國立交通大學

電信工程學系碩士班
碩士論文

用於多輸入多輸出系統具高效率搜尋設計之複數 *K*-Best 球體解碼器

A Complex *K*-Best Sphere Decoder with Efficient Search Design for MIMO Systems

研究生：宋志晟 Student: Chih-Sheng Sung

指導教授：李大嵩 博士 Advisor: Dr. Ta-Sung Lee

中華民國九十八年六月

用於多輸入多輸出系統具高效率搜尋設計之複數 *K*-Best 球體解碼器

A Complex *K*-Best Sphere Decoder with Efficient Search Design for MIMO Systems

研究生：宋志晟　　Student: Chih-Sheng Sung

指導教授：李大嵩 博士　　Advisor: Dr. Ta-Sung Lee

國立交通大學

電信工程學系碩士班

碩士論文

A Thesis
Submitted to Department of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Communication Engineering
June 2009
Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

用於多輸入多輸出系統具高效率搜尋設計之複數 *K*-Best 球體解碼器

學生：宋志晟　　　　　　　　　　指導教授：李大嵩　博士

國立交通大學電信工程學系碩士班

摘要

在無線通訊系統中，多輸入多輸出(Multiple-Input Multiple-Output, MIMO)技術不需要增加額外的頻寬及傳輸功率便能提高傳輸速率及改善傳輸品質。然而，在多輸入多輸出系統中要設計出具高性能且低複雜度之接收機是一項艱難的挑戰。使用最大可能偵測法能得到最佳的效能，然而其所需的運算複雜度會隨著傳送天線個數的增加呈指數的成長。球體解碼演算法能以較低的複雜度達到與最大可能偵測法相同之效能。然而，傳統的球體解碼演算法會有資料吞吐量不穩定之問題。*K*-best 球體解碼演算法在每一層的節點搜尋當中只保留 *K* 個最佳的候選點當作下一次搜尋的依據，因此具有穩定的資料吞吐量。然而，*K*-best 球體解碼演算法需要取相當大的 *K* 值才能達到近似最大可能偵測法之效能。除此之外，在每一層的節點搜尋中將候選點作排序取出 *K* 個最佳的候選點會耗費大量的記憶體存取。在本論文中，吾人提出一具高效率搜尋架構之複數 *K*-best 球體解碼器。此解碼器能夠大幅降低排序時所花費的運算量。針對所提出之複數候選點搜尋方法，吾人亦設計出相對應之電路架構圖。經由分析與模擬的驗證，此解碼器僅需選取較小的 *K* 值即可達到近似最大可能偵測法之效能。

A Complex *K*-Best Sphere Decoder with Efficient Search Design for MIMO Systems

Student: Chih-Sheng Sung Advisor: Dr. Ta-Sung Lee

Department of Communication Engineering

National Chiao Tung University

Abstract

In wireless communication systems, multiple-input and multiple-output (MIMO) technology offers significant increases in data rate and link range without additional bandwidth or transmit power. However, the design of high performance and low complexity receivers for MIMO systems is a challenging task. The maximum-likelihood (ML) detection is the optimal detection scheme but its complexity grows exponentially with the number of transmit antennas. The sphere decoding algorithm (SDA) achieves the ML performance with reduced complexity. Nevertheless, the throughput of the conventional SDA is not stable. The *K*-best SDA which keeps only *K*-best candidates at each layer for the search of next layer is guaranteed to have a stable throughput. However, to achieve a near-ML performance, the value of *K* should be sufficiently large. Besides, applying a sorting algorithm to find *K*-best candidates at each layer requires a large amount of memory access. In this thesis, we propose a complex *K*-best sphere decoder with an efficient search architecture. The proposed *K*-best sphere decoder significantly reduces the sorting complexity. We also provide the hardware architecture of the proposed complex candidate search method. It is demonstrated through analysis and simulations that the proposed *K*-best sphere decoder achieves a near-ML performance without requiring a large value of *K*.

Acknowledgement

I would like to express my deepest gratitude to my advisor, Dr. Ta-Sung Lee, for his enthusiastic guidance and great patience. I learned a lot from his positive attitude in many areas, especially in the training of oral presentation. Besides, I want to thank Chester senior whose knowledge and experience have benefited me tremendously in my research. Thanks are also offered to all members in the Communication System Design and Signal Processing (CSDSP) Lab for their inspiring discussions. Moreover, I sincerely appreciate the encouragement of my friends. I would not be where I am today without their support. At last but not least, I would like to show my sincere thanks to my family for their invaluable love and support.

Contents

List of Figures

List of Tables

Acronym Glossary

CCG	complex candidate generator
CS	compare and select
D-BLAST	diagonal Bell laboratories layered space-time
EGC	equal-gain combining
FP	Fincke and Pohst
MIMO	multiple-input multiple-output
MISO	multiple-input-single-output
ML	maximum-likelihood
MMSE	minimum mean-square error
MRC	maximum ratio combining
SD	spatial diversity
SDA	sphere decoding algorithm
SE	Schnorr and Euchner
SER	symbol error rate
SIC	successive interference cancellation
SISO	single-input-single-output
SIMO	single-input-multiple-output
SM	spatial multiplexing
SNR	signal to noise ratio
STBC	space-time block code
STC	space-time code
STTC	space-time trellis code

V-BLAST	vertical Bell laboratories layered space-time
VLSI	very-large-scale integration
ZF	zero-forcing

Notations

$\lceil\cdot\rceil$	ceiling operator
$(\cdot)^*$	conjugate operator
$E\{\cdot\}$	expectation operator
$\lfloor\cdot\rfloor$	floor operator
$(\cdot)^\dagger$	Moore-Penrose pseudo-inverse operator
$[\cdot]$	rounding operator
$Q(\cdot)$	quantization operator
$(\cdot)^T$	transpose operator
C	channel capacity
$\mathbf{H}$	channel matrix
M	number of receive antennas
M_c	constellation size
N	number of transmit antennas
P	transmit power
$\mathbf{n}$	noise vector
γ	average SNR at the receiver
$\mathbf{x}$	transmit signal vector
$\mathbf{y}$	received signal vector

Chapter 1

Introduction

Next generation wireless communication systems are expected to provide users with higher data rate services including video, audio, data and voice signals. The rapidly growing demand for these services drives the wireless communication technologies towards higher data rate, higher mobility and higher link quality. However, the time-selective and frequency-selective fading in wireless channel caused by multipath propagation, Doppler shifts and carrier frequency/phase drifts severely affect the quality and reliability of wireless communication. Besides, the available bandwidth and power are limited which makes the design of wireless communication systems extremely challenging. Hence, recently there are many innovative techniques that improve the reliability and the spectral efficiency of wireless communication links. Some popular examples include the coded multicarrirer modulation, smart antenna, in particular multiple-input multiple-output (MIMO) technology [1-4] and adaptive modulation [5], [6].

MIMO technology involves the use of multiple antennas at the transmitter and receiver to improve communication performance. The technology offers some benefits that overcome the challenges posed by both the impairments in wireless channel as well as resource constraints. The two important benefits of MIMO technology are the diversity gain and the spatial multiplexing gain. Diversity gain mitigates fading by providing the receiver with multiple (ideally independent) copies of the transmitted signal in space, time or frequency. Spatial multiplexing offers a linear increase in data rate by transmitting multiple independent data streams within the bandwidth of operation.

There are many signal detection schemes for MIMO systems such as linear detection, successive interference cancellation (SIC) [7], [8] and the maximum-likelihood (ML) detection. Both the linear detection and the SIC schemes are easy to be implemented but their detection performance are not optimal. The optimal detection scheme is the ML detection; however, the complexity of the ML detection scheme grows exponentially with the size of the transmit symbol alphabet and the number of transmit antennas. To reduce the complexity of the ML detection, the sphere decoding algorithm (SDA) is introduced in [9-12] to achieve the same performance as the ML detection with reduced complexity. The basic idea of SDA is to search the nearest lattice point to the received signal vector within a given sphere radius. However, the complexity of the conventional SDA is still too high in the low SNR range and its decoding throughput is not stable. Hence, it is not suitable for real time detection and hardware implementations.

To overcome the drawbacks of the conventional SDA, the *K*-best SDA is introduced in [13-16]. The *K*-best SDA uses breadth-first search and keeps *K*-best candidates at each layer for the next layer search. Hence, the decoding throughput of the *K*-best SDA is stable. However, one major drawback of the *K*-best SDA is that the value of *K* has to be chosen sufficiently large to achieve a near-ML performance which increases the computational complexity. Besides, applying sorting algorithm to find *K*-best candidates at each layer causes large amount of memory access.

In this thesis, our major goal is to reduce the sorting complexity of the *K*-best SDA. We propose an efficient sorting method which reduces the large amount of memory access without sacrificing the detection performance. To reduce the number of search layers, we propose an efficient complex domain candidate search method which is simple and can be easily implemented by VLSI process. Moreover, we propose a search method which deals with the poor condition channel to improve the performance of the proposed *K*-best SDA when the value of *K* is small. As a result, the proposed *K*-best SDA has lower complexity and better performance than the conventional *K*-best SDA.

The remainder of the thesis is organized as follows. In Chapter 2, the signal model and the conventional detection schemes of the MIMO systems are introduced first. Secondly, several kinds of the SDA are given. In Chapter 3, the proposed *K*-best SDA is presented. Chapter 4 provides the VLSI architecture of the proposed complex domain candidate search method and the sorting complexity analysis of the proposed *K*-best SDA. Finally, Chapter 5 gives conclusion remarks of this thesis and leads the way to some potential works.

Chapter 2

MIMO Systems

In wireless communication, multiple-input and multiple-output (MIMO) involves the use of multiple antennas at the transmitter and receiver to improve communication performance. MIMO technology offers significant increases in data rate and link range without additional bandwidth or transmit power. In this chapter, we give a review of the MIMO systems. We first introduce the MIMO system model in Section 2.1. Section 2.2 introduces the channel capacity. Then, the spatial diversity (SD) and the spatial multiplexing (SM) techniques are introduced in Section 2.3 and Section 2.4, respectively. The common detection schemes of the MIMO systems will be given in Section 2.5. The sphere decoding algorithm has been studied as a practical solution of the ML detection with reduced complexity. We will give an introduction of the sphere decoding algorithm in Section 2.6.

2.1 System Model

Consider the MIMO system shown in Figure 2-1 with N transmit antennas and M receive antennas. The received signal vector is denoted as $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_M \end{bmatrix}^T \in \mathbb{C}^{M\times 1}$, where y_m is the received signal at the *m*th receive antenna and $[\cdot]^T$ denotes the transpose operator. Similarly, the transmitted signal vector is denoted as $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T \in \mathbb{C}^{N\times 1}$ or $\mathbb{R}^{N\times 1}$, where x_n is the transmitted signal at the *n*th transmit antenna. Assume $M \geq N$ and the channel responses are frequency-flat fading and remain constant during a frame transmission. The channel matrix can be expressed as

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M,1} & h_{M,2} & \cdots & h_{M,N} \end{bmatrix}, \tag{2.1}$$

where $h_{i,j}$ is the channel gain from the *j*th transmit antenna to the *i*th receive antenna. With the assumption that sufficient antenna separation at the transmit and receive antennas, the elements of the channel matrix **H** can be assumed to be i.i.d. complex Gaussian random variables with zero-mean and unit variance. The relation between the received signal vector and the transmitted signal vector can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.2}$$

where $\mathbf{n} = \begin{bmatrix} n_1 & n_2 & \cdots & n_M \end{bmatrix}^T \in \mathbb{C}^{M\times 1}$ is the i.i.d. complex additive white Gaussian noise (AWGN) vector with zero-mean and covariance matrix $\sigma^2\mathbf{I}$.

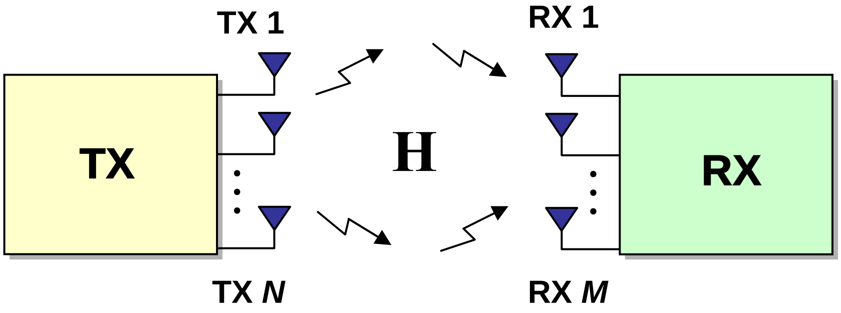


Figure 2-1: Block diagram of a MIMO system

2.2 Channel Capacity

Channel capacity is the highest rate in bits per channel use at which information can be transmitted with an arbitrary probability of error. We first introduce the single-input-single-output (SISO) channel capacity and then study the capacity of a MIMO channel. Note that single-input-multiple-output (SIMO) and multiple-input-single-output (MISO) channels are sub-sets of the MIMO case. The channel capacity is defined as [17]

$$C = \max_{p(x)} \mathrm{I}(X;Y), \tag{2.3}$$

where

$$\mathrm{I}(X;Y) = \mathrm{H}(Y) - H(Y \mid X) \tag{2.4}$$

is the mutual information between X and Y, $\mathrm{H}(Y)$ and $H(Y \mid X)$ are the differential entropy of Y and differential conditional entropy of Y with knowledge of X given, respectively. In (2.3), it states that the mutual information is maximized with respect to all possible transmitter statistical distributions $p(x)$.

The *ergodic capacity* of a SISO system with a random complex channel gain h is given by [17]

$$C = E\left\{\log_2\left(1 + \gamma |h|^2\right)\right\} \quad \text{bits/sec/Hz}, \tag{2.5}$$

where $\gamma = {P}/{\sigma^2}$ is the average SNR at the receiver, P is the transmit power and $E\{\cdot\}$ denotes the expectation over all channel realization. For a MIMO system with N transmit antennas and M receive antennas, the capacity of a random MIMO channel is given by [1]

$$C = \max_{\mathrm{tr}(\mathbf{R}_{xx})=N} E\left\{\log_2\left[\det\left(\mathbf{I}_M + \frac{P}{\sigma^2 N}\mathbf{H}\mathbf{R}_{xx}\mathbf{H}^H\right)\right]\right\} \quad \text{bits/sec/Hz}, \tag{2.6}$$

where $\mathbf{R}_{xx} = E\left\{\mathbf{x}\mathbf{x}^H\right\}$ is the covariance matrix of the transmitted signal vector $\mathbf{x}$. If the channel knowledge is unknown to the transmitter, the signals are chosen to be independent and equal-powered. The covariance matrix of transmit signal vector is then given by $\mathbf{R}_{xx} = \mathbf{I}_M$. As a result, the ergodic capacity of a MIMO system can be written as [1]

$$C = E\left\{\log_2\left[\det\left(\mathbf{I}_M + \frac{P}{\sigma^2 N}\mathbf{H}\mathbf{H}^H\right)\right]\right\} \quad \text{bits/sec/Hz.} \tag{2.7}$$

By using the eigenvalue decomposition, the matrix product of $\mathbf{H}\mathbf{H}^H$ can be decomposed as $\mathbf{H}\mathbf{H}^H = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H$, where $\mathbf{E}$ is an $M \times M$ matrix which consists of the eigenvectors satisfying $\mathbf{E}\mathbf{E}^H = \mathbf{E}^H\mathbf{E} = \mathbf{I}_M$ and $\mathbf{\Lambda} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$ is a diagonal matrix with the eigenvalues $\lambda_i \geq 0$ on the main diagonal. Assuming that the eigenvalues λ_i are ordered so that $\lambda_i \geq \lambda_{i+1}$, we have

$$\lambda_i = \begin{cases} \sigma_i^2, & \text{if } \ 1 \leq i \leq r \\ 0, & \text{if } \ r+1 \leq i \leq M \end{cases}, \tag{2.8}$$

where σ_i^2 is the ith squared singular value of the channel matrix $\mathbf{H}$ and $r = \operatorname{rank}(\mathbf{H}) \leq \min\{N, M\}$ is the rank of the channel matrix. Then the capacity of a MIMO channel can hence be rewritten as

$$\begin{aligned} C &= E\left\{\log_2\left[\det\left(\mathbf{I}_M + \frac{P}{\sigma^2 N}\mathbf{E}\mathbf{\Lambda}\mathbf{E}^H\right)\right]\right\} = E\left\{\log_2\left[\det\left(\mathbf{I}_M + \frac{P}{\sigma^2 N}\mathbf{\Lambda}\right)\right]\right\} \\ &= \sum_{i=1}^{r} E\left\{\log_2\left(1 + \frac{P}{\sigma^2 N}\lambda_i\right)\right\} \quad \text{bits/sec/Hz.} \end{aligned} \tag{2.9}$$

Note that the second equation holds due to the fact $\det(\mathbf{I}_m + \mathbf{A}\mathbf{B}) = \det(\mathbf{I}_n + \mathbf{B}\mathbf{A})$ for matrices $\mathbf{A} \in \mathbb{C}^{m \times n}$ and $\mathbf{B} \in \mathbb{C}^{n \times m}$ and $\mathbf{E}^H\mathbf{E} = \mathbf{I}_M$. Equation (2.9) shows that the capacity of a MIMO channel is made up by sum of the capacities of r SISO sub-channels with power gains λ_i for $i = 1, 2, \ldots, r$ and transmit power P/N individually.

If the channel knowledge is known to the transmitter, the capacity of a MIMO channel is the sum of the capacities associated with the parallel SISO channels and is given by

$$C=\sum_{i=1}^{r}E\left\{\log_2\left(1+\gamma_i\frac{P}{\sigma^2 N}\lambda_i\right)\right\} \quad \text{bits/sec/Hz}, \qquad (2.10)$$

where $\gamma_i = E\left\{\left|x_i\right|^2\right\}$ for $i = 1,2,\ldots,r$ is the transmit power in the *i*th sub-channel and satisfy $\sum_{i=1}^{r}\gamma_i = N$. Since the transmitter can access the spatial sub-channels, we can allocate variable power across the sub-channels to maximize the mutual information. The optimal power allocation of the *i*th sub-channel is given by [1], [17]

$$\gamma_i^{\text{opt}}=\left(\mu-\frac{M\sigma^2}{P\lambda_i}\right)_+ \quad \text{for } i=1,2,\cdots,r\ , \qquad (2.11)$$

where μ is chosen to satisfy the constraint $\sum_{i=1}^{r}\gamma_i^{\text{opt}} = N$ and $(\cdot)_+$ denotes the operation that taking those terms which are positive. The optimal power allocation in (2.11) is found iteratively through the water-filling algorithm [1], [17].

2.3 MIMO Diversity

Diversity techniques are widely used in MIMO systems to improve the reliability of transmission without increasing the transmit power or sacrificing the bandwidth. There many diversity techniques such as time diversity, frequency diversity and space diversity. In this section, we focus on the space diversity that is so called antenna diversity.

2.3.1 Receive Diversity

Receive diversity involves the use of multiple antennas at the receiver. At the receiver, multiple copies of the transmitted signal are received, which can be efficiently combined with an appropriate signal processing algorithm. There are four main types of combining techniques, include selection combining, switch combining, equal-gain combining (EGC) and the maximum ratio combining (MRC). In the selection combining, the received signal with the best quality is chosen and the choosing criterion is based on SNR. Switch diversity switches the received signal path to an alternative antenna when the current received signal level falls below a given threshold. EGC is a simple method since it does not require estimation of the channel. The receiver simply combines the received signals from different receive antennas with weights set to be equal. MRC forms the output signal by a linear combination of all the received signals and is the optimal combination technique which achieves the maximum value of the output SNR.

2.3.2 Transmit Diversity

Transmit diversity techniques which provide diversity benefits at the receiver with multiple transmit antennas, has received much attention, especially in wireless cellular systems. There are two broad categories of transmit diversity: the open loop schemes and the closed loop schemes. In the open loop schemes, the transmitter transmits signals without feedback information from receiver. Space-time code (STC) is an open loop scheme which jointly designs of channel coding and modulation to improve system performance by providing both transmit diversity and coding gain. STC can be classified into two categories, the space-time block code (STBC) and the space-time trellis code (STTC). In this section, we focus on STBC.

The most famous STBC is the Alamouti STBC [3]. This scheme is proposed for two transmit antennas which is shown in Figure 2-2. In the Alamouti scheme, symbols transmitted from the transmit antennas are encoded in both space and time in a simple manner to ensure the transmissions form both the antennas are orthogonal to each other. At a given symbol period, the encoder takes a block of two modulated symbols x_1 and x_2 in each encoding operation and maps them into the transmit antennas according to a code matrix which is given by

$$C = \begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix}, \tag{2.12}$$

where $(\cdot)^*$ denotes the conjugate operator. During the first transmission period, two signals x_1 and x_2 are transmitted simultaneously from antenna one and antenna two respectively. In the second transmission period, $-x_2^*$ is transmitted from the transmit antenna one and x_1^* is transmitted form the transmit antenna two. The Alamouti scheme extracts the diversity order of 2 (full transmit diversity) even in the absence of the channel knowledge at the transmitter.

