# Probabilistic Searching For MIMO Detection Based On Lattice Gaussian Distribution

Zheng Wang, *Member, IEEE,* Cong Ling, *Member, IEEE,* Shi Jin, *Senior Member, IEEE,* Yongming Huang, *Senior Member, IEEE* and Feifei Gao, *Fellow, IEEE*

***Abstract*—In this paper, a deterministic sampling decoding strategy for multiple-input multiple output (MIMO) systems is studied, which performs probabilistic searching according to a probability threshold in the lattice Gaussian distribution. Motivated by model probabilistic twin (MPT), the randomness in obtaining the target decoding solution is overcome by the proposed probabilistic searching decoding (PSD) algorithm, which brings considerable decoding gains in both performance and complexity. Specifically, the decoding radius of PSD is derived while the decoding complexity in terms of the number of visited nodes during the searching is also upper bounded, leading to an explicit decoding trade-off. Meanwhile, we generalize PSD by the mechanism of candidate protection so that it enjoys a flexible performance between the suboptimal successive interference cancelation (SIC) decoding and the optimal maximum likelihood (ML) decoding by adjusting the initial search size $K$. Methods for further optimization and complexity reduction of the proposed PSD algorithm are also given. Finally, simulation results based on MIMO detection are presented to confirm the tractable and flexible decoding trade-off of the proposed PSD algorithm.**

**Keywords:** MIMO detection, lattice Gaussian distribution, model probabilistic twin, massive MIMO system, sampling decoding, sphere decoding, near-ML decoding.

## I. Introduction

As one of the core problems of lattice decoding, the closest vector problem (CVP) has wide applications in signal detection for multiple-input multiple output (MIMO) communications. However, the dramatically increased system size also places a pressing challenge upon solving the CVP in MIMO detection. On one hand, the conventional decoding schemes like lattice-reduction-aided decoding show a substantial performance loss with the increment of system dimension [1]–[5]. On the other hand, a number of maximum-likelihood (ML) decoding schemes that aim to reduce the computational complexity of sphere decoding (SD) turn out to be impractical due to their unaffordable complexity in large-scale systems [6]–[9]. As for those near-ML decoding schemes like fixed-complexity sphere decoding (FCSD), K-best decoder, etc., they are also inapplicable due to the intensive complexity increment and significant performance degradation [10]–[13]. In this condition, a number of competitive decoding schemes have been proposed to either improve the performance or lower the complexity [14]–[18]. Among them, sampling decoding has become a promising one, which performs lattice decoding by sampling from a discrete multidimensional Gaussian distribution [19]–[22].

Typically, sampling decoding converts the conventional decoding problem into a sampling problem, where the optimal decoding solution with the smallest Euclidean distance naturally entails the largest probability to be sampled. However, in sharp contrast with continuous Gaussian density, it is by no means trivial even for sampling from a low-dimensional discrete Gaussian distribution, which means that sampling decoding chiefly relies on how to successfully sample over the target lattice Gaussian distribution (LGD). For this reason, the pioneer works of sampling decoding only perform the sampling over a discrete Gaussian-like distribution [23]–[25]. On the other hand, the classic Markov chain Monte Carlo (MCMC) methods were introduced to perform the exact sampling though the mixing of the Markov chains [26]–[28]. Moreover, in [29], the independent Metropolis-Hastings-Klein (IMHK) sampling algorithm with accessible convergence rate was given and was further adopted to sampling decoding in [30], thus leading to a tractable sampling decoding by adjusting the number of Markov moves. Nevertheless, sampling decoding suffers from inherent randomness during the sampling. On one hand, the possibility of missing the optimal decoding solution does always exist, rendering performance loss inevitable. On the other hand, because of the independent and identically distributed (i.i.d.) samplings, considerable computational complexity is spent in unnecessarily repeating the same calculations.

