

Pipeline structure Schnorr-Euchner Sphere Decoding Algorithm

Xinyu Mao, Jianjun Wu, Haige Xiang

State Key Laboratory of Advanced Optical Communication Systems & Networks,
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
Email: xymao@pku.edu.cn, just@pku.edu.cn, xianghg@pku.edu.cn

Received May, 2013

ABSTRACT

We propose a pipeline structure for Schnorr-Euchner sphere decoding algorithm in this article. It divides the search tree of the original algorithm into blocks and executes the search from block to block. When one block search of a signal is over, the part in the pipeline structure that processes this block search can load another signal and search. Several signals can be processed at the same time in one pipeline. Blocks are arranged to lower the whole complexity in the way that the previously search blocks are the blocks those have more probability to generate the final solution. Simulation experiment results show the average process delay can drop to the range from 48.77% to 60.18% in a 4-by-4 antenna system with 16QAM modulation, or from 30.31% to 61.59% in a 4-by-4 antenna system with 64QAM modulation.

Keywords: Multiple-Input Multiple-Output System; Schnorr-Euchner Sphere Decoding; Pipeline Structure

1. Introduction

With the rapid development of wireless communications, the desire for high speed data service conflicts with the situation that fewer and fewer band width is available. Multiple-input multiple-output (MIMO), which can provide dramatic high spectral efficiency, is the key technology to resolve this conflict. MIMO is one of the most exciting technologies of the last decade [1,2].

The MIMO detection is rather a complex problem. The maximum likelihood (ML) detection provides the optimal performance. But its complexity is so high that it can not be used in actual systems. The linear detection, such as the zero forcing and minimum mean square error detection have low computation complexity. But their performance is very poor. The Vertical Bell Laboratories Layered Space Time (VBLAST) detection provides higher performance than the linear detection with the price of small complexity increase [3]. But the performance is not good enough.

Sphere decoding (SD) algorithm was proposed to provide optimal performance or near optimal performance with low complexity [4]. So it attracts much attention as soon as it was proposed [5]. Schnorr-Euchner SD (SESD) algorithm is an optimized SD algorithm [6]. It can find the optimal solution faster than the original SD algorithm and the modification from SD to SESD is very small. So it is the most popular optimal performance algorithm [7-15].

However, there are two main drawbacks which restrict the usage of the SESD in actual systems, especially the real time systems. One is that the calculation complexity of the SESD is not fixed and changes in a large range. Another is that it is not suitable to use in pipeline structure.

In this article, we propose a pipeline structure for SESD algorithm. The searching process of SESD algorithm is divided into blocks and can be executed sequentially. When the multiple threads searching is executed simultaneously, the processing time decreases. Simulation experiment results show that the proposed structure can save remarkable processing time especially in low signal to noise ratio (SNR).

2. MIMO System and ML Detection

MIMO system can be depicted as an equation

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (1)$$

where \mathbf{s} is the $N_t \times 1$ transmit vector. \mathbf{y} is the $N_r \times 1$ receive vector, and \mathbf{n} stands for the independent and identical distributed complex zero-mean Gaussian noise. \mathbf{H} is the $N_t \times N_r$ channel matrix where h_{ij} is the complex channel gain and satisfies $E[\|h_{ij}^2\|] = 1$. Usually when $N_r \geq N_t$, equation (1) can be calculated.

ML algorithm is to find the solution of

$$\hat{\mathbf{s}}_{ML} = \arg \min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \quad (2)$$

where \mathcal{S} means the set of all possible transmit vectors.

3. Tree Search and Euclidean Distance

For further disposition, the QR decomposition is always used to make the channel matrix triangular. We know that every matrix can be express as

$$\mathbf{H} = \mathbf{QR} \quad (3)$$

equation (1) will be expressed as

$$\boldsymbol{\rho} = \mathbf{R}\mathbf{s} + \boldsymbol{\eta} \quad (4)$$

where $\boldsymbol{\rho} = \mathbf{Q}^H \mathbf{y}$ and $\boldsymbol{\eta} = \mathbf{Q}^H \mathbf{n}$. (4) can be rewritten as

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N_r} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1N_t} \\ 0 & r_{22} & \cdots & r_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{N_t N_t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{N_t} \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_{N_r} \end{pmatrix} \quad (5)$$

The ML algorithm can be expressed as

$$\hat{\mathbf{s}} = \arg \min_{\mathcal{S}} \|\boldsymbol{\rho} - \mathbf{R}\mathbf{s}\|^2 \quad (6)$$

where $\hat{\mathbf{s}} = (\hat{s}_1 \ \hat{s}_2 \ \cdots \ \hat{s}_{N_t})^T$ is the estimated value of transmitted symbol. For sake of simplification, we set $N=N_t$ next.