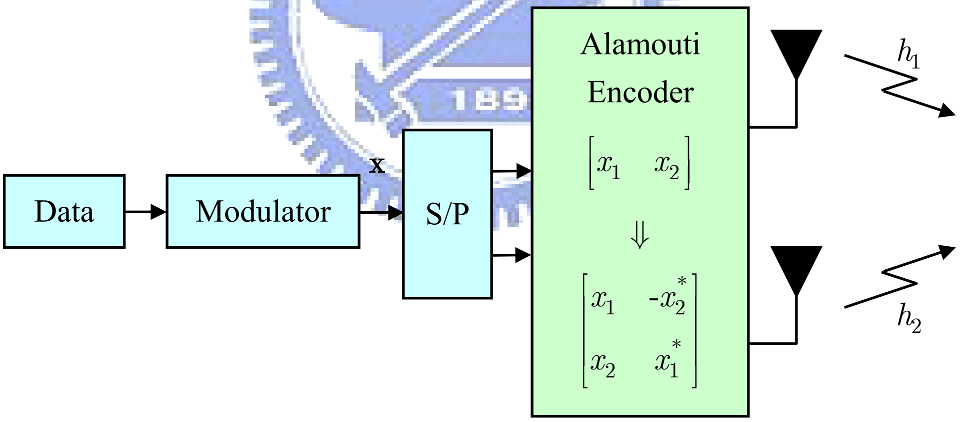

Data
Modulator
x
S/P
Alamouti
Encoder
$\begin{bmatrix} x_1 & x_2 \end{bmatrix}$
⇓
$\begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix}$
h_1
h_2

Figure 2-2: Transmitter block of the Alamouti scheme

2.4 Spatial Multiplexing

Spatial multiplexing is a transmission technique of MIMO wireless communication systems which increases the transmission data rate without additional bandwidth or power consumption. In the spatial multiplexing systems, N different data streams are transmitted from different transmit antennas simultaneously or sequentially and these data streams are separated and demutiplexed to yield the original transmitted signals according to their unique spatial signatures at the receiver, as illustrated in Figure 2-3. The separation of data streams at the receiver can be done possibly by the fact that rich scattering multi-paths contribute to lower correlations between MIMO channel coefficients and hence create a channel matrix with full rank and low condition number to N unknowns from a linear system of M equations. In the following, two typical spatial multiplexing schemes, D-BLAST [4] and V-BLAST [18] are introduced.

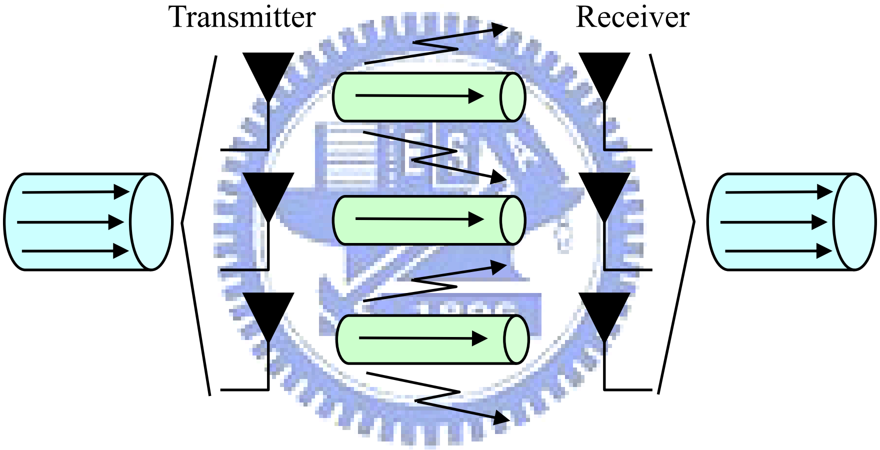


Figure 2-3: An illustration of a spatial multiplexing system

(1) Diagonal Bell Laboratories Layered Space-Time (D-BLAST)

The concept of layered space-time processing was introduced by Foschini at Bell Laboratories [4]. D-BLAST uses multiple antennas at both the transmitter and the receiver, and an elegant diagonally-layered coding sequence in which code blocks are dispersed across the diagonals in space-time. The high-rate information bit stream is first demultiplexed into N substreams, and each substream is encoded by a conventional 1-D constituent code. The encoders apply these coded symbols to the input to form a semi-infinite matrix $\mathbf{X}$ of N rows to be transmitted. The encoding procedure is shown in Figure 2-4.

(2) Vertical Bell Laboratories Layered Space-Time (V-BLAST)

The D-BLAST algorithm suffers from certain implementation complexities which is not suitable for practical implementation. Therefore, a simplified version of the BLAST algorithm is known as V-BLAST. It is capable of achieving high spectral efficiency while being relatively simple to be implemented. The coding procedure of the V-BLAST can be viewed as there is an encoder on each transmit antenna. The output coded symbols of each encoder are transmitted directly from the corresponding antenna which is shown in Figure 2-5.

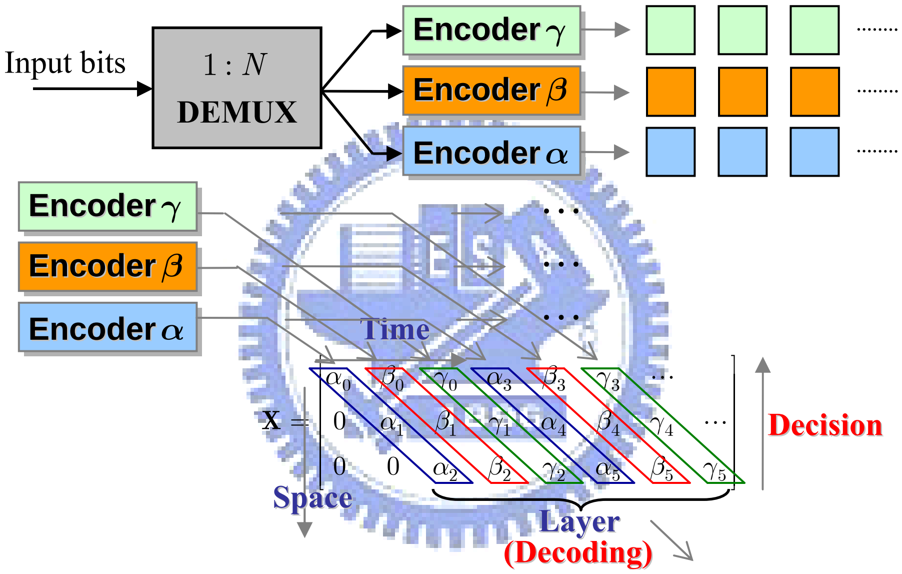


Figure 2-4: Encoding procedure of the D-BLAST scheme $(N = 3)$

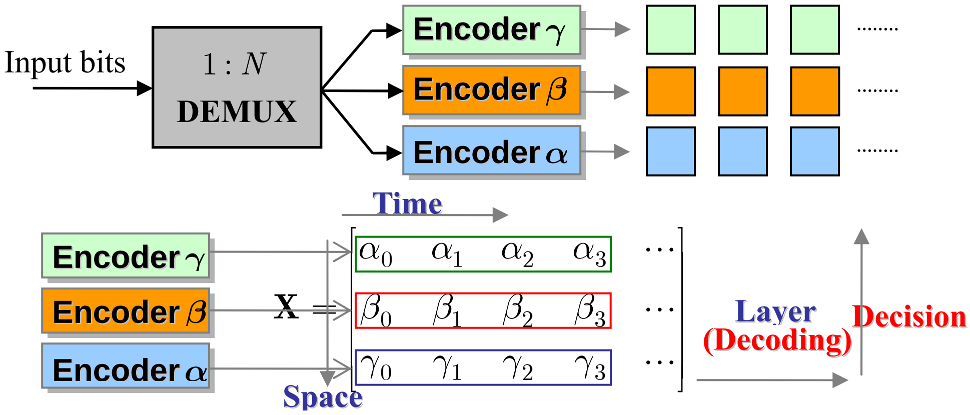


Figure 2-5: Encoding procedure of the V-BLAST scheme $(N = 3)$

2.5 MIMO Detection

The classification of MIMO detection schemes are shown in Figure 2-6. In this section, we introduce linear and non-linear detection schemes in the following subsections.

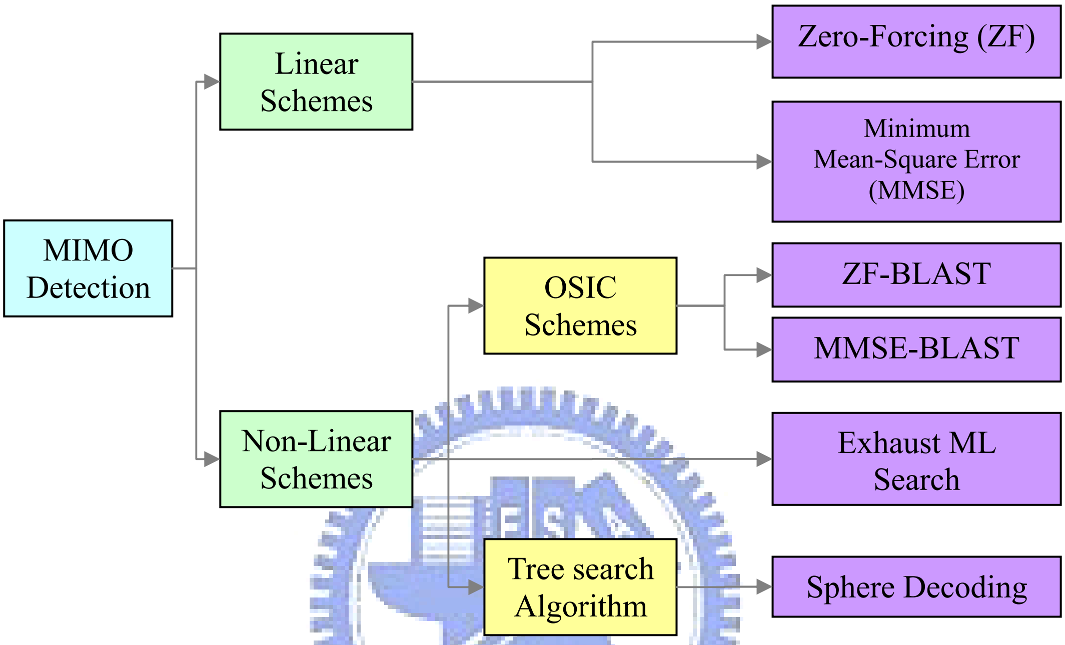


Figure 2-6: Overview of MIMO detection methods

2.5.1 Linear Detection

The linear detection is to preprocess the received signal by transforming it linearly

$$\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y} = \mathbf{W}\mathbf{H}\mathbf{x} + \mathbf{W}\mathbf{n} \tag{2.13}$$

so that the transformed channel-matrix $\mathbf{WH}$ will be close to a diagonal matrix. Here, we introduce two detection schemes: zero-forcing and minimum mean-square error.

(1) Zero-Forcing (ZF)

In the ZF scheme, the preprocessing matrix $\mathbf{W}$ is chosen to remove the off-diagonal terms of $\mathbf{WH}$

$$\tilde{\mathbf{y}} = \mathbf{H}^{\dagger}\mathbf{y} \tag{2.14}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{H}$. The spatial interferences are completely removed from the received signal; however, the main drawback of ZF method is the resulting noise enhancement.

(2)Minimum Mean-Square Error (MMSE)

The MMSE scheme minimizes the joint effects of the interferences and the noise by

$$\tilde{\mathbf{y}} = \left(\mathbf{H}^{H}\mathbf{H} + \frac{\sigma^{2}}{E_{s}}\mathbf{I}\right)^{-1}\mathbf{H}^{H}\mathbf{y} \tag{2.15}$$

where E_s is the average energy of a transmitted symbol. MMSE outperforms than ZF method; however, to estimate the variance of the noise is hard at the receiver.

2.5.2 Non-Linear Detection

(1) V-BLAST Detection

The detection scheme of the V-BLAST system is based on the ordered successive interference cancellation (OSIC) algorithm [7], [8]. In OSIC, the detection procedure first detects the strongest signal, cancels the effect of the detected signal from the received signal, and then proceeds to detect the strongest signal of the remaining transmitted signals, and so on. The basic steps of the OSIC algorithm are as follows:

- *Ordering*: Detect the signals in descending order of power in accordance with some criteria such as ZF or MMSE.
- *Nulling*: Nulling out all weaker signals to extract the strongest signal.
- *Slicing*: Make a hard decision on the strongest signal.
- *Cancelling*: Cancel the effect of the detected strongest signal from the received signal.

There are two common V-BLAST detection schemes: ZF V-BLAST and MMSE

V-BLAST. The differences between these two detection schemes are the ordering criterion and the nulling process. The ZF V-BLAST detection algorithm can be summarized as follows:

Initialization:

$$\mathbf{y}_1 = \mathbf{y},\ \ \mathbf{G}_1 = \mathbf{H}^{\dagger},\ \ n = 1 \tag{2.16}$$

Recursion:

$$o_n = \arg \min_{j \notin \{o_1, o_2, \cdots, o_{n-1}\}} \left\| \left(\mathbf{G}_n\right)_j \right\|^2 \tag{2.17}$$

$$\mathbf{w}_{o_n} = \left(\mathbf{G}_n\right)_{o_n} \tag{2.18}$$

$$z_{o_n} = \mathbf{w}_{o_n}^H \mathbf{y}_n \tag{2.19}$$

$$\hat{x}_{o_n} = Q\left(z_{o_n}\right) \tag{2.20}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{H}_{o_n} \hat{x}_{o_n} \tag{2.21}$$

$$\mathbf{G}_{n+1} = \left(\widetilde{\mathbf{H}}_{o_n}^H\right)^{\dagger} \tag{2.22}$$

$$n = n + 1 \tag{2.23}$$

The MMSE V-BLAST detection algorithm can be summarized as follows:

Initialization:

$$\mathbf{y}_1 = \mathbf{y},\ \ \mathbf{G}_1 = \left(\mathbf{H}^H \mathbf{H} + \frac{\sigma^2}{P}\mathbf{I}\right)^{-1} \mathbf{H}^H = \mathbf{Q}_1 \mathbf{H}^H,\ \ n = 1 \tag{2.24}$$

Recursion:

$$o_n = \arg \min_{j \notin \{o_1, o_2, \cdots, o_{n-1}\}} \left\| \left(\mathrm{diag}\left(\mathbf{Q}_n\right)\right)_j \right\|^2 \tag{2.25}$$

$$\mathbf{w}_{o_n} = \left(\mathbf{G}_n\right)_{o_n} \tag{2.26}$$

$$z_{o_n} = \mathbf{w}_{o_n}^H \mathbf{y}_n \tag{2.27}$$

$$\hat{x}_{o_n} = Q\left(z_{o_n}\right) \tag{2.28}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{H}_{o_n} \hat{x}_{o_n} \tag{2.29}$$

$$\mathbf{G}_{n+1} = \left(\widetilde{\mathbf{H}}_{o_n}^H \widetilde{\mathbf{H}}_{o_n} + \frac{\sigma^2}{P} \mathbf{I} \right)^{-1} \widetilde{\mathbf{H}}_{o_n}^H = \mathbf{Q}_{n+1} \widetilde{\mathbf{H}}_{o_n}^H \tag{2.30}$$

$$n = n + 1 \tag{2.31}$$

where $(\mathbf{A})_j$ denotes the *j*th column of matrix **A**, $Q(\cdot)$ denotes the quantization operator, $\widetilde{\mathbf{A}}_{o_n}$ denotes the matrix obtained by removing columns $o_1, o_2, \cdots, o_n$ of **H**, and $(\mathrm{diag}(\mathbf{A}))_j$ denotes the *j*th diagonal element of matrix **A**.

(2) Maximum-Likelihood (ML) Detection

It is well known that the optimal detection scheme of the MIMO systems is the ML detection. The ML detection searches all possible combinations of transmitted symbols via the following criterion:

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\min_{\mathbf{x} \in S} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{2.32}$$

where $S = \mathcal{O}^N$ denotes the set of all possible transmitted symbol vectors and $\mathcal{O}$ is the modulation symbol alphabet. Note that $|S| = |\mathcal{O}|^N$ so that the computational complexity grows exponentially with N. Therefore, it is difficult to be implemented at the receiver in practice which is the main drawback of this method.

2.6 Sphere Decoding Algorithm (SDA)

The sphere decoding algorithm (SDA) [9-12] achieves ML performance with reduced complexity. Hence, it has recently received considerable attentions as an effective detection scheme for MIMO systems. The basic idea of SDA is to restrict the search region of the optimal solution to a smaller subset. Typically, the search region is constrained to the interior of a hyper-sphere of radius d centered at the received signal vector $\mathbf{y}$ as

$$d^2 \geq \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 . \tag{2.33}$$

The geometrical interpretation of the concept of SDA is shown in Figure 2-7. Generally, the complex signal will be transformed into real signal model by

$$\begin{aligned} \tilde{\mathbf{y}} &= \begin{bmatrix} \operatorname{Re}(\mathbf{y}) \\ \operatorname{Im}(\mathbf{y}) \end{bmatrix}, \quad \tilde{\mathbf{x}} = \begin{bmatrix} \operatorname{Re}(\mathbf{x}) \\ \operatorname{Im}(\mathbf{x}) \end{bmatrix}, \quad \tilde{\mathbf{n}} = \begin{bmatrix} \operatorname{Re}(\mathbf{n}) \\ \operatorname{Im}(\mathbf{n}) \end{bmatrix}, \\ \widetilde{\mathbf{H}} &= \begin{bmatrix} \operatorname{Re}(\mathbf{H}) & -\operatorname{Im}(\mathbf{H}) \\ \operatorname{Im}(\mathbf{H}) & \operatorname{Re}(\mathbf{H}) \end{bmatrix}, \end{aligned} \tag{2.34}$$

where $\operatorname{Re}(\cdot)$ and $\operatorname{Im}(\cdot)$ are the real and imaginary parts of its argument. Hence, the real-valued signal is

$$\tilde{\mathbf{y}} = \widetilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \tag{2.35}$$

where $\tilde{\mathbf{y}} \in \mathbb{R}^{2M\times 1}$, $\tilde{\mathbf{x}} \in \Lambda \subset \mathbb{Z}^{2N\times 1}$, $\tilde{\mathbf{n}} \in \mathbb{R}^{2M\times 1}$, and $\widetilde{\mathbf{H}} \in \mathbb{R}^{2M\times 2N}$. Performing QR-decomposition of $\widetilde{\mathbf{H}}$, we have

$$\widetilde{\mathbf{H}} = \begin{bmatrix} \widetilde{\mathbf{Q}}_1 & \widetilde{\mathbf{Q}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0} \end{bmatrix}, \tag{2.36}$$

where $\widetilde{\mathbf{R}}$ is a $2N \times 2N$ upper triangular matrix, $\mathbf{0}$ is a $2(M-N)\times N$ zero matrix, and $\widetilde{\mathbf{Q}}_1$ and $\widetilde{\mathbf{Q}}_2$ are $2M \times 2N$ and $2M \times 2(M-N)$ unitary matrices respectively.