In this paper, to overcome the randomness during the sampling decoding, a deterministic sampling decoding strategy based on the lattice Gaussian distribution is studied and the probabilistic searching decoding (PSD) algorithm is proposed. Specifically, since the optimal decoding solution has the largest probability in the unimodal lattice Gaussian distribution, if candidate vectors with probabilities larger than a probability threshold can be deterministically obtained, then the optimal decoding solution will be easily obtained. On the other hand, although Klein mentioned a deterministic scheme very briefly in [23], it does not seem to allow for an efficient implementation. Meanwhile, the heuristic implementation of deterministic sampling in [25] is hard to characterize the decoding trade-

This work was supported in part by the National Natural Science Foundation of China under Grant 61801216, 61720106003, 62225107, in part by the National Key R&D Program of China under Grant 2018YFB1800801, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420, in part by the Fundamental Research Funds for the Central Universities 2242022k60002.

Z. Wang, S. Jin and Y. Huang are with National Mobile Communications Research Laboratory, and School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: z.wang@ieee.org); C. Ling is with the Department of Electrical and Electronic Engineering, Imperial College London, London, SW7 2AZ, United Kingdom; F. Gao is with the Department of Automation, Tsinghua University, Beijing 100084, China.

off in theory. To summarize, our work in this paper can be outlined in the following several fronts.

First of all, based on lattice Gaussian distribution, the concept of model probabilistic twin (MPT) is given, where extra degrees of freedom in decoding CVP can be achieved. Motivated by it, the probabilistic searching decoding (PSD) algorithm is proposed for a better decoding trade-off by overcoming the randomness within sampling operations. Specifically, based on the designed probabilistic searching threshold, PSD algorithm recursively performs the searching at each decoding layer in a backwards order, where its decoding trade-off between performance (in terms of decoding radius $d$[1]) and complexity (in terms of the number of visited nodes $|S|$) is derived. Secondly, the mechanism of candidate protection is introduced to upgrade the proposed PSD algorithm. This generalizes PSD to a bounded distance decoding (BDD) scheme, where flexible performance between suboptimal decoding and ML decoding can be achieved by simply adjusting the initial search size $K \geq 1$. We also show that PSD with candidate protection still follows the same complexity upper bound, leading to a tractable and controllable decoding process. Finally, the optimization of standard deviation over the finite state space is presented to strengthen the decoding performance, and a method for complexity reduction of PSD is also proposed which incurs a negligible loss in decoding performance.

The rest of this paper is organized as follows. Section II introduces lattice decoding, and briefly reviews the basics of lattice Gaussian sampling. In Section III, motivated by model probabilistic twin, the probabilistic searching decoding (PSD) algorithm is proposed, followed by a comprehensive analysis of both decoding performance and complexity. Meanwhile the mechanism of candidate protection is also designed for PSD to enjoy a flexible decoding trade-off between suboptimal and optimal ML performance. In Section IV, further optimization and complexity reduction with respect to PSD are given. Simulation results for MIMO detection are shown in Section V. Finally, Section VI concludes the paper.

*Notation:* Matrices and column vectors are denoted by upper and lowercase boldface letters, and the transpose, inverse, pseudoinverse of a matrix $\mathbf{B}$ by $\mathbf{B}^T, \mathbf{B}^{-1}$, and $\mathbf{B}^\dagger$, respectively. We use $\mathbf{b}_i$ for the $i$th column of the matrix $\mathbf{B}$, $b_{i,j}$ for the entry in the $i$th row and $j$th column of the matrix $\mathbf{B}$. Finally, the computational complexity is measured by the number of arithmetic operations (additions, multiplications, comparisons, etc.).

## II. Preliminaries

In this section, we introduce the background and mathematical tools needed to describe and analyze the proposed PSD algorithm based on model probabilistic twin.

[1]Stemming from the concept of proximity factor in [5], decoding radius $d$ serves as an effective metric to evaluate the decoding performance of various lattice decoding schemes [24], [25], [30], [31].