From equation (5), it can be found that there is no inference from other transmission signals on s_N . It can be expressed as

$$\rho_N = r_{NN} \cdot s_N + \eta_N \quad (7)$$

According to equation (7), s_N can be calculated by

$$\hat{s}_N = \text{round} \left(\frac{\rho_N}{r_{NN}} \right) \quad (8)$$

where \hat{s}_N is the estimated transmit signal. When s_N is known, s_{N-1} can be calculated by eliminated the interference of s_N . According to this idea, all transmit signals can be calculated by

$$\hat{s}_i = \text{round} \left(\frac{1}{r_{ii}} \left(\rho_i - \sum_{k=i+1}^N r_{ik} \cdot \hat{s}_k \right) \right) \quad (9)$$

Equation (9) means the i^{th} receive signal is determined by the transmit signals from the i^{th} to the N^{th} one. So the solution of equation (6) can be searched from the N^{th} signal to the i^{th} signal as layer by layer in a tree. That is the principle of VBLAST algorithm. Each possible transmit signal s_i is called a node in the searching tree. The solution finding can be depicted as searching in the tree from the root node to a certain leaf node.

VBLAST only searches the tree by one route. If there

is an error in one layer, signals in next layers will be error in high probability. That is why the performance of VBLAST algorithm is not good enough. ML algorithm searches all the leaf nodes in the tree and selects the node which satisfies equation (6) from all the possible nodes. Obviously it has the optimal performance, but is a NP-hard problem.

To find the best signal as in equation (6), the distance between transmit and receive symbol should be calculated. According to formula (5), the Euclidean distance (ED) can be defined as

$$D = \left(\sum_{j=1}^N \left(\rho_j - \sum_{k=j}^N r_{jk} \cdot \hat{s}_k \right)^2 \right)^{1/2} \quad (10)$$

Also define the partial Euclidean distance (PED) as

$$D_i = \left(\sum_{j=i}^N \left(\rho_j - \sum_{k=j}^N r_{jk} \cdot \hat{s}_k \right)^2 \right)^{1/2} \quad (11)$$

4. Schnorr-Euchner SD algorithm

Despite the optimal performance, the number of visited nodes of ML algorithm is the exponential function of N_t and m , where m is the signal modulation order. That is a very large number, so the complexity is too high to be utilized. SD algorithm reduces the number of visited nodes by setting a hypersphere before search and only nodes in this hypersphere will be searched. When a new leaf node is found in the hypersphere, the radius of the hypersphere is shrunk to the distance of this node. The complexity of the SD is a small fraction of that of ML.

SESD is the most important SD algorithm. It optimizes the SD algorithm by searching the smallest child node of the parent node each layer first. The first found solution is often the solution good enough. As a result, some nodes visiting can be avoided and the calculation complexity can be further lowered.

5. Pipeline Structure SESD Algorithm

The SESD algorithm must search the whole tree to get the solution. Each visited node must compare with the global smallest radius to determine whether or not it should be kept. Where the SESD algorithm is used, symbols should be detected one by one. Another detection can only begin after one is over. As a result, the average processing time can not drop easily.

In this article, we proposed that the whole search tree can be divided into blocks and the search process can be executed from block to block to form a pipeline structure, like **Figure 1**. Each block is one part in the pipeline. The number of block equals to the number of part in the pipeline. The whole search begins from one certain block.

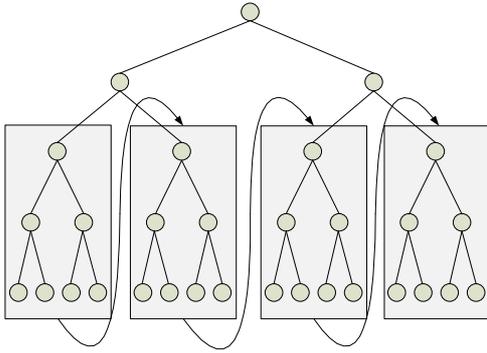


Figure 1. Pipeline structure SEDS algorithm tree.