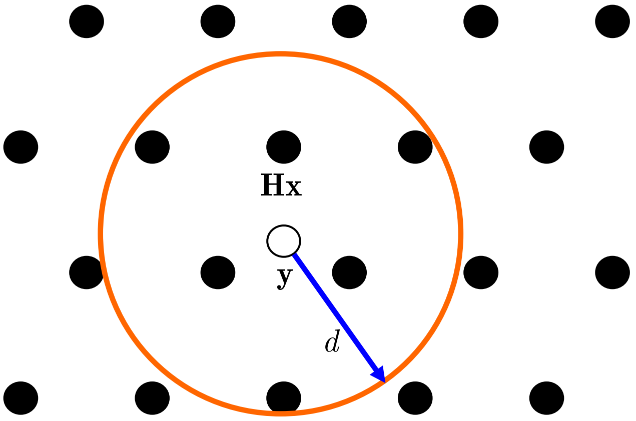


Figure 2-7: Geometrical interpretation of the SDA

Substituting (2.36) into (2.33), we have

$$\left(d^{'}\right)^2 \geq \left\|\tilde{\mathbf{y}}^{'} - \tilde{\mathbf{R}}\tilde{\mathbf{x}}\right\|^2, \tag{2.37}$$

where $\tilde{\mathbf{y}}^{'} = \tilde{\mathbf{Q}}_1^H \tilde{\mathbf{y}}$ and $\left(d^{'}\right)^2 = d^2 - \left\|\tilde{\mathbf{Q}}_2^H \tilde{\mathbf{y}}\right\|^2$. Letting $\tilde{\mathbf{R}} = \left[\tilde{r}_{i,j}\right]$ with $\tilde{r}_{i,i} > 0$ and $\tilde{r}_{i,j} = 0$ for $i > j$, (2.37) can be expended as

$$\begin{aligned}\left(d^{'}\right)^2 &\geq \left\|\tilde{\mathbf{y}}^{'} - \tilde{\mathbf{R}}\tilde{\mathbf{x}}\right\|^2 \\ &= \sum_{i=1}^{2N}\left(\tilde{y}_i - \sum_{j=i}^{2N}\tilde{r}_{i,j}\tilde{x}_j\right)^2 \\ &= \left(\tilde{y}^{'}_{2N} - \tilde{r}_{2N,2N}\tilde{x}_{2N}\right)^2 + \left(\tilde{y}^{'}_{2N\text{-}1} - r_{2N\text{-}1,2N}\tilde{x}_{2N} - \tilde{r}_{2N\text{-}1,2N\text{-}1}\tilde{x}_{2N\text{-}1}\right)^2 + \cdots \\ &= \left(\tilde{y}^{''}_{2N} - \tilde{r}_{2N,2N}\tilde{x}_{2N}\right)^2 + \left(\tilde{y}^{''}_{2N\text{-}1} - \tilde{r}_{2N\text{-}1,2N\text{-}1}\tilde{x}_{2N\text{-}1}\right)^2 + \cdots,\end{aligned} \tag{2.38}$$

where

$$\tilde{y}^{''}_i = \tilde{y}^{'}_i - \sum_{j=i+1}^{2N}\tilde{r}_{i,j}\tilde{x}_i. \tag{2.39}$$

The iterative search for the candidates $\tilde{x}_{2N}, \tilde{x}_{2N\text{-}1}, \ldots, \tilde{x}_2, \tilde{x}_1$ creates a search tree as shown in Figure 2-8.

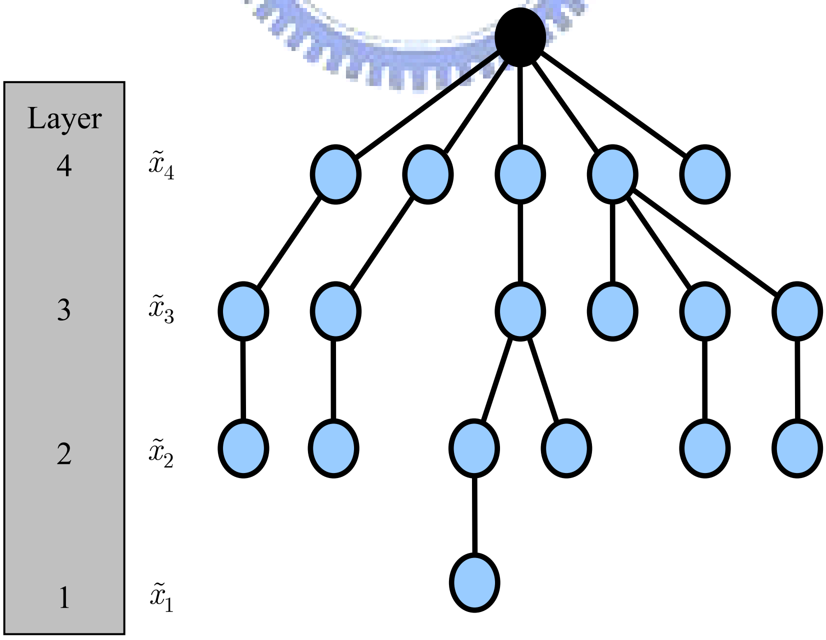


Figure 2-8: A tree search structure of the SDA $(N = 2)$

2.6.1 Fincke and Pohst SDA

In the Fincke and Pohst SDA (FP-SDA) [9], [10], the radius is chosen to be a scaled variance of the noise

$$d^2 = \alpha N \sigma^2, \tag{2.40}$$

where the scalar α is chosen in such a way that the a lattice point $\tilde{\mathbf{H}}\tilde{\mathbf{x}}$ lies in the sphere with probability P_s

$$\int_0^{\alpha} \frac{\lambda^{N-1}}{\Gamma(N)} e^{-\lambda} d\lambda = P_s, \tag{2.41}$$

where the integrand is the pdf of the χ^2 random variable with N degrees of freedom, and P_s is set to a value closed to 1. If no lattice point is not found, we can increase P_s, adjust d, and then search again. For the (2*N*)th layer, the necessary condition for $\tilde{\mathbf{H}}\tilde{\mathbf{x}}$ to lie inside the sphere is

$$\left(d'\right)^2 \geq \left(\tilde{y}''_{2N} - \tilde{r}_{2N,2N}\tilde{x}_{2N}\right)^2. \tag{2.42}$$

Hence, the corresponding range of $\tilde{x}_{2N}$ is

$$\left\lceil \frac{\tilde{y}''_{2N} - d'}{\tilde{r}_{2N,2N}} \right\rceil \leq \tilde{x}_{2N} \leq \left\lfloor \frac{\tilde{y}''_{2N} + d'}{\tilde{r}_{2N,2N}} \right\rfloor, \tag{2.43}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and the floor operations respectively. In general, the necessary condition of the *k*th layer is

$$\left(\tilde{y}''_k - \tilde{r}_{k,k}\tilde{x}_k\right) \leq \left(d'\right)^2 - \sum_{j=k+1}^{2N} \left(\tilde{y}''_j - \tilde{r}_{j,j}\tilde{x}_j\right)^2 = \left(d'_k\right)^2. \tag{2.44}$$

Hence, the corresponding range of $\tilde{x}_k$ leads to

$$LB(\tilde{x}_k) \leq \tilde{x}_k \leq UB(\tilde{x}_k), \tag{2.45}$$

where

$$LB(\tilde{x}_k) = \left\lceil \frac{\tilde{y}''_k - d'_k}{\tilde{r}_{k,k}} \right\rceil \tag{2.46}$$

$$UB\left(\tilde{x}_k\right) = \left\lfloor \frac{\tilde{y}_k^{''} + d_k^{'}}{\tilde{r}_{k,k}} \right\rfloor \tag{2.47}$$

The lattice points satisfying the above equation are searched to find the ML solution. In FP-SDA, the search for $\tilde{x}_k$ is performed in the order of $LB(\tilde{x}_k),\ LB\left(\tilde{x}_k\right)+\delta,\ LB\left(\tilde{x}_k\right)+2\delta,\ \cdots, UB\left(\tilde{x}_k\right)$, where δ is the minimum distance between symbols. If no lattice point satisfying (2.45) is found for all k, the algorithm will set a larger search radius and restart the search again.

2.6.2 Schnorr and Euchner SDA

Schnorr and Euchner SDA (SE-SDA) [12] is a variant of the FP-SDA. In SE-SDA, the search for the candidate $\tilde{x}_k$ of the *k*th layer is started from

$$\hat{x}_k = \left[\frac{\tilde{y}_k^{''}}{\tilde{r}_{k,k}}\right], \tag{2.48}$$

where $[\cdot]$ denotes the rounding operator and the search order is $\hat{x}_k,\ \hat{x}_k - \delta,\ \hat{x}_k + \delta, \cdots$. Hence, the first solution found by the SE-SDA is exactly the Babai estimate [19] (zero-forcing solution) of the real-valued signal. Therefore, the search radius is set to be

$$d_{SE} = \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}_{ZF} \right\|, \tag{2.49}$$

where $\tilde{\mathbf{x}}_{ZF}$ denotes the zero-forcing solution. Note that $\tilde{\mathbf{H}}\tilde{\mathbf{x}}_{ZF}$ is a lattice point; hence it guarantees that there is at least one lattice point inside the sphere which does not have to restart the search again as that in the FP-SDA. Moreover, the searching symbols in each layer of this algorithm starts from the point that is the nearest one to $\tilde{y}_k^{''}$ for each *k*; hence, it is likely to find the optimal solution earlier than the FP-SDA which saves the computational complexity.

2.6.3 *K*-Best SDA

Both FP-SDA and SE-SDA achieve ML performance; however, their complexity in the low-SNR range is still too high. Besides, the decoding throughput is not stable and hence they are not suitable for real-time hardware implementation. The *K*-best SDA [13-16] is proposed for the MIMO detections with lower complexity and a stable throughput. First, define path weight $\tilde{P}_k$ and branch weight $\tilde{B}_k$ of the *k*th layer as

$$\begin{cases} \tilde{P}_k = 0, & \text{for } k = 2N+1 \\ \tilde{P}_k = \tilde{P}_{k+1} + \tilde{B}_k, & \text{for } 1 \le k \le 2N \end{cases}, \tag{2.50}$$

$$\text{where } \tilde{B}_k = \left(\tilde{y}_k'' - \tilde{r}_{k,k}\tilde{x}_k\right)^2. \tag{2.51}$$

The main idea of the *K*-best SDA is to keep only K candidates which have the smallest path weights in each layer which is shown in Figure 2-9. The steps of the *K*-best SDA [13] are summarized as follows:

Step 1. Calculate $C = \left\|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}_{ZF}\right\|^2$

Step 2.

(a). Set $k = 2N$, $\tilde{T}_{2N} = C$. For each symbol in the constellation, calculate:

$$D = \tilde{T}_{2N} - \tilde{B}_{2N}. \tag{2.52}$$

$$\tilde{T}_{2N-1} = \begin{cases} D \text{ , if } D > 0 \\ 0 \text{ , otherwise} \end{cases}. \tag{2.53}$$

(b). Choose those symbols which have the K largest positive $\tilde{T}_{2N-1}$ values.

Step 3.

(a) $k = k - 1$

(b) For each survival partial symbol vector from the previous layer; for each symbol in the constellation, calculate:

$$D = \tilde{T}_k - \tilde{B}_k. \tag{2.54}$$

$$\tilde{T}_{k-1} = \begin{cases} D \ , \ \text{if } D > 0 \\ 0 \ , \ \text{otherwise} \end{cases} . \tag{2.55}$$

(c) Choose those symbols which have the K largest positive $\tilde{T}_{k-1}$ values.

(d) If $(k = 1)$

The solution is the symbol vector with the largest $\tilde{T}_0$.

else

Go back to step 3(a)

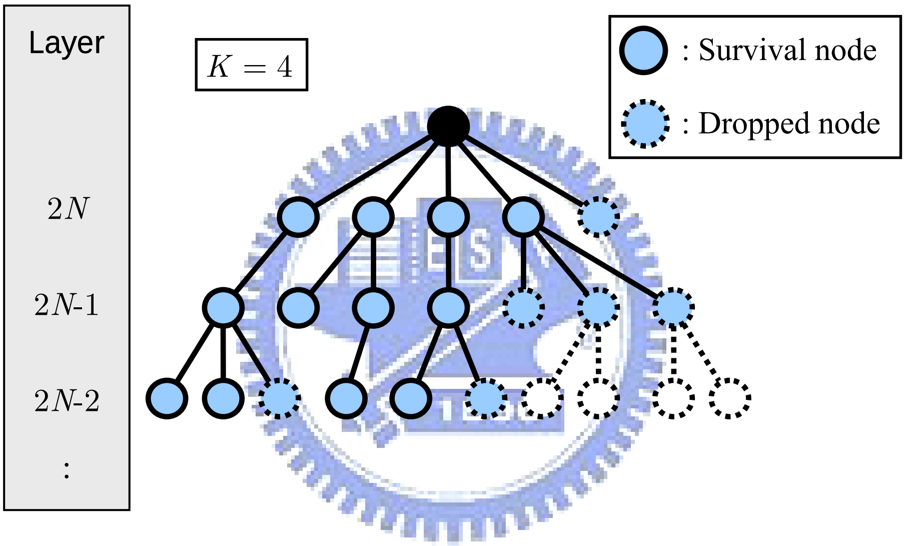


Figure 2-9: The concept of the *K*-best SDA

2.6 Summary

In this chapter, we give a review of the MIMO communication systems. Exploiting multi-path scattering, MIMO systems deliver significant performance enhancements in terms of data rate and link quality. Spatial diversity is one of the MIMO techniques which mitigates fading and is realized by providing the receiver with multiple copies of the transmitted signal in space or time. MIMO systems offer a linear increase in data rate through spatial multiplexing by transmitting multiple and independent data streams without requiring additional bandwidth or transmit power.

The detection of MIMO signals can be classified into two types: linear detection and non-linear detection. Linear detection schemes such as ZF or MMSE are simple but their performances are poor. The optimal detection scheme is the ML detection. However, the computational complexity grows exponentially with the number of transmit antennas. The sphere decoding algorithm achieves the ML performance with reduced complexity. In this chapter, we introduce several kinds of the sphere decoding algorithms for the trade-off between performance and the computational complexity.

Chapter 3

Proposed Complex *K*-Best Sphere Decoding Algorithm

In Chapter 2, we introduce the *K*-best SDA which is a suitable solution for hardware implementation; however, the conventional *K*-best SDA has some disadvantages. First, the complex-valued signal should be converted into real-valued signal which increases the search layers. Besides, performing a sorting algorithm to find K nodes which have the smallest path weights in each layer causes a large amount of memory access. Moreover, to achieve a near-ML performance, the value of K should be chosen sufficiently large, and this increases the computational complexity.

In this chapter, we introduce the proposed complex *K*-best SDA which requires a lower complexity and achieves a near-ML performance with smaller K than the conventional *K*-best SDA. The details of the proposed algorithm will be introduced through Section 3.1 to Section 3.5. The simulation results will be provided in Section 3.6 to show that the proposed algorithm works better than the conventional *K*-best SDA.

3.1 Complex *K*-Best SDA

The complex *K*-best SDA is similar to the conventional *K*-best SDA. First, performing complex QR-decomposition to the channel matrix $\mathbf{H}$, we obtain

$$\mathbf{H} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{3.1}$$

where $\mathbf{Q}_1 \in \mathbb{C}^{M \times N}$ and $\mathbf{Q}_2 \in \mathbb{C}^{M \times (M-N)}$ are unitary matrices, $\mathbf{R}$ is an $N \times N$ upper triangular matrix and $\mathbf{0}$ is an $(M-N) \times N$ zero matrix. Substituting (3.1) into (2.33), we have

$$\left(d'\right)^2 \geq \left\| \mathbf{y}' - \mathbf{R}\mathbf{x} \right\|^2 \tag{3.2}$$

where $\mathbf{y}' = \mathbf{Q}_1^H \mathbf{y}$ and $\left(d'\right)^2 = d^2 - \left\| \mathbf{Q}_2^H \mathbf{y} \right\|^2$. In this thesis, we choose $d = \left\| \mathbf{y} - \mathbf{H}\mathbf{x}_{ZF} \right\|$ and $\mathbf{x}_{ZF}$ denotes the zero-forcing solution. The right-hand-side of (3.2) can be expanded as

$$\begin{aligned} \left(d'\right)^2 &\geq \left\| \mathbf{y}' - \mathbf{R}\mathbf{x} \right\|^2 \\ &= \sum_{i=1}^{N} \left\| y_i' - \sum_{j=i}^{N} r_{i,j} x_j \right\|^2 \\ &= \left\| y_N' - r_{N,N} x_N \right\|^2 + \left\| y_{N\text{-}1}' - r_{N\text{-}1,N} x_N - r_{N\text{-}1,N\text{-}1} x_{N\text{-}1} \right\|^2 + \cdots \\ &= \left\| y_N'' - r_{N,N} x_N \right\|^2 + \left\| y_{N\text{-}1}'' - r_{N\text{-}1,N\text{-}1} x_{N\text{-}1} \right\|^2 + \cdots, \end{aligned} \tag{3.3}$$

where $y_i'' = y_i' - \sum_{j=i+1}^{N} r_{i,j} x_j$. We define the path weight P_k and the branch weight B_k of the *k*th layer as

$$\begin{cases} P_k = 0, & \text{for } k = N+1 \\ P_k = P_{k+1} + B_k, & \text{for } 1 \leq k \leq N \end{cases} \tag{3.4}$$

$$B_k = \left\| y_k'' - r_{k,k} x_k \right\|^2 \tag{3.5}$$

For the *i*th layer, the complex *K*-best SDA keeps K survival nodes with the minimum P_i s. Next, for the (*i*+1)th layer, the algorithm finds all possible nodes which are extended from the previous K survival nodes at the *i*th layer and then again keeps K survival nodes with the minimum P_{i+1} s. The procedures are recursively executed and stoped at the 1st layer. The solution is the symbol vector with the minimum P_1. We summarize the steps of the complex *K*-best SDA as follows.

Step 1. Calculate $\mathbf{y}^{'}$, $d^{'}$

Step 2.

(a). Set $k = N$. For each symbol in the complex-domain constellation, calculate:

$$P_N = P_{N+1} + B_N, \tag{3.6}$$

$$S_N = \begin{cases} P_N, & \text{if } P_N \le \left(d^{'}\right)^2 \\ \text{-1}, & \text{if } P_N > \left(d^{'}\right)^2 \end{cases}. \tag{3.7}$$

(b). Choose those symbols which have the *K* smallest non-negative S_N s.

Step 3.

(a). $k \leftarrow k-1$

(b). For each survival partial symbol vector from the previous layer; for each symbol in the complex-domain constellation, calculate:

$$P_k = S_{k+1} + B_k, \tag{3.8}$$

$$S_k = \begin{cases} P_k, & \text{if } P_k \le \left(d^{'}\right)^2 \\ \text{-1}, & \text{if } P_k > \left(d^{'}\right)^2 \end{cases}. \tag{3.9}$$

(c). Choose those symbols which have the *K* smallest non-negative S_k s.

Step 4.

If $k = 1$

The solution is the symbol vector with positive and the smallest S_1.

Else

Go back to Step 3.

3.2 Efficient Sorting Strategy

The main idea of the *K*-best SDA is to keep only K survival nodes in each layer. The conventional *K*-best SDA has to sort among all possible nodes in each layer to find K survival nodes which is shown in Figure 3-1. Note that the number inside the circle denotes the path weight of the node. Generally, the sorting algorithm applied in the *K*-best SDA is the bubble sort algorithm [13], [20]. When the size of the symbol alphabet of the transmitted symbol or the value of K is large, using bubble sort to sort among all possible nodes will cause a large amount of memory access.

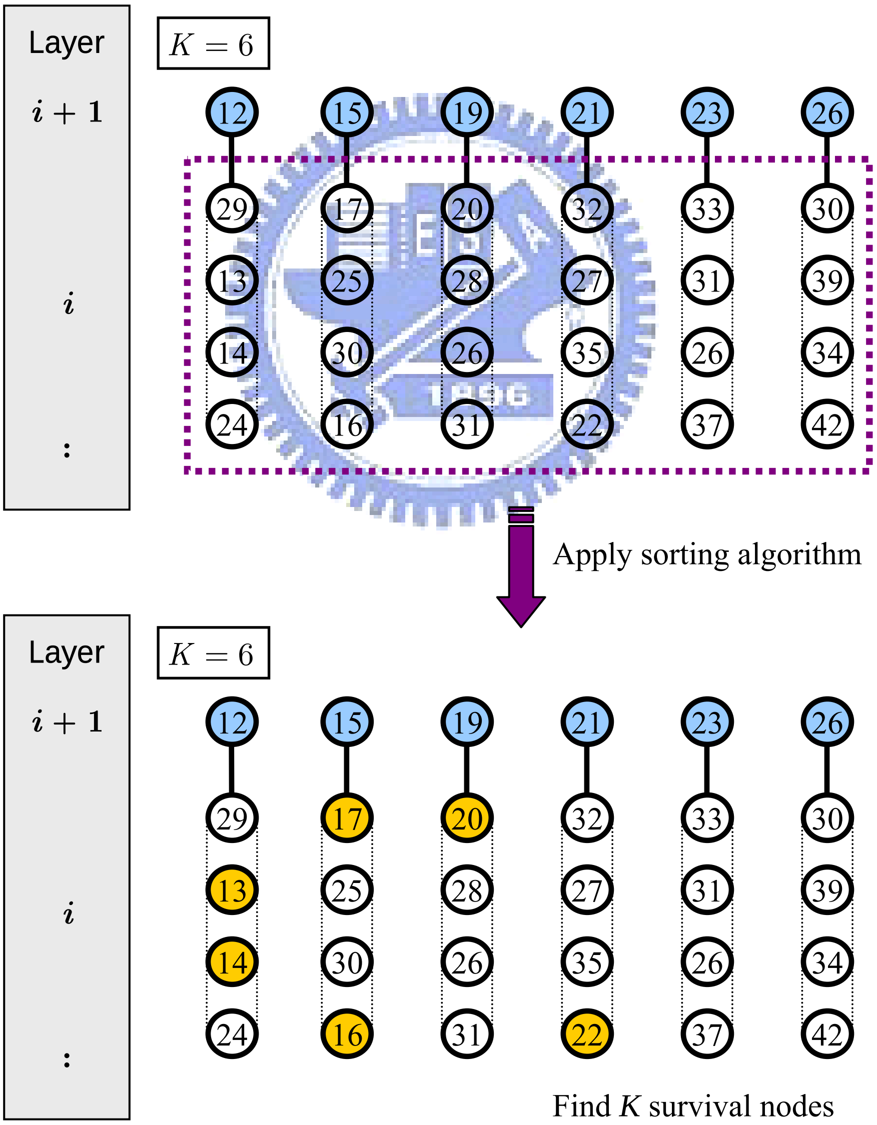


Figure 3-1: Sorting of the conventional *K*-best SDA

In the proposed method, the child nodes of each parent node in the *i*th layer are first sorted in an ascending order according to their path weights. We propose an efficient method to obtain the sorted child nodes sequence of each parent node which does not require any complexity in sorting. The details of constructing the sorted child node sequences will be introduced in the next section. Since the child nodes of each parent node are already sorted, the candidate nodes in the *i*th layer are automatically divided into K sorted groups which is shown in Figure 3-2.