**Algorithm 1** Klein's Sampling Algorithm

**Input:** $\mathbf{B}, \sigma, \mathbf{c}$
**Output:** $\mathbf{Bx} \in \Lambda$
1: let $\mathbf{B} = \mathbf{QR}$ and $\mathbf{y} = \mathbf{Q}^\dagger \mathbf{c}$
2: **for** $i = n, \ldots, 1$ **do**
3: let $\sigma_i = \frac{\sigma}{|r_{i,i}|}$ and $\widetilde{x}_i = \frac{y_i - \sum_{j=i+1}^{n} r_{i,j} x_j}{r_{i,i}}$
4: sample $x_i$ from $D_{\mathbb{Z},\sigma_i,\widetilde{x}_i}$
5: **end for**
6: return $\mathbf{Bx}$

### A. Lattice Decoding

Given the full $n \times n$ column-rank matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, the $n$-dimensional lattice $\Lambda$ generated by it is defined by

$$\Lambda = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}, \tag{1}$$

where $\mathbf{B}$ is called the lattice basis. Here we consider the decoding of an $n \times n$ real-valued system; the extension to the complex-valued system is straightforward [24], [32]. In the considered MIMO systems, let $\mathbf{x} \in \mathbb{Z}^n$ denote the transmitted signal, then the corresponding received signal $\mathbf{c}$ is given by

$$\mathbf{c} = \mathbf{Bx} + \mathbf{w} \tag{2}$$

where $\mathbf{w}$ is the noise vector with zero mean and variance $\sigma_w^2$. Typically, the conventional maximum likelihood (ML) decoding reads

$$\widehat{\mathbf{x}}_{\text{ML}} = \arg\min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{Bx} - \mathbf{c}\|^2 \tag{3}$$

where $\|\cdot\|$ denotes Euclidean norm. Clearly, the ML decoding in MIMO systems corresponds to CVP in lattice decoding [9].

Here, for notational simplicity and better presentation, QR-decomposition with $\mathbf{B} = \mathbf{QR}$ is applied and we express the system model in (2) as

$$\mathbf{y} = \mathbf{Q}^T \mathbf{c} = \mathbf{Rx} + \mathbf{n}, \tag{4}$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix. Accordingly, the ML decoding in (3) becomes

$$\widehat{\mathbf{x}}_{\text{ML}} = \arg\min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{Rx} - \mathbf{y}\|^2. \tag{5}$$

In the classic Babai's nearest plane algorithm, $\widehat{x}_i$ is decoded in a backwards order layer by layer (i.e., $i = n, n-1, \ldots, 1$) by direct rounding

$$\widehat{x}_i = \lceil \widetilde{x}_i \rfloor, \tag{6}$$

where

$$\widetilde{x}_i = \frac{y_i - \sum_{j=i+1}^{n} r_{i,j} \widehat{x}_j}{r_{i,i}}. \tag{7}$$

In the scenario of MIMO detection, this can be interpreted as successive interference cancelation (SIC) detection [25].

Note that different from the low-complexity decoding schemes designed for cases $\mathbf{B} \in \mathbb{R}^{n \times m}$, $n > m$ [33]–[36], no extra receive diversity can be exploited under the case of $\mathbf{B} \in \mathbb{R}^{n \times n}$ (i.e., $n = m$). Therefore, with $\mathbf{B} \in \mathbb{R}^{n \times n}$, there is a substantial performance gap between the linear-based low complexity decoding schemes and ML decoding especially in high-dimensional MIMO systems. Since we are seeking for a

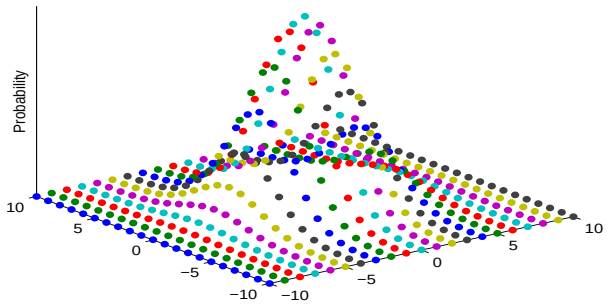


Fig. 1. Illustration of a two-dimensional lattice Gaussian distribution.

