. The child nodes numbers of different nodes in one layer is reduced step by step according to their PED rank. The calculation complexity is fixed and can be calculated parallel. The structure of it is suitable for the real-time system implement. Simulation results show that the proposed algorithm achieves performance very close to that of the K-best SD with the same breadth and twice complexity. Its performance surpasses the famous FSD algorithm with the same calculation complexity or even a little higher complexity.

References

[1] A. Paulraj, D. Gore, R. Nabar, and H. Bolcskei, “An overview of MIMO communications - a key to gigabit wireless,” *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[2] A. Goldsmith, S. Jafar, N. Jindal, and S. Vishwanath, “Capacity limits of MIMO channels,” *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 5, pp. 684–702, 2003.

[3] X. Zhu and R. Murch, “Performance analysis of maximum likelihood detection in a MIMO antenna system,” *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 187–191, Feb. 2002.

[4] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[5] C. P. Schnorr and M. Euchner, “Lattice basis reduction: Improved practical algorithms and solving subset sum problems,” *Mathematical Programming*, vol. 66, no. 1-3, pp. 181–199, Aug. 1994.

[6] K. wai Wong, C. ying Tsui, R. Cheng, and W. ho Mow, “A VLSI architecture of a k-best lattice decoding algorithm for MIMO channels,” in *2002 IEEE International Symposium on Circuits and Systems. Proceedings*, Phoenix-Scottsdale, AZ, USA, 2002, pp. III–273–III–276.

[7] H. G. Kang, I. Song, J. Oh, J. Lee, and S. Yoon, “Breadth-First signal decoder: A novel Maximum-Likelihood scheme for Multi-InputCMulti-Output systems,” *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 3, pp. 1576–1584, 2008.

[8] B. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *Communications, IEEE Transactions on*, vol. 51, no. 3, pp. 389–399, 2003.

[9] C. Shen and A. M. Eltawil, “A radius adaptive K-Best decoder with early termination: Algorithm and VLSI architecture,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 9, pp. 2476–2486, 2010.

[10] X. G. Dai, S. W. Cheung, and T. I. Yuk, “Simplified ordering for fixed-complexity sphere decoder,” in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC ’10*, Caen, France, 2010, p. 804.

[11] X. Mao, S. Ren, L. Lu, and H. Xiang, “A reduced complexity K-Best SD algorithm based on Chi- Square distribution for MIMO detection,” in *2011 IEEE Vehicular Technology Conference (VTC Fall)*. San Francisco, United States: IEEE, Sep. 2011, pp. 1–4.

[12] L. Barbero and J. Thompson, “Fixing the complexity of the sphere decoder for MIMO detection,” *Wireless Communications, IEEE Transactions on*, vol. 7, no. 6, pp. 2131–2142, 2008.

[13] C. Xiong, X. Zhang, K. Wu, and D. Yang, “A simplified fixed-complexity sphere decoder for V-BLAST systems,” *Communications Letters, IEEE*, vol. 13, no. 8, pp. 582–584, 2009.

[14] C. Zheng, X. Chu, J. McAllister, and R. Woods, “Real-Valued Fixed-Complexity sphere decoder for high dimensional QAM-MIMO systems,” *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4493–4499, Sep. 2011.