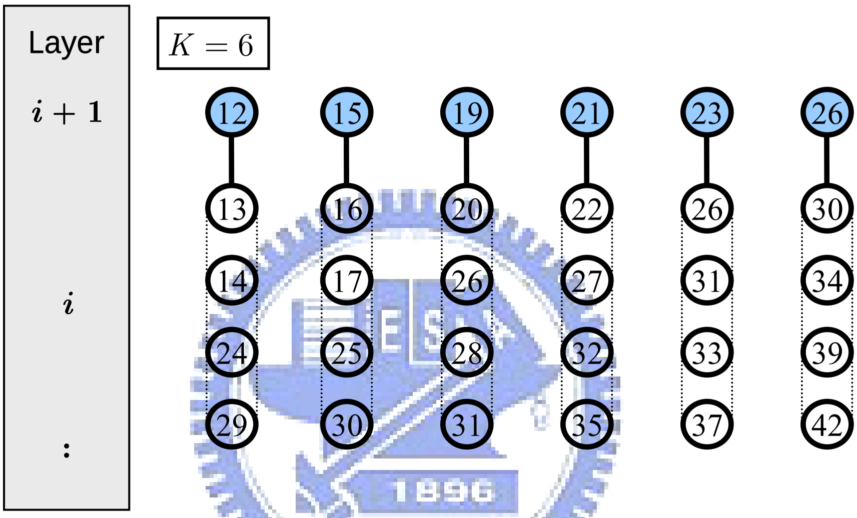


Figure 3-2: *K* sorted groups of the proposed method

When there are several sorted groups, it is efficient to apply the *merge algorithm* [20] to obtain the sorted group containing all elements. The merge algorithm is one of the algorithms that run sequentially over multiple sorted groups. The general merge algorithm has a set of pointers $\{p_1, p_2, \ldots, p_n\}$ that point to the positions in a set of the sorted groups $\{G_1, G_2, \ldots, G_n\}$. Initially they point to the first item in each group. The merge algorithm is described as follows:

Step 1. Extract those elements which $\{p_1, p_2, \ldots, p_n\}$ point to in their respective groups.

Step 2. Find out which element of those pointers point to with the minimum (or maximum) value.

Step 3. Advance one of those pointers to the next element in its group.

Step 4.

If any of $\{p_1, p_2, \ldots, p_n\}$ still points to data inside of $\{G_1, G_2, \ldots, G_n\}$

Go back to Step 1.

Else

The elements are all merged; stop the merge algorithm.

Figure 3-3 shows an example of the merge algorithm applied to two sorted groups.

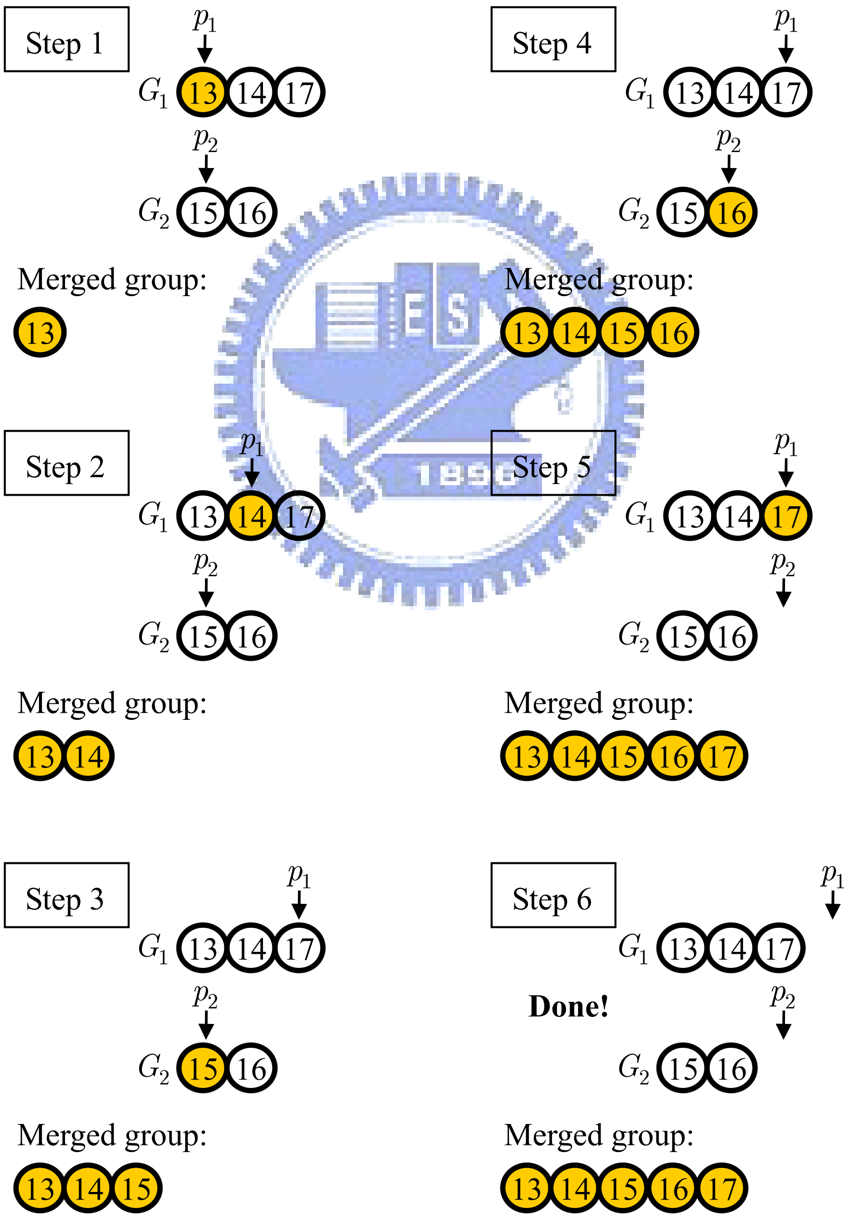


Figure 3-3: An example of the merge algorithm

To reduce the size of the storage memory, we repeatedly apply the merge algorithm which deals with two sorted groups at a time. The steps of the proposed sorting strategy of the *i*th layer are described as follows:

Step 1.

(a). $k \leftarrow 2$. Find the groups of the sorted child nodes of the (*k*-1)th and the *k*th parent nodes.

(b). Apply the merge algorithm to the two sorted groups to find K nodes with the minimum path weights.

Step 2.

$k \leftarrow k+1$.

Step 3.

If $P_{i+1}^{k} > P_{i}^{\max}$

The *K* survival nodes of this layer are found.

Else

(i). Find the sorted child nodes group of the *k*th parent node.

(ii). Apply the merge algorithm to the current survival group and the sorted child nodes group of the *k*th parent node to find *K* nodes with the minimum path weights.

(iii). Go back to Step 2.

P_{i}^{k} and $P_{i}^{\max}$ denote the path weight of the *k*th parent node in the *i*th layer and the maximum path weight of current survival node respectively. Figure 3-4 shows an example of the proposed method. Note that in the proposed method, only a small part of the candidate nodes have to be merged; hence, it significantly reduces the sorting complexity.

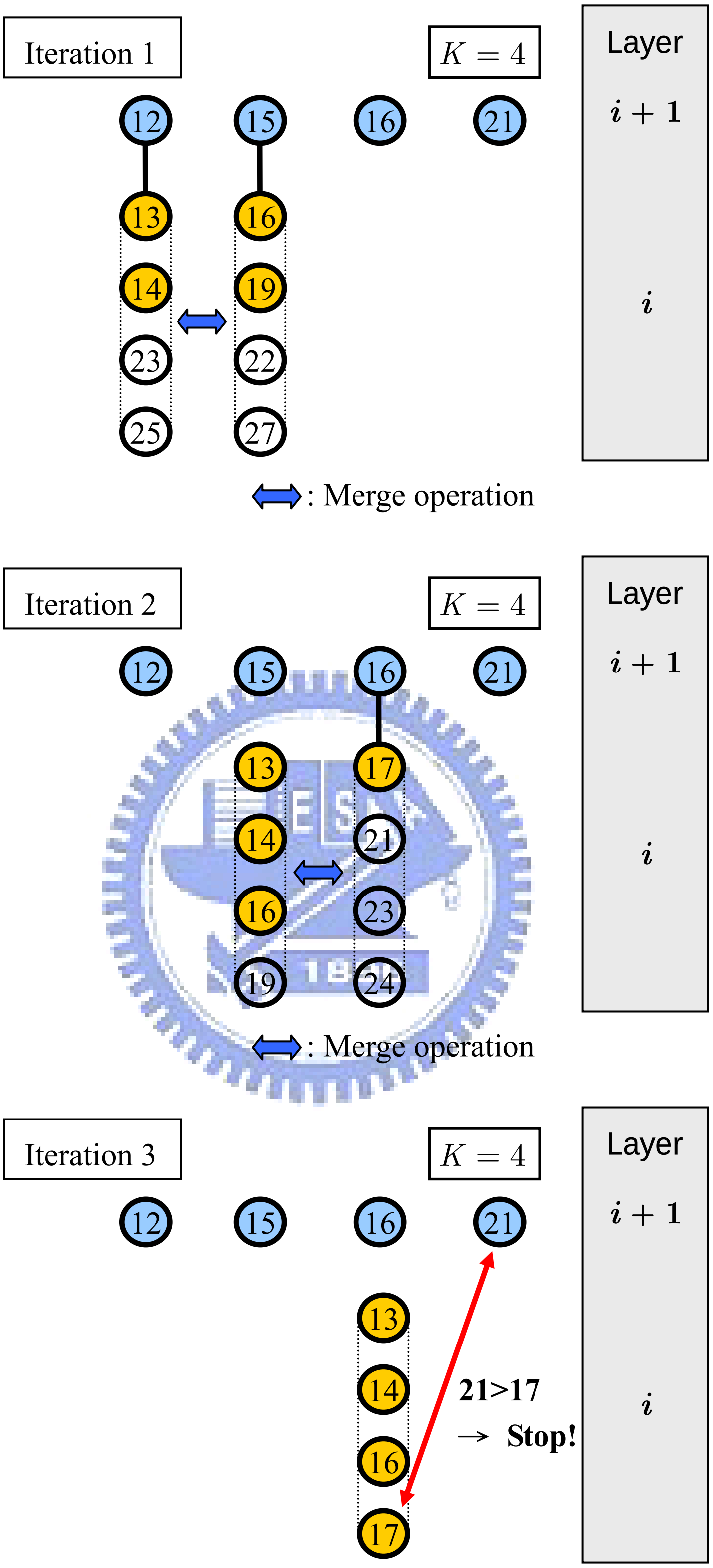


Figure 3-4: An example of the proposed sorting method

3.3 Efficient Complex Domain Search Method

In each layer, searching for the child nodes of each parent node has to satisfy the following constraint:

$$\left\|s_i - x_i\right\|^2 \le C_i^2, \tag{3.10}$$

where $s_i = y_i^{''} / r_{i,i}$ and $C_i^2 = r_{i,i}^{-2} \cdot \left[\left(d^{'}\right)^2 - \sum_{j=i+1}^{N} \left\| y_j^{''} - r_{j,j} x_j \right\|^2 \right]$. We assume the modulation scheme of the transmitted symbols is QAM. The geometrical interpretation of (3.10) is that finding all possible symbols inside the circle centered at s_i with radius C_i which is shown in Figure 3-5.

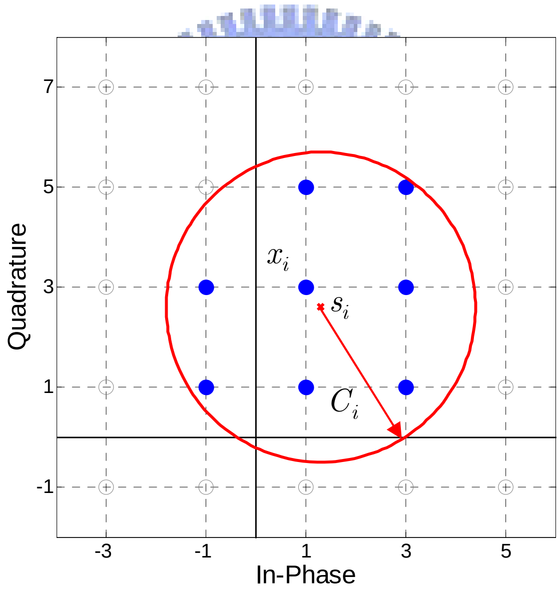


Figure 3-5: Geometrical interpretation of the search constraint

To search the symbols efficiently, it is useful to construct the table of the symbol sequences within a given region [21], [22]. In the proposed method, we construct the table of the sequences of the 11 nearest constellation symbols for those points bounded by $\{0,\ 1,\ j,\ 1+j\}$. For example $s_i = 0.7 + 0.2j$, the sequence of the nearest 11 symbols in an ascending order according to their distances from s_i is {35, 36, 27, 28, 43, 44, 34, 37, 26, 29, 42} which is shown in Figure 3-6.

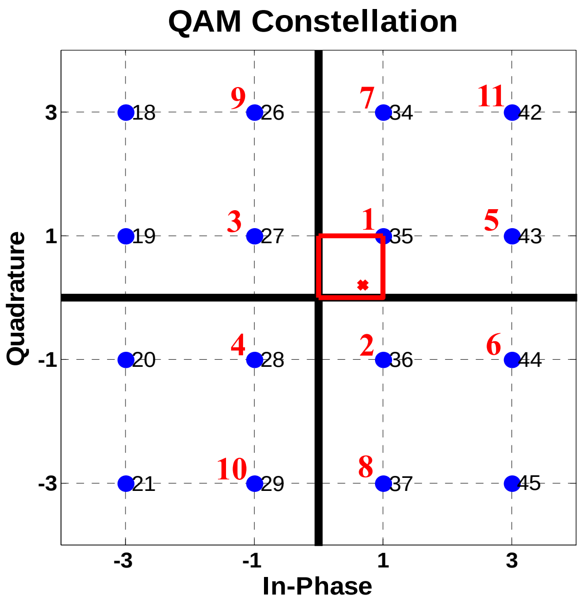


Figure 3-6: An example of the nearest 11 points from the search center s_i

For those points having the same symbol sequence, we will classify them into the same search group and share the same symbol sequence. Figure 3-7 shows the boundaries of the search groups and the corresponding symbol sequences are listed in Table 3-1.

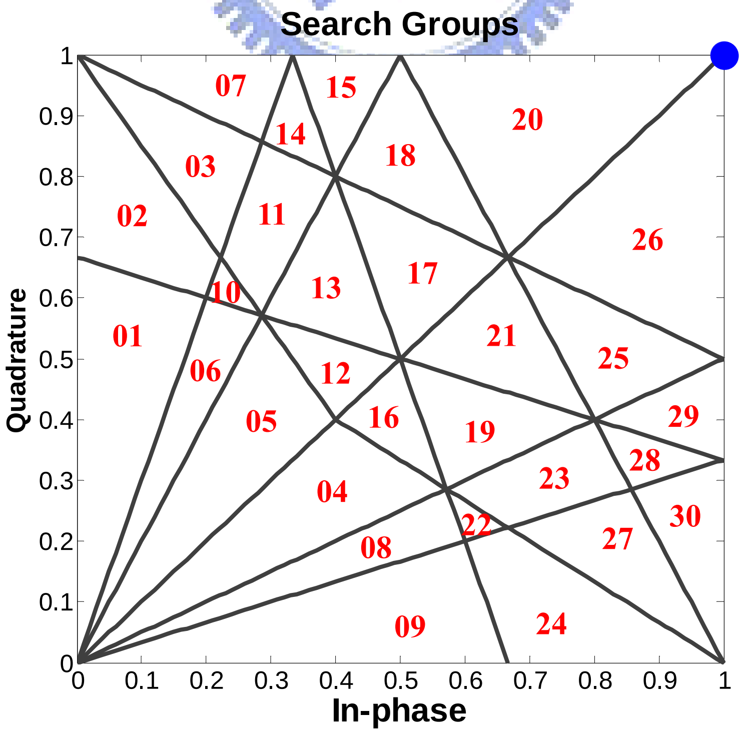


Figure 3-7: Boundaries of the search groups

Table 3-1: List of candidate sequences

Group ID	Candidate Sequence
01	35→27→36→28→34→26→43→19→44→20→37
02	35→27→36→28→34→26→43→19→44→20→42
03	35→27→36→28→34→26→43→19→44→42→20
04	35→36→27→28→43→34→44→26→37→19→29
05	35→27→36→28→34→43→26→44→19→37→20
06	35→27→36→28→34→26→43→44→19→20→37
07	35→27→36→34→28→26→43→19→44→42→20
08	35→36→27→28→43→44→34→26→37→29→19
09	35→36→27→28→43→44→34→37→26→29→19
10	35→27→36→28→34→26→43→44→19→20→42
11	35→27→36→28→34→26→43→44→19→42→20
12	35→27→36→28→34→43→26→44→19→37→42
13	35→27→36→28→34→43→26→44→19→42→37
14	35→27→36→34→28→26→43→44→19→42→20
15	35→27→36→34→28→26→43→44→42→19→20
16	35→36→27→28→43→34→44→26→37→19→42
17	35→27→36→28→34→43→26→44→42→19→37
18	35→27→36→34→28→43→26→44→42→19→37
19	35→36→27→28→43→34→44→26→37→42→19
20	35→27→36→34→43→28→26→44→42→19→37
21	35→36→27→28→43→34→44→26→42→37→19
22	35→36→27→28→43→44→34→26→37→29→42
23	35→36→27→28→43→44→34→26→37→42→29

Group ID	Candidate Sequence
24	35→36→27→28→43→44→34→37→26→29→42
25	35→36→27→43→28→34→44→26→42→37→19
26	35→36→27→43→34→28→44→26→42→37→19
27	35→36→27→28→43→44→34→37→26→42→29
28	35→36→27→43→28→44→34→26→37→42→29
29	35→36→27→43→28→44→34→26→42→37→29
30	35→36→27→43→28→44→34→37→26→42→29

Due to the symmetry property of the QAM constellation shown in Figure 3-8, those points lying in the region bounded by $\{1+j,\ 1-j,\ -1+j,\ -1-j\}$ in quadrant II, III and IV can use the same table of quadrant I by the following transformation:

$$\begin{aligned}
&\text{swap}\left(\mathrm{Re}\left(s_i\right),\mathrm{Im}\left(s_i\right)\right)\\
&\text{if }\ \left(\mathrm{Re}\left(s_i\right)\geq 0 \text{ and } \mathrm{Im}\left(s_i\right)<0\right)\\
&\begin{cases}\mathrm{Re}\left(\tilde{s}_i\right)\leftarrow \mathrm{Re}\left(s_i\right)\\ \mathrm{Im}\left(\tilde{s}_i\right)\leftarrow -\mathrm{Im}\left(s_i\right)\end{cases}\\
&\text{else if }\ \left(\mathrm{Re}\left(s_i\right)<0 \text{ and } \mathrm{Im}\left(s_i\right)<0\right)\\
&\begin{cases}\mathrm{Re}\left(\tilde{s}_i\right)\leftarrow -\mathrm{Re}\left(s_i\right)\\ \mathrm{Im}\left(\tilde{s}_i\right)\leftarrow -\mathrm{Im}\left(s_i\right)\end{cases}\\
&\text{else if }\ \left(\mathrm{Re}\left(s_i\right)<0 \text{ and } \mathrm{Im}\left(s_i\right)\geq 0\right)\\
&\begin{cases}\mathrm{Re}\left(\tilde{s}_i\right)\leftarrow -\mathrm{Re}\left(s_i\right)\\ \mathrm{Im}\left(\tilde{s}_i\right)\leftarrow \mathrm{Im}\left(s_i\right)\end{cases}
\end{aligned}\tag{3.11}$$