better decoding scheme with flexible and tractable decoding trade-off between performance and complexity, the case of an $n \times n$ MIMO system is considered throughout the context. Nevertheless, the proposed decoding algorithm as well as the related analysis can be easily adopted to the case $n \times m$ (i.e., massive MIMO system) via the following transformation

$$\begin{aligned}\mathbf{c} = \mathbf{B}\mathbf{x} + \mathbf{w} &= \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}_{(n-m)\times m}\end{bmatrix}\mathbf{x} + \mathbf{w}\\ &= [\mathbf{Q}_1\ \mathbf{Q}_2]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}_{(n-m)\times m}\end{bmatrix}\mathbf{x} + \mathbf{w}, \end{aligned} \tag{8}$$

where $\mathbf{Q} = [\mathbf{Q}_1\ \mathbf{Q}_2]$ is an $n \times n$ orthogonal matrix and $\mathbf{R}$ is an $m \times m$ upper triangular matrix. The matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ represent the first $m$ and last $n-m$ orthogonal columns of $\mathbf{Q}$, respectively. Then, via a simple transformation, the system model shown above becomes

$$\mathbf{Q}^T\mathbf{c} = \mathbf{Q}^T\mathbf{B}\mathbf{x} + \mathbf{Q}^T\mathbf{w} = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}_{(n-m)\times m}\end{bmatrix}\mathbf{x} + \mathbf{Q}^T\mathbf{w}, \tag{9}$$

which can be further reformatted as

$$\mathbf{y} = \mathbf{Q}_1^T\mathbf{c} = \mathbf{R}\mathbf{x} + \mathbf{n} \tag{10}$$

with $\mathbf{n} = \mathbf{Q}_1^T\mathbf{w}$. Clearly, by doing this, the decoding of an $n \times m$ system ($n > m$) is reduced to decoding of an $m \times m$ system, and the proposed decoding algorithm can be applied thereafter.

### B. Lattice Gaussian Sampling

Given the lattice $\Lambda = \{\mathbf{R}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\}$, define the Gaussian function centered at $\mathbf{y} \in \mathbb{R}^n$ for standard deviation $\sigma > 0$ as

$$\rho_{\sigma,\mathbf{y}}(\mathbf{z}) = e^{-\frac{\|\mathbf{z}-\mathbf{y}\|^2}{2\sigma^2}}, \tag{11}$$

for all $\mathbf{z} \in \mathbb{R}^n$. When $\mathbf{y}$ or $\sigma$ are not specified, it is assumed that they are $\mathbf{0}$ and 1 respectively. Then, the discrete Gaussian distribution over lattice $\Lambda$ (i.e., *lattice Gaussian distribution*) is defined as [37]

$$D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{y}}(\mathbf{R}\mathbf{x})}{\rho_{\sigma,\mathbf{y}}(\Lambda)} = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{R}\mathbf{x}-\mathbf{y}\|^2}}{\sum_{\mathbf{x}\in\mathbb{Z}^n} e^{-\frac{1}{2\sigma^2}\|\mathbf{R}\mathbf{x}-\mathbf{y}\|^2}} \tag{12}$$

for all $\mathbf{x} \in \mathbb{Z}^n$, where $\rho_{\sigma,\mathbf{y}}(\Lambda) \triangleq \sum_{\mathbf{R}\mathbf{x}\in\Lambda} \rho_{\sigma,\mathbf{y}}(\mathbf{R}\mathbf{x})$ is a Gaussian scalar to ensure a probability distribution. Due to the central role of the lattice Gaussian distribution (LGD) played in various research fields, sampling from the lattice Gaussian distribution (i.e., *lattice Gaussian sampling*) becomes an important but challenging problem [29], [30], [38], [39].