The top node of blocks is called the sub-root node of this block, from which the block search starts. Unlike the tree root node which has no practical significance, the sub-root node has its counterpart transmit signal and its PED.

There is a solution for each part, which is called the sub-solution of this part. The radius used in the SEDS is still the threshold to cut node in each part. All parts share one radius although it should be updated anytime when the update condition is satisfied. The radius is called globe radius in our algorithm. Each sub-solution will update the globe radius, which will act as the cutting radius in the remaining parts.

Because of the node cutting, there may be no sub-solution for some parts. This is not so much a bad news as a good news because that means all nodes in this part are cut and the visited nodes number in this part is often small. When a sub-solution in one part is found, the globe radius will update to the ED of this sub-solution.

Each part executes independently except one input parameter and two output parameters. One input parameter is the cutting radius, and the output parameters are the new radius and the sub-solution of this part. So it can be processed in a pipeline structure. Each part can process at the same time. If the process of one part is over, it passes the new radius and sub-solution to the next part and begins a new process. There are several processes for decoding in one pipeline at the same time. The average processing time for one detection will drop.

The structure of the proposed algorithm can be depicted as Figure 2(a). Each detection is divided into M parts. One signal begins its step K^{th} detection at the time when the previous signal ends its step K^{th} detection and begins its step $(K+1)^{\text{th}}$ detection. So in a pipeline with M parts, there are M signals detection at the same time, like Figure 2(b).

As mentioned before, if the previously processed block generates a radius small enough, most nodes in following blocks will be cut quickly. The number of visited nodes will be small and the calculation complexity will be low. So we can deduce that the sequence of block

detection should be arranged in a way that the first detected block should be the block that will most likely generate the smallest sub-solution for the whole tree, which is the final solution. So the sequence of block detection should be arranged in the rising order of the PED of the sub-root, that is, just the sequence of the SEDS algorithm.

6. Simulation Results

On the basis of section 5, we show the complexity of our algorithm and the SEDS algorithm. The performance of the proposed algorithm is optimal as the performance of SEDS, so that it is not necessary to be shown. The complexity is represented by the visited node number as in [16].

The complexity of the proposed algorithm is compared with the SEDS algorithm in different SNRs. Also the visited node numbers of every part are listed. We convert the calculation into real field instead of complex field in both figures. But all results are the same as in complex field.

Figure 3 shows the complexity of a 4-by-4 antenna system with 16QAM modulation. Figure 4 shows the complexity of a 4-by-4 antenna system with 64QAM modulation.

From both figures, it can be found that the sum of the visited node number of all parts almost equals to that of the SEDS algorithm, which means that the proposed algorithm does not increase the calculation complexity. The average process time is determined by the part that has the largest visited node number, which is the first part. Specially, in 16QAM system of Figure 3, the average process time is about 48.77%, 54.34% or 60.18% of that of the SEDS algorithm when SNR=0, 10 or 20dB. In

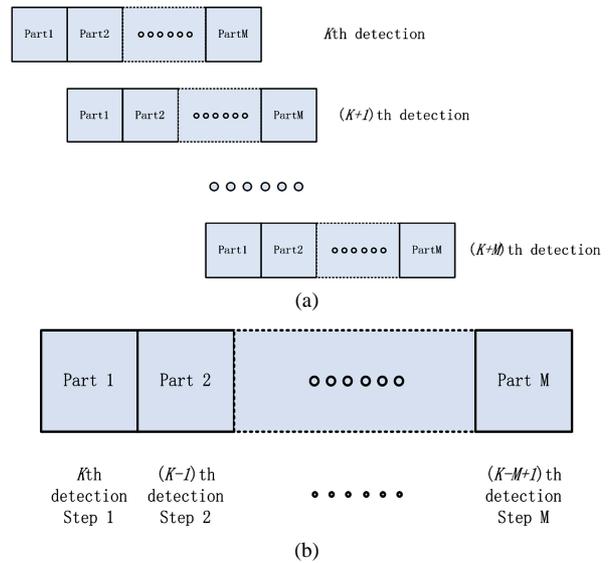


Figure 2. Pipeline structure of SEDS algorithm.