We first use the transformed search center $\tilde{s}_i$ to find the nearest 11 candidate symbols by looking up the table of the symbol sequences. When the candidate symbol x_i is found, we will transform it back to the original quadrant by

$$
\begin{aligned}
&\operatorname{swap}\left(\operatorname{Re}\left(x_i\right), \operatorname{Im}\left(x_i\right)\right) \\
&\text{if } \left(\operatorname{Re}\left(s_i\right) \geq 0 \text{ and } \operatorname{Im}\left(s_i\right) < 0\right) \\
&\begin{cases} \operatorname{Re}\left(x_i\right) \leftarrow -\operatorname{Re}\left(x_i\right) \\ \operatorname{Im}\left(x_i\right) \leftarrow \operatorname{Im}\left(x_i\right) \end{cases} \\
&\text{else if } \left(\operatorname{Re}\left(s_i\right) < 0 \text{ and } \operatorname{Im}\left(s_i\right) < 0\right) \\
&\begin{cases} \operatorname{Re}\left(x_i\right) \leftarrow -\operatorname{Re}\left(x_i\right) \\ \operatorname{Im}\left(x_i\right) \leftarrow -\operatorname{Im}\left(x_i\right) \end{cases} \\
&\text{else if } \left(\operatorname{Re}\left(s_i\right) < 0 \text{ and } \operatorname{Im}\left(s_i\right) \geq 0\right) \\
&\begin{cases} \operatorname{Re}\left(x_i\right) \leftarrow \operatorname{Re}\left(x_i\right) \\ \operatorname{Im}\left(x_i\right) \leftarrow -\operatorname{Im}\left(x_i\right) \end{cases}
\end{aligned}
\tag{3.12}
$$

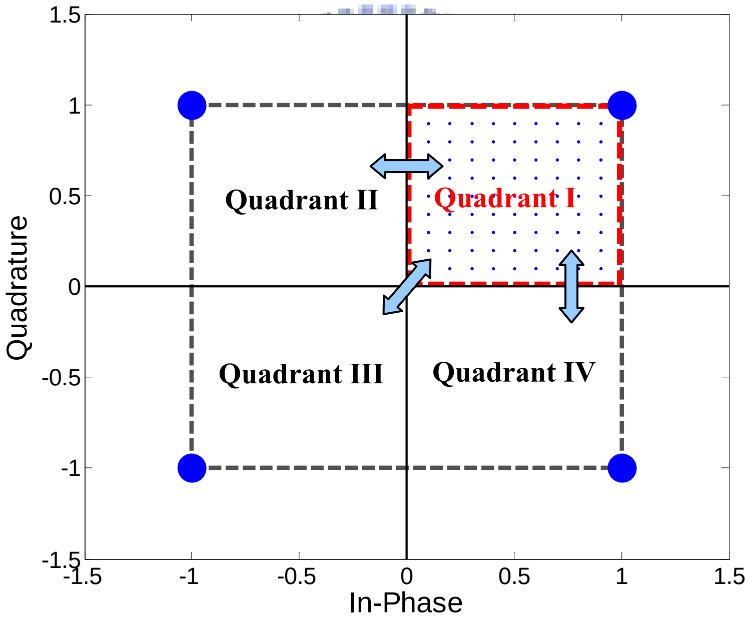


Figure 3-8: Symmetry property of the QAM constellation

For those search centers lying outside the region bounded by {1+j, 1-j, -1+j, -1-j}, we can first round them into the bounded region and then use the transformation relationship described above to find the relative nearest constellation symbols. Then the nearest constellation symbols are obtained by adding the coordinate offsets to the coordinates of the relative nearest constellation points which is shown in Figure 3-9. To improve the search efficiency, a special quantization for the search center s_i is performed at the *N*th layer. When the search center s_i is far away from the QAM symbols, we will round it into a given boundary by

$$\operatorname{Re}(s_c)=\begin{cases}\operatorname{Re}(s_c) & \text{if} \quad |\operatorname{Re}(s_c)|\le B_{\max}\\ \operatorname{sign}[\operatorname{Re}(s_c)]\cdot B_{\max} & \phantom{\text{if}} \quad |\operatorname{Re}(s_c)|> B_{\max}\end{cases} \tag{3.13}$$

$$\operatorname{Im}(s_c)=\begin{cases}\operatorname{Im}(s_c) & \text{if} \quad |\operatorname{Im}(s_c)|\le B_{\max}\\ \operatorname{sign}[\operatorname{Im}(s_c)]\cdot B_{\max} & \phantom{\text{if}} \quad |\operatorname{Im}(s_c)|> B_{\max}\end{cases} \tag{3.14}$$

$$B_{\max} = \left[0.5\left(\sqrt{M_c}-1\right)+1\right]d \tag{3.15}$$

where M_c is the size of the QAM symbol alphabet and d is the minimum distance between two QAM symbols. Figure 3-10 shows an example of the special quantization at the *N*th layer.

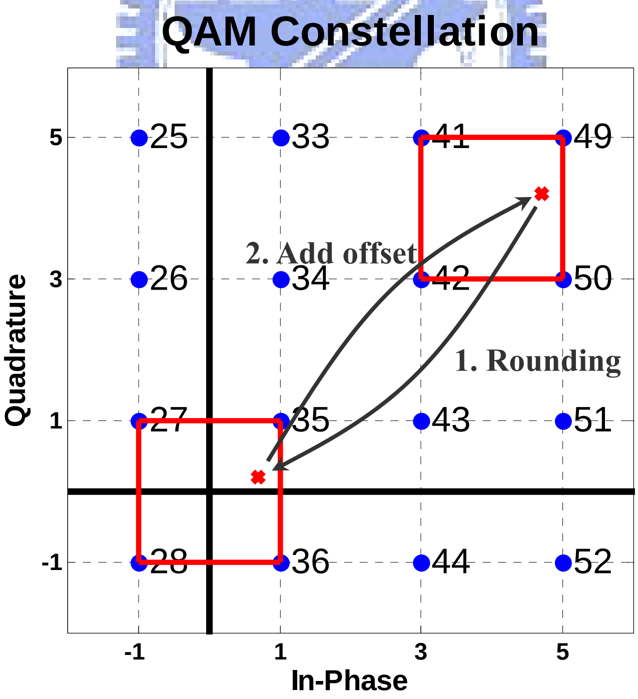


Figure 3-9: Rounding operation of the search center

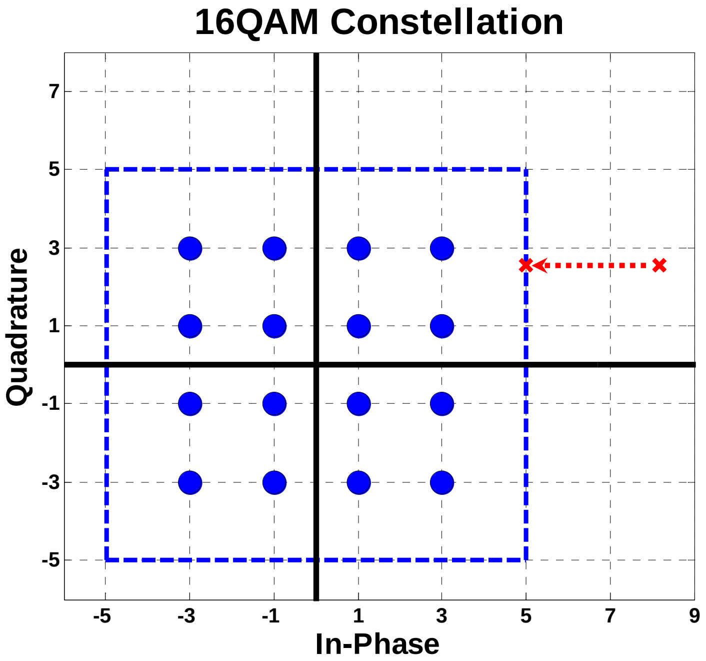


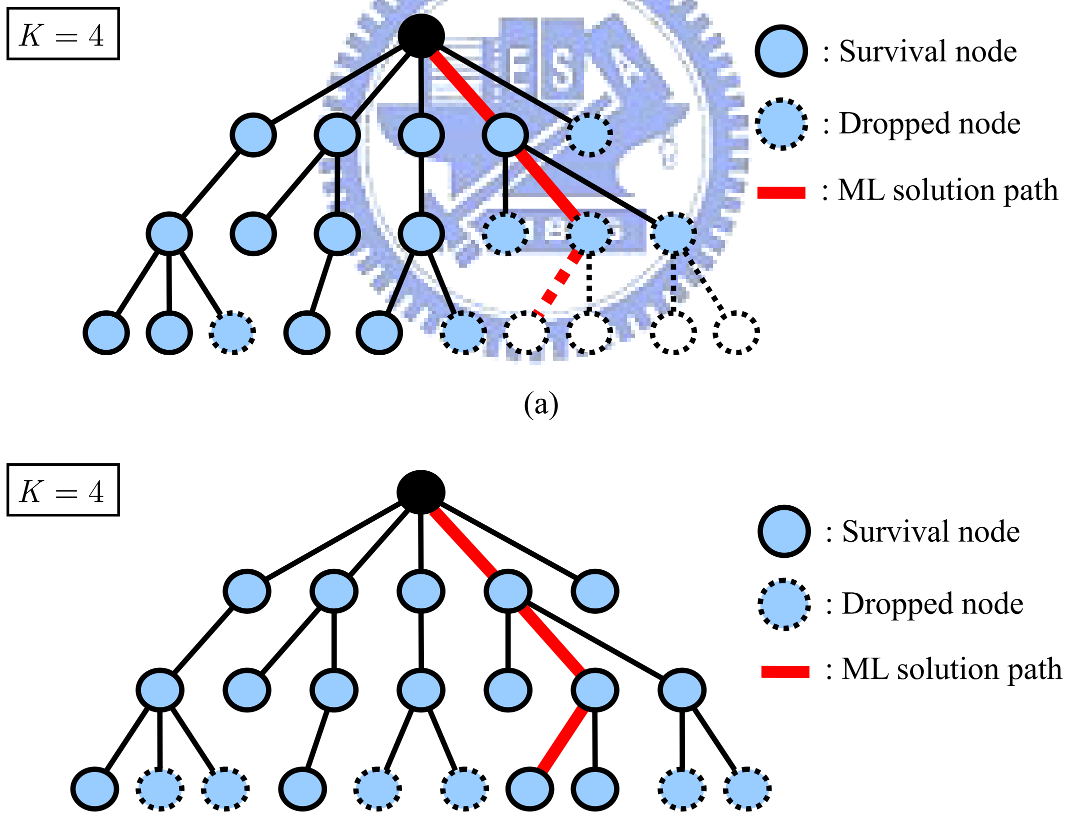
Figure 3-10: Special quantization for the search center at the *N*th layer

3.4 Preprocessing

The channel matrix is preprocessed with some techniques, which reduces the complexity of searching the candidates or improves the performance of the *K*-best SDA. There are many preprocessing techniques such as scaling [23], lattice reduction [24], [25] and the column permutation [25]. In this thesis, we consider the column permutation of the channel matrix. The permutation order is based on the column norms of the channel matrix in an ascending order. The ordering mechanism increases the expectation of $r_{i,i}^2$ in the higher layers which has two benefits. First, for a fixed value of K in the *K*-best SDA, increasing the expectation of $r_{i,i}^2$ in the higher layers reduces the effective search range of the candidates; therefore, it reduces the probability of the ML path being dropped in the early stages. Another benefit is that it constrains the growth of the tree and hence reduces the complexity of searching the candidate nodes.

3.5 ML-Like Search Strategy

One way to reduce the complexity of the conventional *K*-best SDA is to choose a smaller number of the survival nodes in each layer. However, this will cause performance degradation in the error rate. Instead of choosing a sufficiently large K to achieve a near-ML performance, we propose an ML-like search strategy. The proposed ML-like search method preserves all candidate nodes in the higher layers and then starts to keep only *K* candidate nodes at a suitable layer with a smaller *K*. Figure 3-11 shows the comparison between the conventional *K*-best algorithm and the proposed ML-like search method with $K = 4$. The determination of the number of layers performing the proposed ML-like search will be discussed in Chapter 4.



(a)

(b)

Figure 3-11: Comparison between (a) conventional *K*-best algorithm

(b) proposed ML-like search strategy

3.6 Computer Simulations

In this section, we simulate the symbol-error-rate (SER) of the proposed complex K-best SDA and compare it with SE SDA and the conventional *K*-best SDA. For the 4×4 MIMO systems, only the 4th layer performs the proposed ML-like search method. For the 8×8 MIMO systems, we first search all possible candidates in the 8th and the 7th layers. Then we keep K survival nodes at the 7th layer. For each simulation, we apply the preprocessing technique mentioned in Section 3.4 to all algorithms to do a fair comparison.

Figure 3-12 shows the simulations of SER as a function of SNR with 4×4 16QAM and $K = 8$. Note that the performance of the SE SDA is the same as the ML detection. With the proposed ML-like search method performed at the 4th layer, the performance of the proposed *K*-best SDA is better than the conventional *K*-best SDA. The SER of the proposed *K*-best SDA is close to that of the SE SDA. In contrast, the SER of the conventional *K*-best SDA tends to saturate at given value. This phenomenon is due to the fact that the conventional *K*-best SDA with a smaller *K* drops the ML-path with a high probability when the channel is in poor condition and the channel has a fixed probability of being in poor condition. Figure 3-13 shows the simulations of SER with 8×8 16QAM and $K = 14$. In this simulation, the value of *K* has to be chosen larger to reduce the probability of the ML path being dropped in the higher layers. Hence, the performance gap between the proposed *K*-best SDA and the conventional *K*-best SDA is smaller than the 4×4 case. However, the proposed *K*-best SDA still achieves a 0.5 dB gain over the conventional *K*-best SDA.

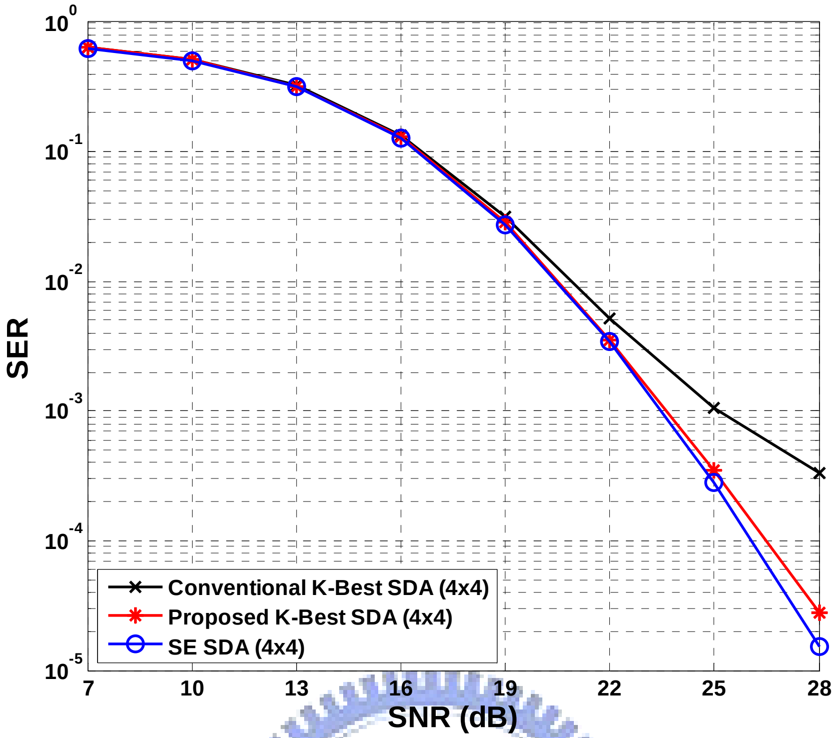


Figure 3-12: Simulations of 4×4 16QAM with $K = 8$

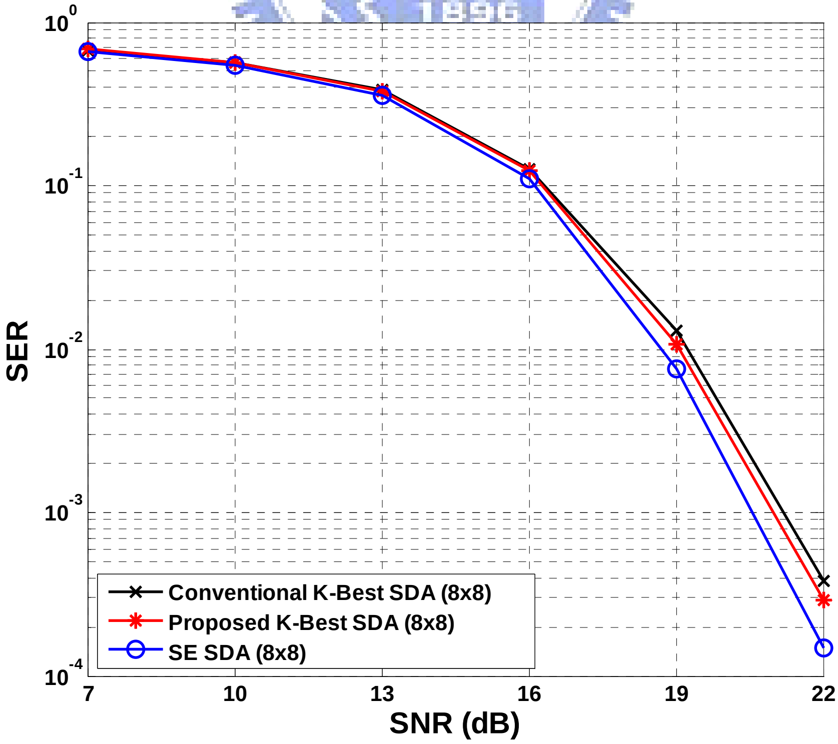


Figure 3-13: Simulations of 8×8 16QAM with $K = 14$

Figure 3-14 shows the simulations of SER with 4×4 64QAM and $K=8$. The proposed *K*-best SDA achieves nearly a 2 dB gain over the conventional *K*-best SDA at $\mathrm{SER}=10^{-3}$. Note that the performance gap between the proposed *K*-best SDA and the conventional *K*-best SDA is larger than that of the 4×4 16QAM case. This is because the probability of the ML-path being dropped becomes higher when the size of the modulation symbol alphabet becomes larger. However, the proposed ML-like search method keeps all possible candidates in the higher layers; hence, it significantly reduces the probability of the ML path being dropped. Figure 3-15 shows the simulation of SER with 8×8 64QAM and $K=36$. The proposed *K*-best SDA works better than the 8×8 16QAM case. This is because we choose $K=36$ which covers only half of the 64QAM constellation points. Hence, the probability of the ML-path being dropped of the conventional *K*-best SDA is higher than that of the 8×8 16QAM case.