As an approximation of lattice Gaussian sampling, Klein's sampling algorithm was proposed in [23], which is able to sample from a discrete Gaussian-like distribution. Klein's sampling can be viewed as a statistical variant of Babai's nearest plane algorithm. Specifically, $\widehat{x}_i$ is randomly chosen from the following 1-dimensional conditional lattice Gaussian distribution

$$\widehat{x}_i \sim p(\widehat{x}_i) \triangleq D_{\mathbb{Z},\sigma_i,\widetilde{x}_i}(x_i = \widehat{x}_i) = \frac{e^{-\frac{1}{2\sigma_i^2}\|\widehat{x}_i-\widetilde{x}_i\|^2}}{\sum_{\widehat{x}_i\in\mathbb{Z}} e^{-\frac{1}{2\sigma_i^2}\|\widehat{x}_i-\widetilde{x}_i\|^2}} \tag{13}$$

in a backwards order with $\sigma_i = \frac{\sigma}{|r_{i,i}|}$, which makes the sample $\widehat{\mathbf{x}}$ obey the following Klein's sampling probability

$$\begin{aligned} P_{\text{Klein}}(\widehat{\mathbf{x}}) &= \prod_{i=1}^{n} D_{\mathbb{Z},\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(x_{n-i+1})\\ &= \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{R}\widehat{\mathbf{x}}-\mathbf{y}\|^2}}{\prod_{i=1}^{n}\sum_{\widetilde{x}_{n-i+1}\in\mathbb{Z}} e^{-\frac{1}{2\sigma_{n-i+1}^2}\|x_{n-i+1}-\widetilde{x}_{n-i+1}\|^2}}\\ &= \frac{\rho_{\sigma,\mathbf{y}}(\mathbf{R}\widehat{\mathbf{x}})}{\prod_{i=1}^{n}\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})}. \end{aligned} \tag{14}$$

However, it has been demonstrated in [40] that $P_{\text{Klein}}(\mathbf{x})$ can be close to $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x})$ when $\sigma$ is sufficiently large. Unfortunately, such a requirement is extremely stringent, rendering it inapplicable in many cases of interest. To this end, Markov chain Monte Carlo (MCMC) methods were introduced to lattice Gaussian sampling [27], [29], which randomly generates the next Markov state conditioned on the previous one. In this way, after the mixing time for convergence, the distribution of samples from the Markov chain will be statistically close to the target distribution, where samples from the lattice Gaussian distribution can be obtained thereafter [41].

## III. Probabilistic Searching Decoding Algorithm

In this section, motivated by model probabilistic twin, a deterministic sampling decoding scheme is proposed based on the lattice Gaussian distribution, where decoding gains in both performance and complexity are exploited.

### A. Model Probabilistic Twin

Intuitively, the CVP given in (3) can be solved by lattice Gaussian sampling. Since the distribution is centered at the query point $\mathbf{c}$, the closest lattice point $\mathbf{B}\mathbf{x}$ to $\mathbf{c}$ (i.e., equivalent to $\|\mathbf{R}\mathbf{x}-\mathbf{y}\|$) in ML decoding criterion also accounts for the largest probability in the lattice Gaussian distribution, i.e.,

$$\widehat{\mathbf{x}}_{\text{ML}} = \arg\min_{\mathbf{x}\in\mathbb{Z}^n} \|\mathbf{c}-\mathbf{B}\mathbf{x}\|^2 = \arg\max_{\mathbf{x}\in\mathbb{Z}^n} D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}), \tag{15}$$

where such an equivalent paradigm transformation is named as *model probabilistic twin* (MPT).