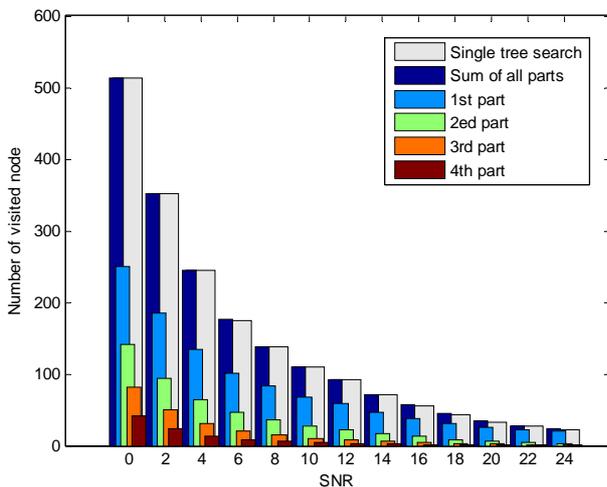


Figure 3. Complexity of 4-by-4 antenna 16QAM system.

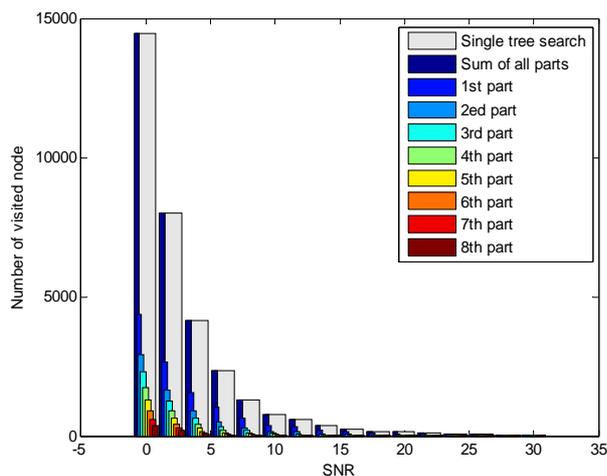


Figure 4. Complexity of 4-by-4 antenna 64QAM system.

64QAM system of **Figure 4**, the average process time is about 30.31%, 52.37% or 61.59% of that of the SESD algorithm when SNR=0, 10 or 20 dB. From these data, we may find that the proportion of the visited node number of the first part increases with SNR. That is because that in high SNR the probability of finding the final solution in the first part is high. When the final solution is found, most nodes in later parts will be cut quickly. As a result, the visited nodes number in later parts is small and the proportion of the first part visited node number is high. This is also indicated that the early visited part should be the part that has high probability to generate the final solution.

7. Conclusions

A novel algorithm is proposed, which divides the search tree of the SESD into blocks so that calculation can process from block to block as a pipeline structure. Several signals can be searched at the same time in one pipe-

line structure. The average processing delay drops to the range from 48.77% to 60.18% in a 4-by-4 antenna system with 16QAM modulation, or from 30.31% to 61.59% in a 4-by-4 antenna system with 64QAM modulation.

8. Acknowledgements

This work is supported by Nature Science Funding of China No. 61071083.