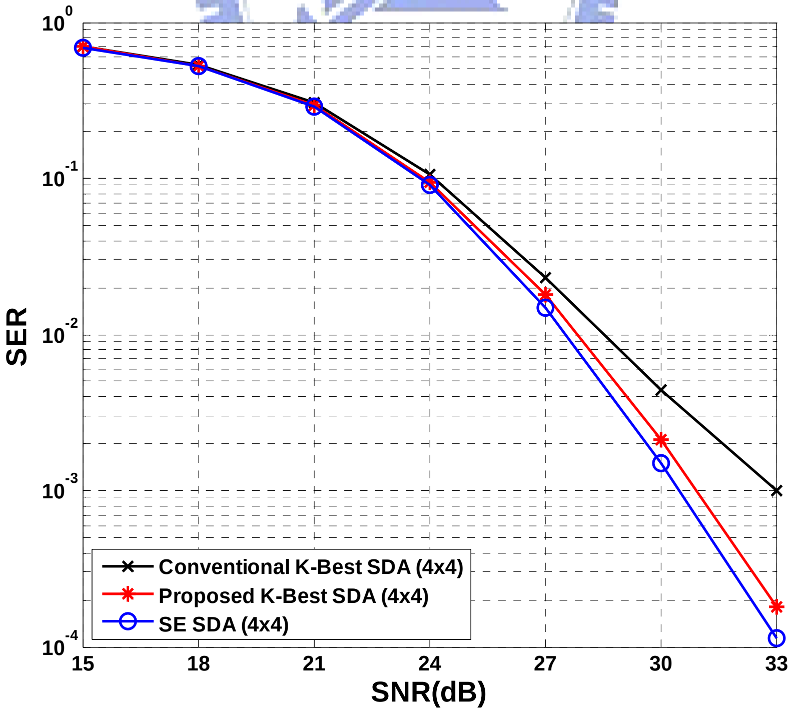


Figure 3-14: Simulations of 4×4 64QAM with $K=8$

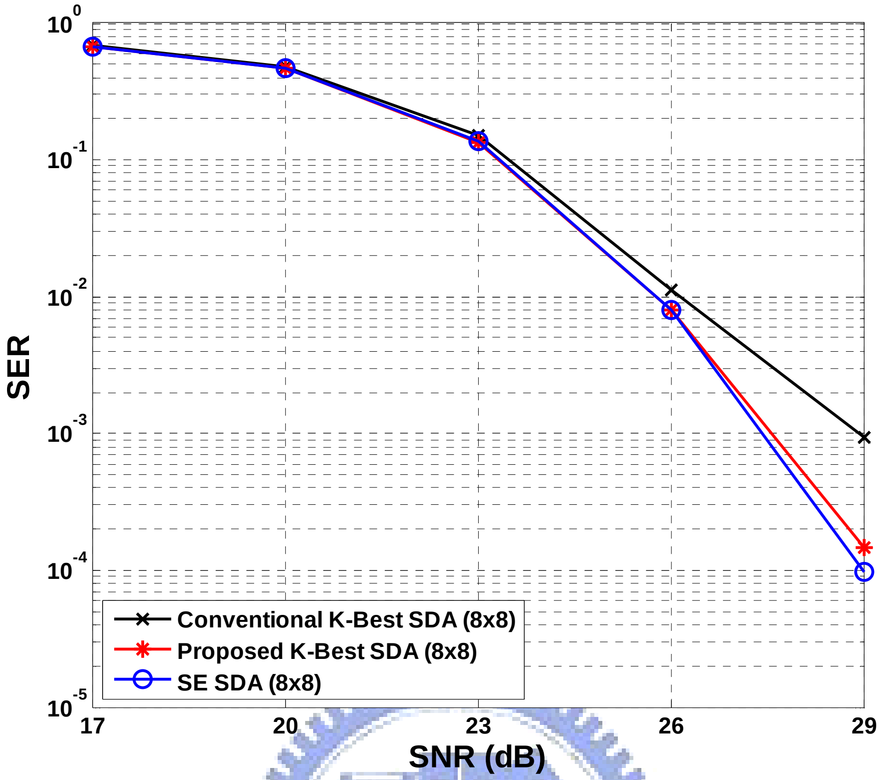


Figure 3-15: Simulations of 8×8 64QAM with $K = 36$

3.7 Summary

In this chapter, we give a detailed description of the proposed complex *K*-best SDA. Applying the *K*-best algorithm directly to the complex-valued signal reduces the number of search layers. The merge algorithm combined with the proposed complex domain search method works more efficiently than the conventional sorting algorithm. The column ordering of the channel matrix reduces the number of candidates in the higher layers and also reduces the probability of the ML-path being dropped. To further enhance the performance of the *K*-best SDA, we propose an ML-like search method which improves the performance in SER without requiring a sufficiently large value of *K*. Simulations show that the proposed *K*-best SDA works better than the conventional *K*-best SDA under 16QAM and 64QAM MIMO systems with different dimensions.

Chapter 4

Hardware Architecture and Sorting Complexity Analysis of Proposed Algorithm

In this chapter, we first give the hardware architecture of the proposed complex domain search method. In Section 4.2, we discuss the determination of the number of layers performing the proposed ML-like search method which is based on the derived pdf of $r_{i,i}^2$. In Section 4.3, we give an analysis of the sorting complexity in each layer under different operation modes. The complexity and the performance simulations will be provided in Section 4.4 to show that the proposed *K*-best SDA has lower complexity and better performance than the conventional *K*-best SDA.

4.1 Hardware Architecture

The hardware architecture of the proposed complex domain search method is introduced in this section. We call the proposed complex candidate search unit the “Complex Candidate Generator” (CCG) and its functional block diagram is shown in Figure 4-1.

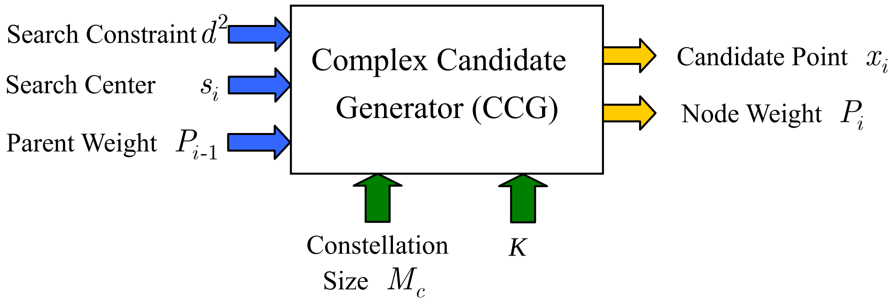


Figure 4-1: Functional block diagram of the CCG unit

When we give a set of input $\left(s_i, d^2, P_{i\text{-}1}\right)$, CCG will output candidate points according to their path weights in an ascending order. The value of M_c is used for checking whether the candidate point x_i lies outside the constellation boundary or not. The value of K controls the number of the output candidate points. The CCG unit is composed of three functional blocks which is shown in Figure 4-2.

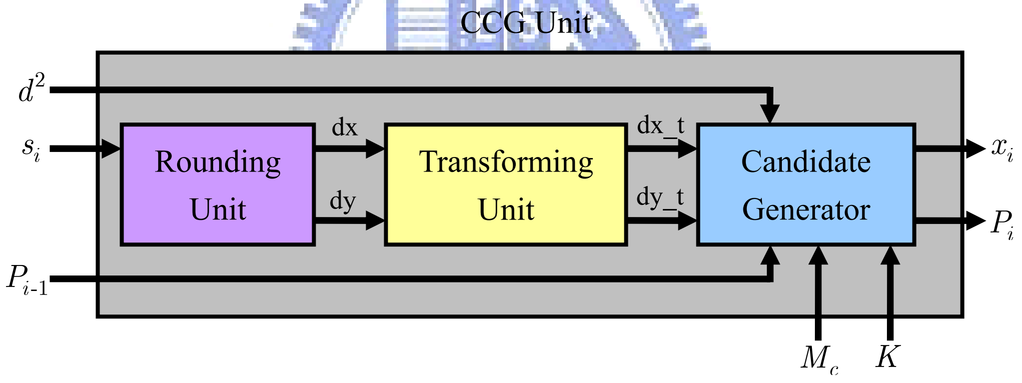


Figure 4-2: Detailed block diagram of the CCG unit

When we input a new search center s_i, CCG first rounds it to the relative position $\tilde{s}_i$ which lies inside the region bounded by $\{1+j,\ 1-j,\ \text{-}1-j,\ \text{-}1+j\}$. The rounding procedures are as follows:

Rounding for $\operatorname{Re}\left(s_i\right)$:

$$\begin{cases} \text{X\_offset} = \operatorname{floor}\left(\operatorname{Re}\left(s_i\right)\right) + \operatorname{mod}\left(\operatorname{floor}\left(\operatorname{Re}\left(s_i\right)\right), 2\right) \\ \operatorname{Re}\left(\tilde{s}_i\right) = \operatorname{Re}\left(s_i\right) - \text{X\_offest} \end{cases} \tag{4.1}$$

Rounding for $\mathrm{Im}(s_i)$:

$$\begin{cases} \text{Y\_offset} = \mathrm{floor}(\mathrm{Im}(s_i)) + \mathrm{mod}(\mathrm{floor}(\mathrm{Im}(s_i)),2) \\ \mathrm{Im}(\tilde{s}_i) = \mathrm{Im}(s_i) - \text{Y\_offest} \end{cases} \tag{4.2}$$

The rounding unit of $\mathrm{Re}(s_i)$ is shown in Figure 4-3 which is composed of only two adders and two multiplexers. We use 2's complement to store the value of $\mathrm{Re}(s_i)$. *S* is the sign bit of $\mathrm{Re}(s_i)$ and *b*0 is the LSB of the integer part of $\mathrm{Re}(s_i)$. The value of dx is used for $\mathrm{Re}(\tilde{s}_i)$. Since the rounding procedures of $\mathrm{Im}(s_i)$ is the same as $\mathrm{Re}(s_i)$, the rounding unit of $\mathrm{Im}(s_i)$ is the same as $\mathrm{Re}(s_i)$.

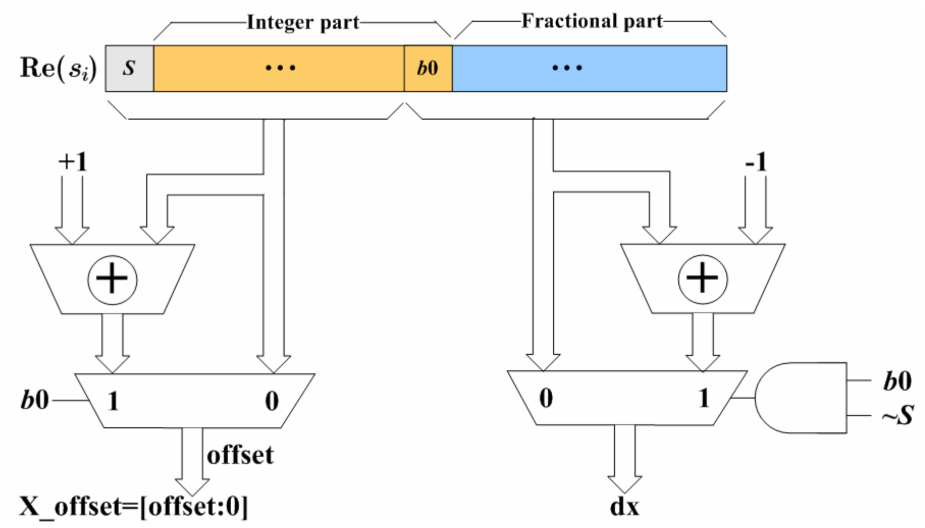


Figure 4-3: Rounding unit of $\mathrm{Re}(s_i)$

Now, the search center s_i is rounded to the point $\tilde{s}_i = \mathrm{dx} + j \cdot \mathrm{dy}$ which lies in the region bounded by $\{1+j,\ 1-j,\ -1-j,\ -1+j\}$. In the next step, if $\tilde{s}_i$ lies in quadrants II, III or IV, the CCG unit will map $\tilde{s}_i$ into quadrant I by the transformation in (3.11). The transformation circuit is shown in Figure 4-4. The multiplexers will choose a right data path based on the values of MSBs of dx and dy. dx_t and dy_t denote the transformed value of $\mathrm{Re}(\tilde{s}_i)$ and $\mathrm{Im}(\tilde{s}_i)$ respectively.

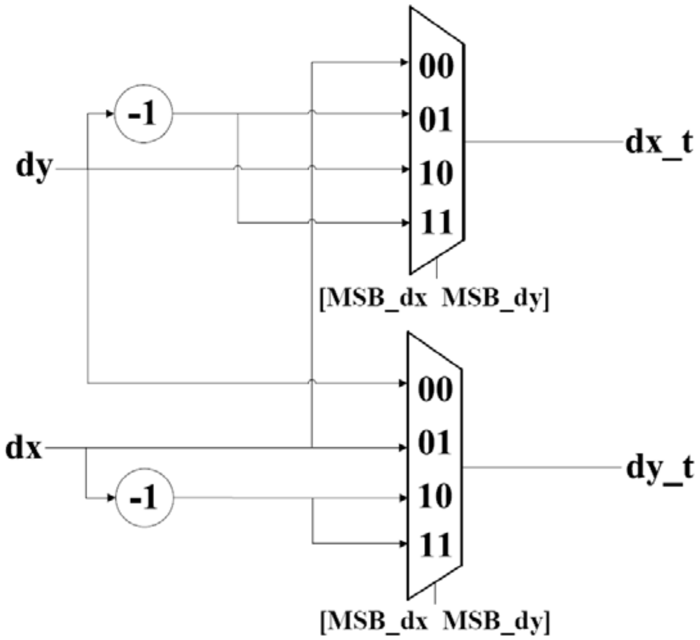


Figure 4-4: Transforming unit of $\tilde{s}_i$

The set (dx_t, dy_t) is sent to the candidate generator unit to generate the candidate point x_i and its corresponding path weight P_i. The hardware architecture of the candidate generator is shown in Figure 4-5. The contents of the group ID and its corresponding candidate sequence are stored in ROM 1 and ROM 2 respectively. We first use (dx_t, dy_t) as a memory address to obtain the group ID stored in ROM 1. Then we use the group ID as a memory address to obtain the sequence of the candidate points stored in ROM 2. After adding the offset pair (X_offset, Y_offset) to the coordinates of the found candidate point, the constellation boundary checker will check whether the found point lies inside the constellation boundary. If the found candidate point lies inside the constellation boundary, the quadrant restoring unit will transform it back to the original quadrant. The distance calculator calculates the value of $\|s_i - x_i\|^2$. Multiplying the value of $\|s_i - x_i\|^2$ by $r_{i,i}^2$ and adding the parent weight $P_{i\text{-}1}$ to the multiplied result, we obtain the path weight P_i of the found point. The path weight P_i will be compared with the search constraint d^2. If the found candidate point lies outside the constellation boundary or its path weight P_i is larger than the search constraint d^2, the value of the valid indicator will be 0 which indicates the found point is not a valid one.

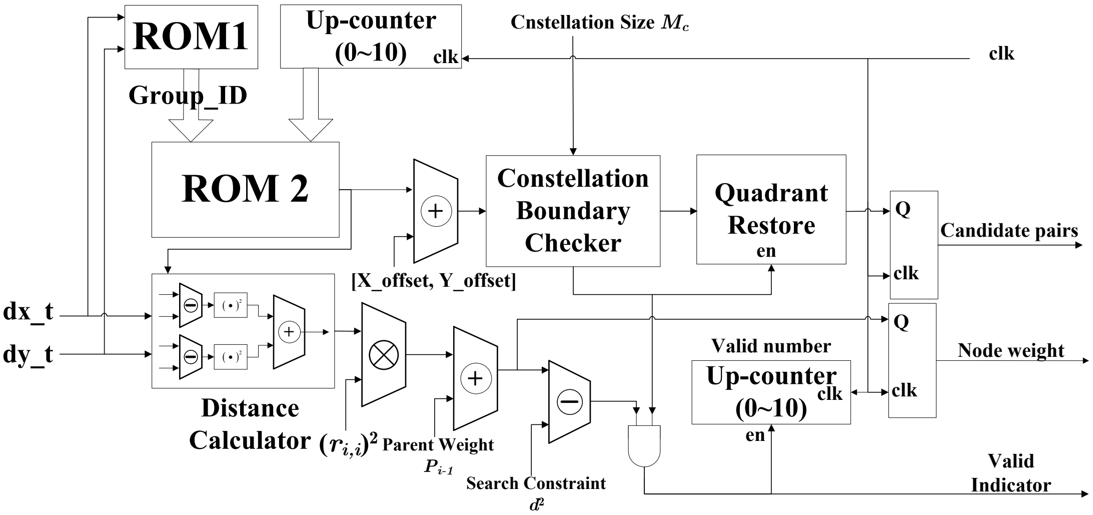


Figure 4-5: Hardware architecture of the candidate generator

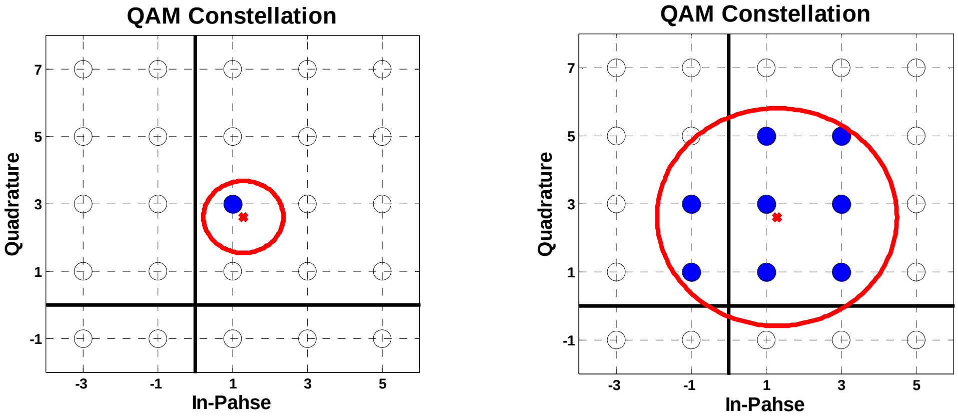
4.2 Discussion on Proposed ML-Like Search Strategy

For the search of candidates at the *N*th layer, the candidate symbol should satisfy the following constraint:

$$\left\| s_N - x_N \right\|^2 \le \left(\frac{d'}{r_{N,N}} \right)^2 . \tag{4.3}$$

It is obvious that $\frac{1}{r_{N,N}}$ will enlarge the constraint region when $r_{N,N}$ is smaller than 1. In such case, the probability of the ML path being dropped will increase when we keep only K nodes at the *N*th layer. Hence, the number of layers performing the proposed ML-like search depends on the distribution of $r_{i,i}^2$. Figure 4-6 (a) and 4-6 (b) show the impact of $r_{N,N}$ on the constrained search region. To determine the number of layers performing the proposed ML-like search, we derive the pdf and cdf of $r_{i,i}^2$

after channel column ordering. Based on the derived results, we can determine the number of layers performing the proposed ML-like search under different M and N.



(a) (b)

Figure 4-6: Search constraint at the *N*th layer with $d^{'} = 1.1$ (a) $r_{N,N} = 1$

(b) $r_{N,N} = 0.33$

Reordering the columns of **H** according to their vector norms in an ascending order, we have

$$\mathbf{H}_o = \left[\mathbf{h}_{o(1)}, \mathbf{h}_{o(2)}, \cdots, \mathbf{h}_{o(N)}\right], \tag{4.4}$$

where $\left\|\mathbf{h}_{o(1)}\right\| \leq \left\|\mathbf{h}_{o(2)}\right\| \leq \cdots \leq \left\|\mathbf{h}_{o(N)}\right\|$. From [25], $\mathbf{h}_{o(i)}$ can be expressed as

$$\mathbf{h}_{o(i)} = \sqrt{X_i}\boldsymbol{\theta}_i, \tag{4.5}$$

where X_i is the *i*th order statistic of N independent $\text{Gamma}(M,1)$ distribution random variables with $X_1 \leq X_2 \leq \cdots \leq X_N$ and $\{\boldsymbol{\theta}_i\}$s are i.i.d. uniformly distributed on the unit sphere in $\mathbb{C}^M$. Note that X_i and $\boldsymbol{\theta}_i$ are independent. With the QR decomposition of $\mathbf{H}_o = \mathbf{Q}_o\mathbf{R}_o$, we are now going to characterize the distribution of the square of the diagonal entries of $\mathbf{R}_o$ denoted by $r_{o,i,i}^2$.