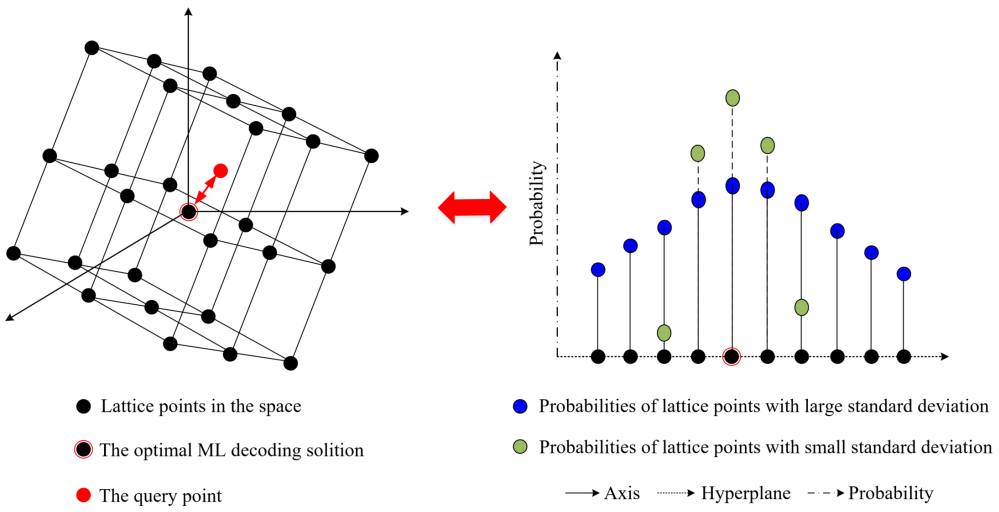


Fig. 2. The paradigm transformation of solving the 3-dimensional decoding problem from the distance in Euclidean space to probability in lattice Gaussian distribution.

According to model probabilistic twin in (15), the decoding problem in (3) can be viewed as a sampling problem. By multiple samplings, the optimal decoding solution is most likely to be returned due to its largest sampling probability. It has been demonstrated that lattice Gaussian sampling is equivalent to CVP via a polynomial-time dimension-preserving reduction [42]. Most importantly, the standard deviation $\sigma > 0$ is introduced by model probabilistic twin, which brings extra degrees of freedom in addressing CVP. Typically, it is clear to see that model probabilistic twin always holds regardless of the value of $\sigma$. This means $\sigma$ is flexible to choose for serving the efficient decoding. To make it clear, an illustration of the paradigm transformation with respect to solving a 3-dimensional decoding problem is shown in Fig. 2. Interestingly, one can observe that a small $\sigma$ significantly increases the probability of the target ML decoding solution $D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}_{\text{ML}})$ (i.e., $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}})$) making $\mathbf{x}_{\text{ML}}$ more likely to be returned. This effectively relieves the *curse of dimensionality* in lattice decoding so that lattice Gaussian sampling with small $\sigma$ turns out to be promising especially in high-dimensional systems [31], [43]–[45].

Unfortunately, sampling decoding based on the lattice Gaussian distribution suffers from inherent randomness. On one hand, although the probability of obtaining $\mathbf{x}_{\text{ML}}$ can be improved by increasing the number of samplings, the probability of missing $\mathbf{x}_{\text{ML}}$ always exists, which results in inevitable decoding performance loss. On the other hand, multiple independent samplings require considerable computational cost but a large number of samples can result in repetition, and an associated increase in complexity.

### B. Algorithm Description

In accordance with the lattice Gaussian distribution, one only needs to pay attention to those candidate vectors with probabilities larger than a probability threshold, namely, $\mathbf{x} \in L_{\text{LGD}}$ and

$$L_{\text{LGD}} = \{\mathbf{x} \in \mathbb{Z}^n : D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}) \geq P_{\text{threshold}}\}. \tag{16}$$