REFERENCES

- [1] A. J. Paulraj, D. A. Gore, R. U. Nabar and H. Bolcskei, "An Overview of MIMO Communications - A Key to Gigabit Wireless," *Proc. Ieee*, Vol. 92, No. 2, pp. 198-218, 2004. [doi:10.1109/JPROC.2003.821915](https://doi.org/10.1109/JPROC.2003.821915)
- [2] A. Goldsmith, S. A. Jafar, N. Jindal and S. Vishwanath, "Capacity Limits of MIMO Channels," *Selected Areas in Communicatio IEEE J.*, Vol. 21, No. 5, 2003, pp. 684-702. [doi:10.1109/JSAC.2003.810294](https://doi.org/10.1109/JSAC.2003.810294)
- [3] P. W. Wolniansky, G. J. Foschini, G. D. Golden and R. A. Valenzuela, "V-BLAST: An Architecture for Realizing very High Data Rates over the Rich-scattering Wireless Channel," in *1998 URSI International Symposium on Signals, Systems, and Electronics*. Conference Proceedings, Pisa, Italy, 1998, pp. 295-300.
- [4] B. Hassibi and H. Vikalo, "On the Sphere-decoding Algorithm I. Expected Complexity," *Signal Process. IEEE Transacions*, Vol. 53, No. 8, 2005, pp. 2806-2818. [doi:10.1109/TSP.2005.850352](https://doi.org/10.1109/TSP.2005.850352)
- [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, 1994, pp. 181-199. [doi:10.1007/BF01581144](https://doi.org/10.1007/BF01581144)
- [6] K. Nikitopoulos and G. Ascheid, "Approximate MIMO Iterative Processing With Adjustable Complexity Requirements," *IEEE Transactions Vehicular Technology*, Vol. 61, No. 2, 2012, pp. 6390-650. [doi:10.1109/TVT.2011.2179324](https://doi.org/10.1109/TVT.2011.2179324)
- [7] X. Mao, S. Ren and H. Xiang, "Adjustable Reduced Metric-First Tree Search," *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing*, 23-25 Sep. 2011, pp. 1-4.
- [8] T. Cui, S. Han and C. Tellambura, "Probability Distribution Based Nodes Pruning for Sphere Decoding," *IEEE Transactions Vehicular Technology*, No. 99, 2012, p. 1.
- [9] J. Ahn, H.-N. Lee and K. Kim, "A Near-ML Decoding with Improved Complexity over Wider Ranges of SNR and System Dimension in MIMO Systems," *Wireless Communications IEEE Transactions*, Vol. 11, No. 1, 2012, pp. 33-37. [doi:10.1109/TWC.2011.110811.110471](https://doi.org/10.1109/TWC.2011.110811.110471)
- [10] X. Dai, R. Zou, J. An, X. Li, S. Sun and Y. Wang, "Reducing the Complexity of Quasi-Maximum-Likelihood Detectors Through Companding for Coded MIMO Systems," *Veh. Technol. IEEE Transactions*, Vol. 61, No. 3, 2012, pp. 1109-1123. [doi:10.1109/TVT.2012.2183008](https://doi.org/10.1109/TVT.2012.2183008)

- [11] B. Shim and I. Kang, "Sphere Decoding With a Probabilistic Tree Pruning," *Signal Process. IEEE Transactions*, Vol. 56, No. 10, 2008, pp. 4867-4878. [doi:10.1109/TSP.2008.923808](https://doi.org/10.1109/TSP.2008.923808)
- [12] X. Y. Mao, Y. X. Cheng, L. L. Ma and H. G. Xiang, "Step Reduced K-best Sphere Decoding," *Proceedings of the 76th IEEE Vehicular Technology Conference*, Quebec City, Canada, 3-6, Sep. 2012, pp. 1-4.
- [13] J. W. Choi, B. Shim and A. C. Singer, "Efficient Soft-Input Soft-Output Tree Detection via an Improved Path Metric," *Inf. Theory Ieee Trans.*, Vol. 58, No. 3, 2012, pp. 1518-1533. [doi:10.1109/TIT.2011.2177590](https://doi.org/10.1109/TIT.2011.2177590)
- [14] K. J. Choi and K. S. Kim, "Hierarchical Tree Search for Runtime-Constrained Soft-Output MIMO Detection," *IEEE Transactions Vehicular Technology*, Vol. 62, No. 2, 2013, pp. 890-896. [doi:10.1109/TVT.2012.2227868](https://doi.org/10.1109/TVT.2012.2227868)
- [15] S.-L. Shieh, R.-D. Chiu, S.-L. Feng and P.-N. Chen, "Low-Complexity Soft-Output Sphere Decoding with Modified Repeated Tree Search Strategy," *IEEE Communications Letters*, Vol. 17, No. 1, 2013, pp. 51-54. [doi:10.1109/LCOMM.2012.112012121728](https://doi.org/10.1109/LCOMM.2012.112012121728)
- [16] A. Schenk, R. F. H. Fischer and L. Lampe, "A Stopping Radius for the Sphere Decoder and Its Application to MSDD of DPSK," *Communications Letters, IEEE*, Vol. 13, No. 7, 2009, pp. 465-467. [doi:10.1109/LCOMM.2009.090940](https://doi.org/10.1109/LCOMM.2009.090940)