Letting $\mathbf{Q}_o = \left[\mathbf{q}_{o(1)}, \mathbf{q}_{o(2)}, \cdots, \mathbf{q}_{o(N)}\right]$ and performing the QR decomposition of

$\mathbf{H}_o$, we obtain

$$r_{o,i,i}^2 = X_i\left[1-\sum_{k=1}^{i\text{-}1}\left(\mathbf{q}_{o(k)}^H\boldsymbol{\theta}_i\right)^2\right] = X_i\left[1-\sum_{k=1}^{i\text{-}1}\boldsymbol{\theta}_i^2(k)\right] = \begin{cases} X_i & ,\text{for } i=1 \\ X_i S_i, & \text{for } 2\le i\le N \end{cases}, \tag{4.6}$$

where

$$S_i = \left[1-\sum_{k=1}^{i\text{-}1}\boldsymbol{\theta}_i^2(k)\right] \tag{4.7}$$

and $\boldsymbol{\theta}_i(k)$ denotes the *k*th element of $\boldsymbol{\theta}_i$. Note that the second equation holds due to the fact that the distribution of $\boldsymbol{\theta}_i$ is invariant under the orthogonal transformation $\mathbf{Q}_o$. To derive the cdf of $r_{o,i,i}^2$, we should first obtain the pdf of X_i and S_i. The pdf of X_i is available in [26] as

$$f_{X_i}(x) = \frac{N!}{(i-1)!(N-i)!}[F(x)]^{i\text{-}1}[1-F(x)]^{N\text{-}i}f(x), \tag{4.8}$$

where

$$f(x) = \frac{x^{M-1}\cdot e^{-x}}{\Gamma(M)} \quad \text{for } x>0, \tag{4.9}$$

$$F(x) = 1-\sum_{i=0}^{M-1}\frac{x^i}{i!}e^{-x}. \tag{4.10}$$

From [27], $\boldsymbol{\theta}_i$ can be modeled from a 2*M*-dimensional random vector $\mathbf{V} = \begin{bmatrix} v_1 & v_2 & \cdots & v_{2M}\end{bmatrix}^T$ with v_i ~ i.i.d. $N(0,1)$, where

$$\boldsymbol{\theta}_i(k) = \frac{v_{2k-1}+j\cdot v_{2k}}{\|\mathbf{V}\|} = \frac{v_{2k-1}+j\cdot v_{2k}}{\sqrt{v_1^2+v_2^2+\cdots+v_{2M}^2}}. \tag{4.11}$$

Due to the fact that $\boldsymbol{\theta}_i^H\boldsymbol{\theta}_i = 1$, S_i can be rewritten as

$$S_i = \left[1-\sum_{k=1}^{i-1}\boldsymbol{\theta}_i^2(k)\right] = \left[\sum_{k=1}^{M-i+1}\boldsymbol{\theta}_i^2(k)\right] \tag{4.12}$$

Substituting (4.11) into (4.12), we have

$$S_i = \sum_{k=1}^{M-i+1}\boldsymbol{\theta}_i^2(k) = \frac{v_1^2+v_2^2+\cdots+v_{2\cdot(M-i+1)}^2}{v_1^2+v_2^2+\cdots+v_{2M}^2} = \frac{Q_i}{P_i}, \tag{4.13}$$

where Q_i and P_i are chi-square random variables with $2\cdot(M-i+1)$ and $2M$ degrees of freedom respectively.

The joint pdf of Q_i and P_i is

$$\begin{aligned} f_{Q_i,P_i}(q,p) &= f_{Q_i}(q)\cdot f_{P_i\text{-}Q_i}(p-q) \\ &= f_{\chi(2\cdot(M-i+1))}(q)\cdot f_{\chi(2i-2)}(p-q) \\ &= \frac{q^{(M-i)}\cdot(p-q)^{(i-2)}\cdot e^{-p/2}}{2^M\cdot\Gamma(M-i+1)\cdot\Gamma(i-1)}, \quad \text{for } p>0 \text{ and } q>0, \end{aligned} \tag{4.14}$$

where $f_{\chi(k)}(x)$ denotes the pdf of the chi-square random variable with k degrees of freedom. The pdf of S_i can be obtained by

$$\begin{aligned} f_{S_i}(s) &= \int_{\text{-}\infty}^{\infty} |p| f_{Q_i,P_i}(ps,p)\,dp \\ &= \int_0^{\infty} p\cdot\frac{(ps)^{(M\text{-}i)}\cdot(p-ps)^{(i-2)}\cdot e^{-p/2}}{2^M\cdot\Gamma(M-i+1)\cdot\Gamma(i-1)}dp \\ &= \frac{s^{(M-i)}\cdot(1-s)^{(i-2)}}{2^M\cdot\Gamma(M-i+1)\cdot\Gamma(i-1)}\int_0^{\infty} p^{(M-1)}\cdot e^{-p/2}dp \\ &= \frac{s^{(M\text{-}i)}\cdot(1-s)^{(i\text{-}2)}\cdot(M-1)!}{\Gamma(M-i+1)\cdot\Gamma(i-1)}. \end{aligned} \tag{4.15}$$

Since X_i and S_i are independent, the joint pdf of X_i and S_i is

$$f_{X_i,S_i}(x,s) \;=\; f_{X_i}(x)\cdot f_{S_i}(s) \tag{4.16}$$

The cdf of $r_{o,i,i}^2$ for $2\le i\le N$ can be obtained by

$$F_{r_{o,i,i}^2}(r) \;=\; \int_0^1\left\{\int_0^{r/s} f_{X_i,S_i}(x,s)\,dx\right\}ds \;=\; \int_0^1\int_0^{r/s} f_{X_i}(x)f_{S_i}(s)\,dxds \tag{4.17}$$

Finally, the pdf and the cdf of $r_{o,i,i}^2$ are as follows:

For $i=1$

$$\begin{cases} f_{r_{o,i,i}^2}(r) \;=\; \dfrac{N!}{(N-1)!(M-1)!}\left[\displaystyle\sum_{k=0}^{M-1}\frac{r^k}{k!}e^{-r}\right]^{N-1}\cdot r^{M-1}e^{-r} \\ F_{r_{o,i,i}^2}(r) \;=\; \displaystyle\int_0^r \frac{N!}{(N-1)!(M-1)!}\left[\sum_{k=0}^{M-1}\frac{x^k}{k!}e^{-x}\right]^{N-1}\cdot x^{M-1}e^{-x}dx \end{cases}, \tag{4.18}$$

For $2 \le i \le N$

$$F_{r_{o,i,i}^2}(r) = C_{ii}\int_0^1\int_0^{r/s}\left[1-\sum_{k=0}^{M-1}\frac{x^k}{k!}e^{-x}\right]^{i-1}\left[\sum_{k=0}^{M-1}\frac{x^k}{k!}e^{-x}\right]^{N-i} x^{M-1}e^{-x}(s)^{M-i}(1-s)^{i-2}\,dxds \quad (4.19)$$

where

$$C_{ii} = \frac{N!}{(N-i)!(M-i)!(i-1)!(i-2)!}. \quad (4.20)$$

Figure 4-7 shows the pdf curves of $r_{o,i,i}^2$ of the 4×4 MIMO channel. We can see that the probability of $r_{o,i,i}^2$ being smaller than 1 in the 4th layer is larger than that of the other layers. Hence, only the 4th layer has to perform the proposed ML-like search method. Figure 4-8 shows the pdf curves of $r_{o,i,i}^2$ of the 8×8 MIMO channel. In this case, the probabilities of $r_{o,i,i}^2$ being smaller than 1 in the 8th and the 7th layers are larger than that of the other layers. However, the number of the possible candidates in the 7th layer is $(M_c)^2$ in the worst case which is too large to store for hardware implementation when M_c is large. Hence, we first keep all possible candidates in the 8th layer. For the 7th layer, we first find all possible candidates and then start to keep *K* survival nodes with the minimum path weights.

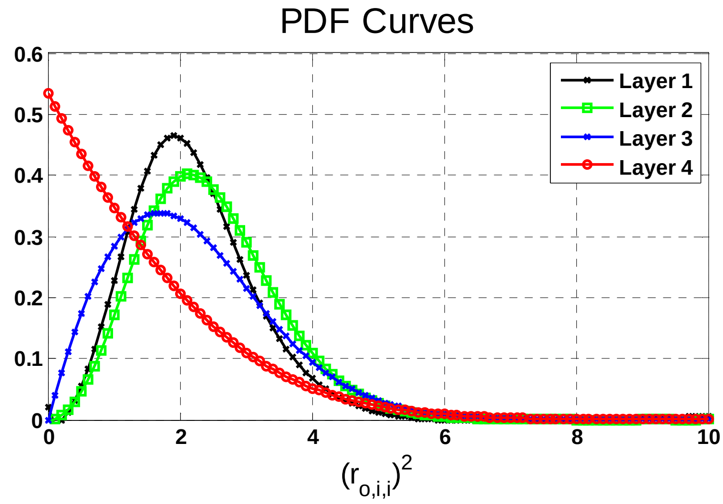


Figure 4-7: PDF curves of $r_{o,i,i}^2$ of the 4×4 channel

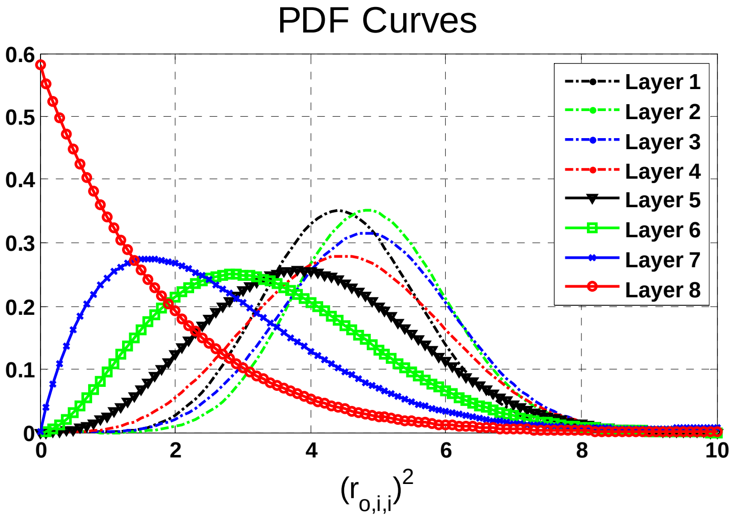


Figure 4-8: PDF curves of $r_{o,i,i}^2$ of the 8×8 channel

To further reduce the complexity, the proposed ML-like search is performed only when any of $r_{i,i}$ for $N - L_{ML} + 1 \le i \le N$ is smaller than a given threshold T_r, where L_{ML} is the number of layers performing the proposed ML-like search. Recalling that for the *N*th layer, the search for the candidate symbols should satisfy the following constraint:

$$r_{N,N}^2 \left\| s_N - x_N \right\|^2 \le \left(d' \right)^2 \tag{4.21}$$

In the proposed algorithm, we keep only k constellation symbols which are the nearest ones from s_N, where $k = \min(K, 11)$; hence, the value of $\left\| s_N - x_N \right\|^2$ has a limit range. From this property, the threshold T_r is chosen based on the following criterion:

$$\frac{D^2 \cdot T_r^2}{\mathrm{E}\left\{ \left(d' \right)^2 \right\}} = 1, \tag{4.22}$$

$$D^2 = \min\left(D_{K,\min}^2, D_{11,\min}^2 \right), \tag{4.23}$$

where $D_{K,\min}$ and $D_{11,\min}$ denote the minimum distances from the *K*th and the 11th nearest constellation symbols to s_N respectively. The geometrical interpretation of (4.22) is that when the nearest *k* constellation symbols fail to cover all possible symbols inside the average constraint region $\mathrm{E}\left\{\left(d'\right)^2\right\}$, the proposed ML-like search method will be performed to keep all possible symbols. Note that the value of $\mathrm{E}\left\{\left(d'\right)^2\right\}$ varies with SNR. We can choose $\mathrm{E}\left\{\left(d'\right)^2\right\}$ at the SNR when the symbol error rate of the proposed *K*-best SDA without the proposed ML-like search method tends to be T times larger than that of the ML detection. When T is close to 1, the performance of the proposed *K*-best SDA will be close to the ML detection; however, its complexity will increase because the probability of performing the proposed ML-like search increases. This shows a trade-off between the complexity and the performance.

4.3 Sorting Complexity Analysis

The main reduction of the complexity in the proposed *K*-best SDA is about sorting. Hence, we focus on the analysis of the sorting complexity in this section. The evaluation of the sorting complexity is based on the number of data “compare and select” (CS) operations. We first consider the sorting complexity in the *N*th layer. For the normal *K*-best operation, there is no sorting complexity because the candidates are directly generated from the table of the candidate sequences. When the ML-like operation is performed, we assume that all constellation points have to be sorted and the applied sorting algorithm is the bubble sort algorithm. Hence, the number of CS of the ML-like operation in the *N*th layer, $N_{CS,ML,N}$, is

$$N_{CS,ML,N} = \left(M_c - 1\right) + \left(M_c - 2\right) + \cdots + 2 + 1 = \frac{M_c \cdot \left(M_c - 1\right)}{2}, \qquad (4.24)$$

where M_c denotes the constellation size. Table 4-1 shows the sorting complexity in the *N*th layer.

Table 4-1: Sorting complexity in the *N*th layer

	***K*-Best Operation**	**ML-Like Operation**
Total Complexity	None	$\frac{M_c \cdot (M_c - 1)}{2}$

For the (*N*－1)th layer, we assume that each parent node has K child nodes. Constructing the group of the sorted child nodes of each parent node has no sorting complexity when the *K*-best operation is performed; however, when the ML-like operation is performed, we apply the bubble sort algorithm to find *K* child nodes of each parent node. The number of CS for constructing a sorted group of the ML-like operation, $N_{CS,ML,N-1,G}$, is

$$N_{CS,ML,N-1,G} = (M_c - 1) + (M_c - 2) + \cdots + (M_c - K) = \frac{K \cdot M_c \cdot (2M_c - K + 1)}{2}. \quad (4.25)$$

Applying the merge algorithm to two sorted groups takes *K* (CS)s. We consider the worst case that the merge algorithm will be applied to the last group of the sorted child nodes. Hence, the merge algorithm is performed $(K-1)$ times if the *N*th layer applies the normal *K*-best operation and is performed $(M_c - 1)$ times if the *N*th layer applies the ML-like operation. The sorting complexity in the (*N*－1)th layer is shown in Table 4-2.

For the ith $(2 \le i \le N-2)$ layer, the sorted child nodes of each parent node are always obtained from the table of the candidate sequences which takes no sorting complexity. The number of parent nodes of each layer is assumed to be K and we consider the worst case that the merge algorithm will be applied to the last group of the sorted child nodes. The sorting complexity in the ith $(2 \le i \le N-2)$ layer is shown in Table 4-3.

Table 4-2: Sorting complexity in the (*N*-1)th layer

	K-Best Operation	**ML-Like Operation**
Obtain the sorted group	None	$\frac{K(2M_c-K+1)}{2}$
Merge operation	K	K
Total complexity	If *N*th layer applies *K*-best operation	$\frac{K\cdot M_c\cdot(2M_c-K+1)}{2}$ $+K\cdot(M_c-1)$
	$K\cdot(K-1)$	
	If *N*th layer applies ML-like operation	
	$K\cdot(M_c-1)$	

Table 4-3: Sorting complexity in the *i*th layer $(2\le i\le N-2)$

	***K*-Best Operation**
Obtain the sorted group	None
Merge operation	K
Total Complexity	$K\cdot(K-1)$

For the 1st layer, the proposed *K*-best SDA only has to find one child node of each parent node. Hence, performing the merge operation once takes one “compare and select” operation. The complexity in the 1st layer is shown in Table 4-4.

Table 4-4: Sorting complexity in the 1st layer

	Normal Operation
Obtain the sorted group	None
Merge operation	1
Total Complexity	$K-1$

4.4 Simulation Results

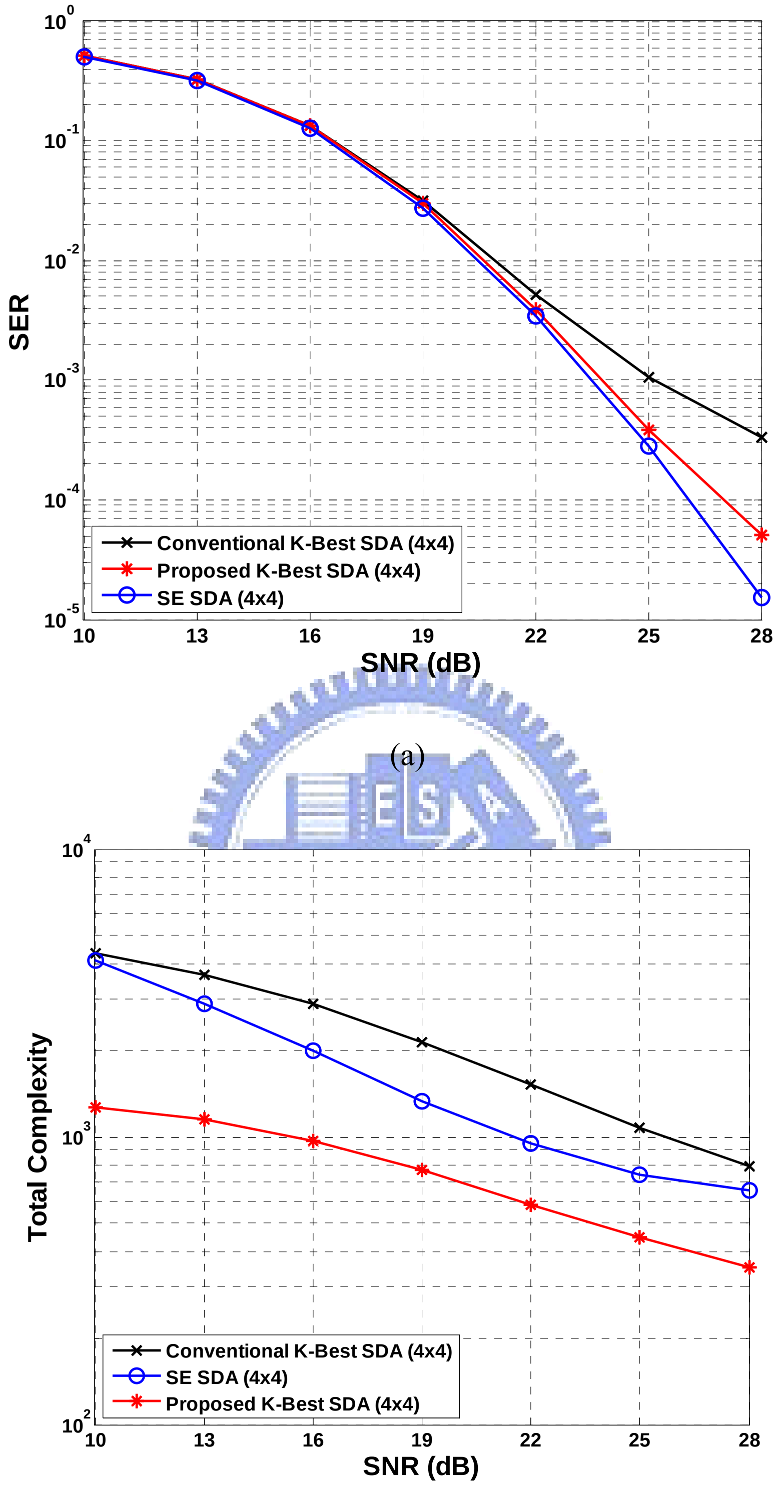
In this section, we simulate the SER and the complexity of the proposed *K*-best SDA and compare it with the SE SDA and the conventional *K*-best SDA. To compare the complexity of different algorithms, we define the complexity weight of different operations according to [13], [28], [29] which are shown in Table 4-5. The total complexity of each simulated algorithm is the sum of the number of times of each operation multiplied by its corresponding weight.

Table 4-5: Complexity weight of different operations

Operation	**Weight**	**Operation**	**Weight**
Real addition/subtraction	1	Complex multiplication	6
Real multiplication	1	Reading data from memory	1
Real division	2	Writing data to memory	1
Real square operation	1	Memory data comparison	1
Complex addition/subtraction	2		

Figure 4-9 (a) and Figure 4-9 (b) show the 4×4 16QAM simulations of SER and the complexity with $K = 8$. The threshold T_r is chosen to be 0.291 and its corresponding probability of performing the proposed ML-like search is 4.72 %. We can see that the conventional *K*-best SDA has higher complexity than that of the SE SDA. This phenomenon is due to the fact that the conventional *K*-best SDA visits more candidate nodes than the SE SDA when the number of the layers is small. The proposed efficient sorting method reduces the number of visited nodes and the sorting complexity in each layer; hence, the complexity of the proposed *K*-best SDA is lower than that of the SE SDA and the conventional *K*-best SDA. Comparing Figure 4-9 (a) with Figure 3-14, the SER curve of the proposed *K*-best SDA in Figure 4-9 (a) is nearly the same as that in Figure 3-14. This shows that the threshold constraint significantly reduces the probability of performing the proposed ML-like search and there is nearly no performance degradation of the proposed *K*-best SDA.