Because the lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x})$ is unimodal, the ML decoding solution $\mathbf{x}_{\text{ML}}$ will be obtained for sure if $P_{\text{threshold}} \leq D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}})$, namely, $\mathbf{x}_{\text{ML}} \in L_{\text{LGD}}$. In this way, the randomness during the sampling can be avoided while how to design a deterministic sampling scheme to efficiently collect these candidate vectors belong to $L_{\text{LGD}}$ becomes the key. However, this turns out to be quite challenging as the Gaussian scalar $\rho_{\sigma,\mathbf{y}}(\Lambda) > 0$ in the lattice Gaussian distribution is difficult to factorize or compute. For this reason, the probabilistic searching decoding (PSD) algorithm is proposed, which applies Klein's sampling probability as an approximation of the lattice Gaussian distribution, i.e.,

$$L = \{\mathbf{x} \in \mathbb{Z}^n : P_{\text{Klein}}(\mathbf{x}) \geq P_{\text{threshold}}\}. \tag{17}$$

Specifically, the proposed PSD algorithm adopts a tree-search structure, where recursive searching is performed layer by layer in a backwards order from $i = n$ to $i = 1$. To concisely state the operations, the following definitions based on the tree-search structure are made.

Define the *initial search size* $K > 1, K \in \mathbb{R}$, which is set initially to control decoding performance and complexity. Meanwhile, given $\widetilde{x}_i$ in (7), let $\widehat{x}_i^j$ denote the $j$th closest integer candidate node to $\widetilde{x}_i$. Accordingly, the *search size* $K(\widehat{x}_i^j) > 0, K(\widehat{x}_i^j) \in \mathbb{R}$ for each integer candidate node $\widehat{x}_i^j$ is defined as

$$K(\widehat{x}_i^j) \triangleq K(\underline{\widehat{x}_i^j}) \cdot p(\widehat{x}_i^j) \tag{18}$$

with defined *normalized probability*

$$p(\widehat{x}_i^j) \triangleq \frac{e^{-\frac{1}{2\sigma_i^2}\|\widehat{x}_i^j - \widetilde{x}_i\|^2}}{\rho_{\sigma_i,\widetilde{x}_i}(\mathbb{Z})}, \tag{19}$$

where

$$\rho_{\sigma_i,\widetilde{x}_i}(\mathbb{Z}) = \sum_{\widehat{x}_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma_i^2}\|\widehat{x}_i - \widetilde{x}_i\|^2}. \tag{20}$$

Here, $\underline{\widehat{x}_i^j}$ indicates the parent node of $\widehat{x}_i^j$ at the previous searching layer $i+1$. It is easy to check that the initial search

size $K = K(\underline{\widehat{x}_n^j})$. Intuitively, via the normalized probability in (18), the search size $K(\underline{\widehat{x}_i^j})$ of a parent node is reasonably allocated to its children nodes. Note that several children candidate nodes $\widehat{x}_i^j$ may have the same parent node $\underline{\widehat{x}_i^j}$.

Next, based on the search size $K(\widehat{x}_i^j)$, the integer candidate node $\widehat{x}_i^j$ at layer $i$ will be saved if it satisfies the following *searching threshold*

$$K(\widehat{x}_i^j) \geq 1. \tag{21}$$

Otherwise, the candidate node $\widehat{x}_i^j$ will be pruned while the searching steps into the next layer $i-1$ given those saved candidate nodes. As shown in Fig. 3, candidate nodes $\widehat{x}_{i-1}^1$ and $\widehat{x}_{i-1}^2$ at layer $i-1$ are saved to enable the searching at the next layer. Intuitively, since the normalized probability $p(\widehat{x}_i^j)$ decays exponentially with the index $j$, all the candidate nodes with index $j > 3$ are deterministically pruned if node $\widehat{x}_{i-1}^3$ fails to satisfy the searching threshold.

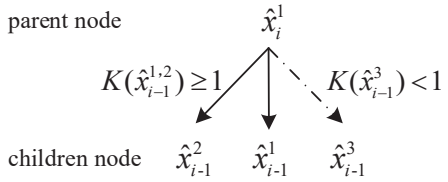


Fig. 3. The illustration of searching threshold, where candidate nodes $\widehat{x}_{i-1}^3$ and $\widehat{x}_{i-1}^{j>3}$ stemmed from $\widehat{x}_i^1$ are pruned.