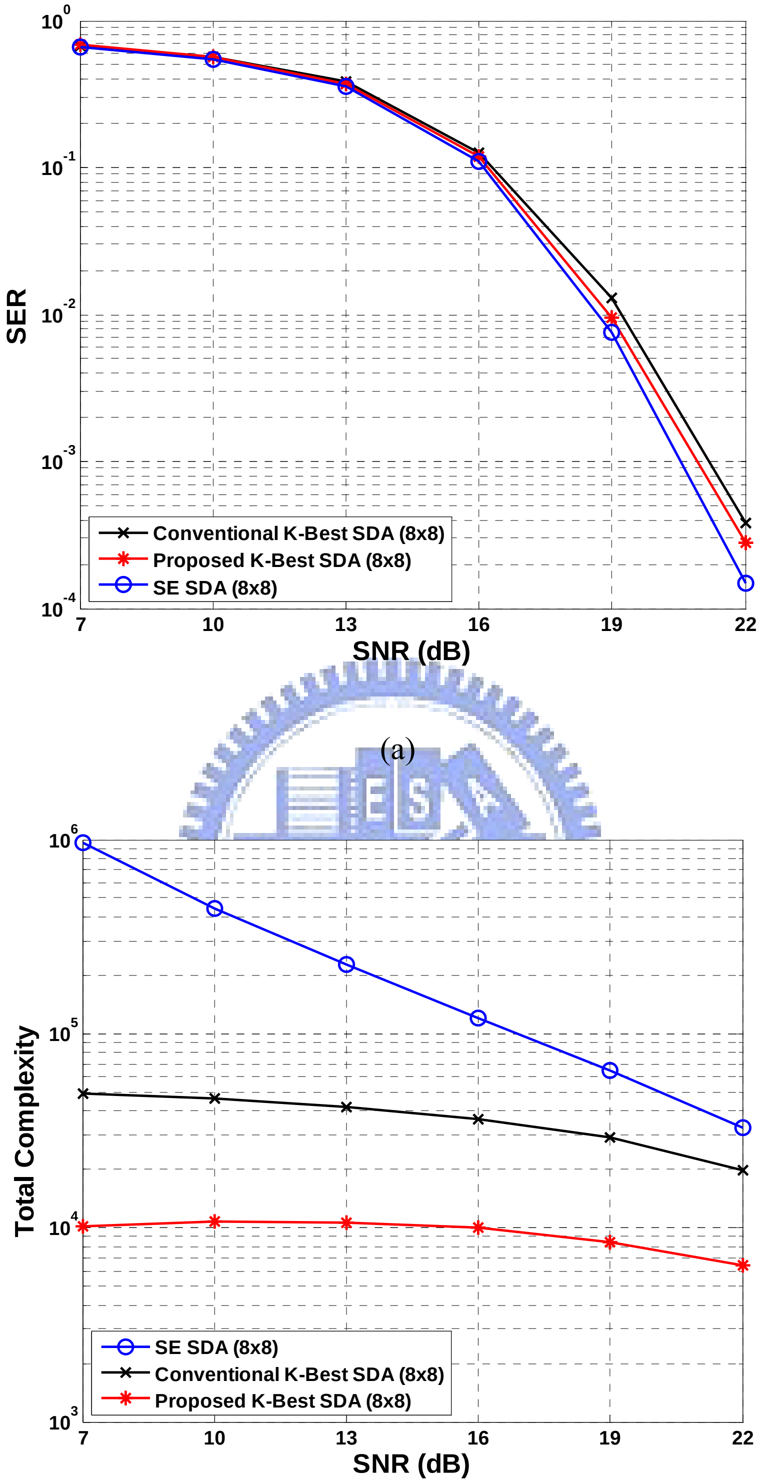
Figure 4-10 (a) and Figure 4-10 (b) show the 8×8 16QAM simulations of SER and the complexity with $K = 14$. The threshold T_r is chosen to be 0.833 and its corresponding probability of performing the proposed ML-like search is 38.4 %. The probability of performing the proposed ML-like search is higher than that of the 4×4 case because the probability of the ML-path being dropped in the *K*-best SDA is higher in the 8×8 case. We can see that the performance of the proposed *K*-best SDA is better than the conventional *K*-best SDA and the complexity of the proposed *K*-best SDA is lower than that of the SE SDA and the conventional *K*-best SDA. Because the number of search layers is larger in this case, the proposed sorting method reduces more complexity in sorting. Hence, we can see that the gap of the complexity between the proposed *K*-best SDA and the conventional *K*-best SDA is larger than that in the 4×4 case.



(a)

(b)

Figure 4-9: 4×4 16QAM simulations of (a) SER (b) complexity , $K = 8$ and $T_r = 0.291$



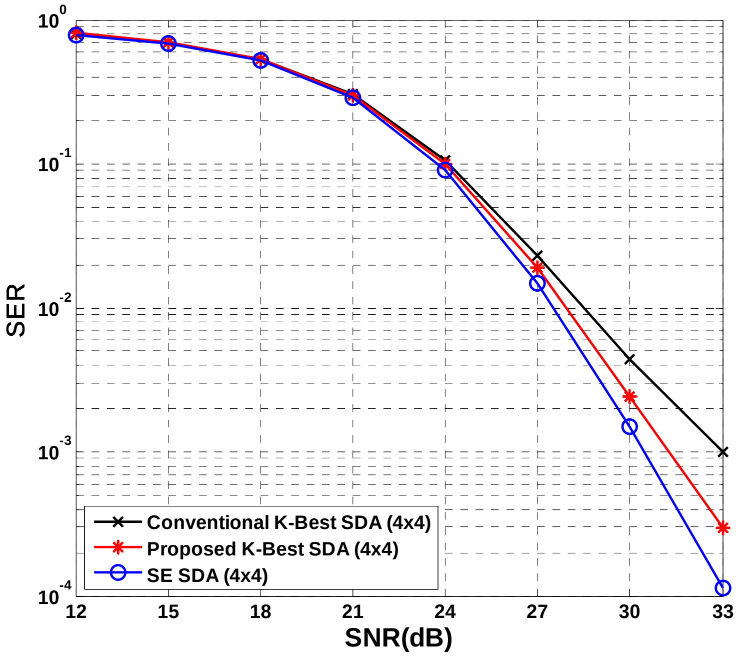
(a)

(b)

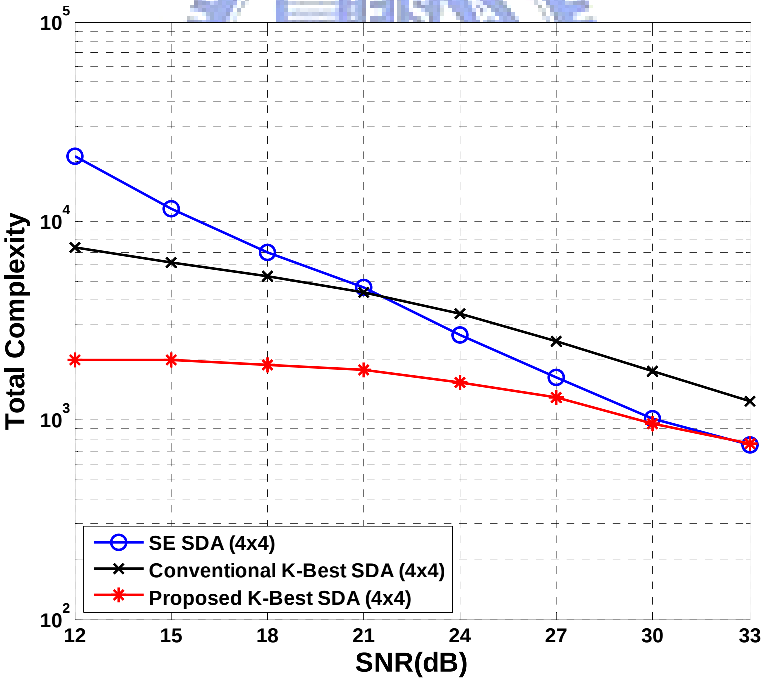
Figure 4-10: 8×8 16QAM simulations of (a) SER (b) complexity , $K=14$ and $T_r=0.833$

Figure 4-11 (a) and Figure 4-11 (b) show the 4×4 64QAM simulations of SER and the complexity with $K = 8$. The threshold T_r is chosen to be 0.353 and its corresponding probability of performing the proposed ML-like search is 6.44 %. In this case, the proposed *K*-best SDA still has better performance than the conventional *K*-best SDA and has lower complexity than that of the SE SDA and the conventional *K*-best SDA. The value of T_r is higher than that of the 4×4 16QAM case because the probability of the ML-path being dropped is higher when the size of the modulation symbol alphabet becomes larger. We can further improve the performance of the proposed *K*-best SDA by choosing a higher threshold; however, the complexity of the proposed *K*-best SDA may be higher than that that of the SE SDA in the high SNR range.

Figure 4-12 (a) and Figure 4-12 (b) show the 8×8 64QAM simulations of SER and the complexity with $K = 36$. The threshold T_r is chosen to be 1.143 and its corresponding probability of performing the proposed ML-like search is 65.8 %. In this case, the proposed *K*-best SDA still works better than the conventional *K*-best SDA. Due to the fact that the expectation of the constraint search radius is much larger than that in the 8×8 16QAM and the 4×4 64QAM cases, the threshold T_r has to set higher to reduce the probability of the ML-path being dropped in higher layers. We can see that the gap of the complexity between the proposed *K*-best SDA and the conventional *K*-best SDA is larger than that in the 8×8 16QAM case. The reason is that the proposed efficient sorting method reduces much more complexity when the size of the modulation alphabet becomes larger.

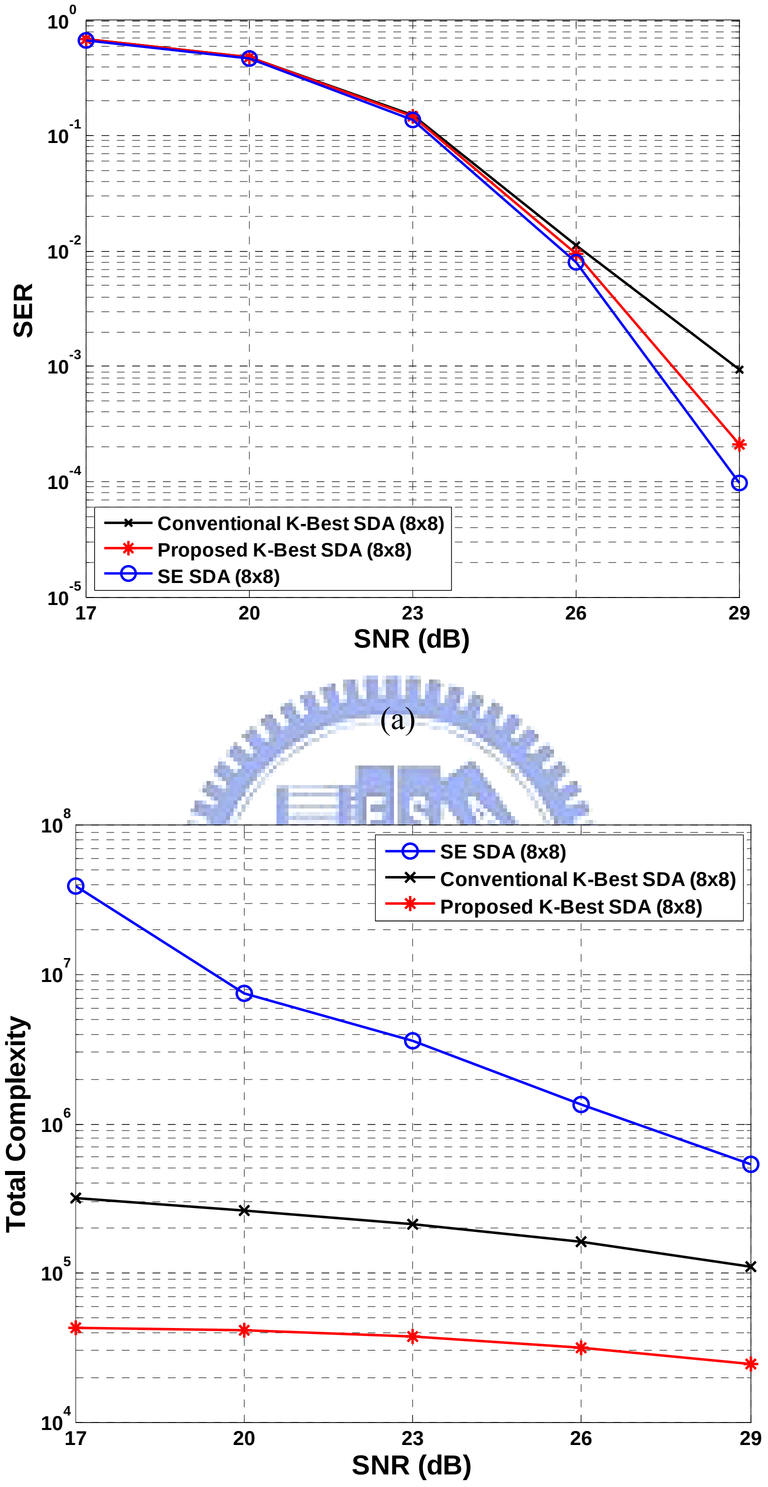


(a)



(b)

Figure 4-11: 4×4 64QAM simulations of (a) SER (b) complexity , $K = 8$ and $T_r = 0.353$



(a)

(b)

Figure 4-12: 8×8 64QAM simulations of (a) SER (b) complexity , $K = 36$ and $T_r = 1.143$

4.5 Summary

In this chapter, we provide the hardware architecture of the proposed complex domain search method. In Section 4.2, we discuss the determination of the number of layers performing the ML-like search. Based on the derived pdf of $r_{o,i,i}^2$, we can easily determine the number of layers performing the ML-like search method with different values of *M* and *N*. To reduce the overall complexity of the proposed algorithm, we give a threshold constraint to reduce the probability of performing the ML-like search method. In Section 4.3, we give a discussion of the sorting complexity of each layer of the proposed algorithm. In Section 4.4, we simulate the SER and the complexity of the proposed *K*-best SDA and compare it with the SE SDA and the conventional *K*-best SDA. Simulations show that the proposed *K*-best SDA has better performance and lower complexity than the conventional *K*-best SDA. Besides, the performance of the proposed *K*-best SDA is close to that of the SE SDA (ML performance).

Chapter 5

Conclusions and Future Works

In this thesis, we propose a complex *K*-best SDA with an efficient search architecture. In the proposed sorting method, only a small part of the candidates in each layer have to be sorted which significantly reduces the sorting complexity. The *K* survival nodes found by the proposed sorting method in each layer are the same as that found by the conventional sorting algorithm. Hence, the proposed sorting method does not sacrifice any performance. Moreover, the proposed complex domain candidate search method makes it possible that the child nodes of each parent node are already sorted without requiring any sorting. The hardware architecture of the proposed complex domain candidate search method is also provided. Finally, the proposed ML-like search reduces the probability of the ML-path being dropped in higher layers. As a result, the proposed *K*-best SDA has better performance and lower complexity than that of the conventional *K*-best SDA.

In Chapter 2, we give a review of the MIMO system and introduce several kinds of SDAs. The SE SDA achieves the ML performance with lower complexity. However, its decoding throughput is not stable. The *K*-best SDA has fixed decoding throughput and is suitable for pipelined hardware implementation. However, sorting the candidates in each layer requires a large amount of memory access. Hence, the theme of this thesis is focus on the reduction of the sorting complexity while improving the performance of the *K*-best SDA.

In Chapter 3, we give a detailed description of the proposed *K*-best SDA. The merge algorithm combined with proposed complex domain candidate search method has lower complexity than the conventional bubble sort algorithm. The concept of the proposed ML-like search method is to keep all possible candidates in higher layers to reduce to probability of the ML-path being dropped. With a small number of layers performing the proposed ML-like search, the proposed ML-like search method significantly improves the decoding performance of the *K*-best SDA when the value of *K* is small. Simulations demonstrate that the proposed *K*-best SDA achieves a near-ML performance which does not require a large value of *K*.

The hardware architecture of the proposed complex domain candidate search method is provided in Chapter 4. Besides, we discuss the determination of the number of layers performing the proposed ML-like search which is based on the statistical properties of the channel matrix after column ordering. We also propose a threshold constraint of the proposed ML-like search which simplifies the decision of the trade-off between the performance and the complexity. The detailed analysis of the sorting complexity of the proposed *K*-best SDA is also included. Simulations show that the proposed *K*-best SDA significantly reduces the complexity and improves the performance of the conventional *K*-best SDA. Moreover, the performance of the proposed *K*-best SDA is close to that of the ML detection.

The main contributions of this thesis are as follows. First, the proposed sorting method significantly reduces the sorting complexity and requires a smaller storage memory. Hence, it is more suitable to be implemented in hardware than the conventional bubble sort algorithm. In addition, the proposed complex domain candidate search method overcomes the difficulty in searching and sorting the constellation points in the complex domain within a constraint region. Hence, the complex-valued signals do not have to be converted into real-valued signals which reduces the number of search layers. Moreover, the proposed ML-like search method with the threshold constraint improves the decoding performance of the *K*-best SDA while maintaining lower complexity than that of the conventional *K*-best SDA and the SE SDA.

There are some future works worthy of further investigation. The first one is that the MIMO channel is assumed to be perfectly estimated. However, the MIMO channel cannot be perfectly estimated in practice. The second one is that it is not likely to have sufficiently many receive antennas in practice to decouple the spatial signals especially in the downlink path. When the number of receive antennas is less than the number of transmit antennas, searching for the optimal solution at the receiver becomes an underdetermined problem. In such a case, the *K*-best SDA cannot be directly applied to the underdetermined MIMO systems.

Bibliography

[1] E. Telatar, “Capacity of multi-antenna Gaussian channels,” *AT&T Bell Labs Internal Tech. Memo.*, June 1995.

[2] G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment using multiple antennas,” *Wireless Personal Commun.*, vol. 6, no.3, pp. 311-335, Mar. 1998.

[3] S. M. Alamouti, “A simple transmit diversity scheme for wireless communication,” *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1451-1458, Oct. 1998.

[4] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *AT&T Bell Labs Tech. J.*, pp. 41-59, Autumn 1996.

[5] A. J. Goldsmith and S. G. Chua, “Variable-rate variable-power MQAM for fading channels,” *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1218-1230, Oct. 1997.

[6] S. Catreux, V. Erceg, D. Gesbert and R. W. Heath, “Adaptive modulation and MIMO coding for broadband wireless data networks,” *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 108-115, Jun. 2002.

[7] G. J. Foschini, G. D. Golden, R. A. Valenzuela and P. W. Wolniansky, “Simplified processing for high spectral efficiency wireless communication employing multi-element arrays,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1841-1852, Nov. 1999.

[8] G. D. Golden, G. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, ”Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture,” *Electronic Letters*, vol. 35, no. 1, pp. 14-16, Jan. 1999.

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

[10] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806-2818, Aug. 2005.

[11] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm II. generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2819-2834, Aug. 2005.

[12] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181-191, Aug. 1994.

[13] K. W. Wong, C. Y. Tsui, R. S. Cheng and W. H. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE ISCAS'02*, vol. 3, pp. 273-276, May 2002.

[14] Q. W. Li and Z. F. Wang, "Improved K-best sphere decoding algorithms for MIMO systems," in *Proc. IEEE ISCAS'06*, pp. 1159-1162, May 2006.

[15] S. Roger, A. Gonzalez, V. Almenar and A. M. Vidal, "Combined K-best sphere decoder based on the channel matrix condition number," *ISCCSP'08*, pp. 1058-1061, May 2008.

[16] S. Mondal, K. N. Salama and W. H. Ali, "A novel approach for K-best MIMO detection and its VLSI implementation," in *Proc. IEEE ISCAS'08*, pp. 936-939, May 2008.

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., 1991.

[18] 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 *Proc. URSI ISSSE-98*, pp. 295-300, 1998.

[19] M. Grotschel, L. Lovasz and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. New York: Springer-Verlag, 1993.

[20] D. Knuth, *The Art of Computer Programming, Volumn 3: Sorting and Search*, 3rd ed., Addison-Wesley, 1997.

[21] A. Wiesel, X. Mestre, A. Pages and J. R. Fonollosa, "Efficient implementation of sphere demodulation," *IEEE Workshop on Signal Processing Advances in Wireless Communications* (SPAWC), pp. 36-40, Jun. 2003, Rome, Italy.

[22] N. Kobayashi, M. Ohnishi, M. Kova, N. Tsukamoto and Y. Kokuryou, "Simplified viterbi decoding for high speed data modem," in *Proc. IEEE GLOBECOM*'87, vol. 1, pp. 472-478, Nov. 1987, Japan.

[23] K. Lee and J. Chun, "ML symbol detection based on the shortest path algorithm for MIMO systems," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5477-5484, Nov. 2007.

[24] A. K. Lenstra, H. W. Lenstra and L. Lovasz, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 513-534, 1982.

[25] W. Zhao and G. B. Giannakis, "Reduced complexity closest point decoding algorithms for random lattices," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 101-111, Jan. 2006.

[26] N. Balakrishnan and A. C. Cohen, *Order Statistics and Inference Estimation Methods*, New York: Academic, 1991.

[27] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Comm. Assoc. Comp. Mach.*, vol. 2, no. 4, pp. 19-20, Apr. 1959.

[28] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 491-503, Mar. 2006.

[29] D. Pham, K. R. Pattipati, P. K. Willett and J. Luo, "An improved complex sphere decoder for V-BLAST systems," *IEEE Signal Process. Letters*, vol. 11, no. 9, pp.748-751, Sep. 2004.