According to the searching threshold in (21), the searching along the tree structure works layer by layer from $i = n$ to $i = 1$ while the survived decoding candidate vectors (i.e., $\widehat{\mathbf{x}} = [x_1, \ldots, x_n]^T$) are saved by the candidate set $L$.

**Theorem 1.** *The proposed PSD algorithm will return the candidate vectors in the following set*

$$L = \{\mathbf{x} \in \mathbb{Z}^n : P_{\text{Klein}}(\mathbf{x}) \geq P_{\text{threshold}}\}. \tag{22}$$

*Proof:* According to (18), the searching threshold $K(\widehat{x}_i^j) \geq 1$ can be expressed as

$$p(\widehat{x}_i^j) \geq \frac{1}{K(\underline{\widehat{x}_i^j})} = \frac{1}{K \cdot p(\underline{\widehat{x}_{i+1}^j}) \cdots p(\underline{\widehat{x}_n^j})}. \tag{23}$$

From (23), for any candidate vector $\widehat{\mathbf{x}}$ being obtained by PSD, its normalized probability $p(\widehat{x}_1)$ of node $\widehat{x}_1$ at the layer $i = 1$ must satisfy

$$p(\widehat{x}_1) \geq \frac{1}{K \cdot p(\widehat{x}_2) \cdots p(\widehat{x}_n)}, \tag{24}$$

which results in the following lower bound

$$\begin{aligned}\prod_{i=1}^{n} p(\widehat{x}_{n-i+1}) &= \prod_{i=1}^{n} \frac{e^{-\frac{1}{2\sigma_{n-i+1}^2}\|\widehat{x}_{n-i+1}-\widetilde{x}_{n-i+1}\|^2}}{\sum_{\widehat{x}_{n-i+1}\in\mathbb{Z}} e^{-\frac{1}{2\sigma_{n-i+1}^2}\|\widehat{x}_{n-i+1}-\widetilde{x}_{n-i+1}\|^2}} \\ &= \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{R}\mathbf{x}-\mathbf{y}\|^2}}{\prod_{i=1}^{n}\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})} \\ &= P_{\text{Klein}}(\widehat{\mathbf{x}}) \geq \frac{1}{K}. \end{aligned} \tag{25}$$

Here, since the initial search size $K > 1$ is flexible to choose, we can make the following definition

$$P_{\text{threshold}} \triangleq \frac{1}{K}, \tag{26}$$

completing the proof. ∎

Finally, the candidate vector $\widehat{\mathbf{x}}$ with the smallest Euclidean distance $\|\mathbf{R}\widehat{\mathbf{x}}-\mathbf{y}\|$ (i.e., $\|\mathbf{B}\widehat{\mathbf{x}}-\mathbf{c}\|$) among $L$ will be outputted as the decoding solution.

Different from obtaining the candidate vectors with $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}) \geq P_{\text{threshold}}$, PSD returns the candidate vectors with $P_{\text{Klein}}(\mathbf{x}) \geq P_{\text{threshold}}$. Such an approximation can be evaluated in terms of *decoding radius* $d$, which serves as an effective measurement for decoding performance [24], [30]. Theoretically, the targeted decoding solution with $\|\mathbf{R}\mathbf{x}_{\text{target}} - \mathbf{y}\|$ less than the decoding radius will be correctly decoded so that a larger decoding radius leads to a better decoding performance. Here, with $P_{\text{threshold}} = 1/K$, it is easy to verify that the deterministic sampling decoding based on (16) corresponds to enumerating all the candidate vectors within decoding radius

$$d_{\text{LGD}} = \sigma\sqrt{2\ln\frac{K}{\rho_{\sigma,\mathbf{y}}(\Lambda)}} \tag{27}$$

while the decoding radius of the proposed PSD algorithm is

$$d_{\text{PSD}} = \sigma\sqrt{2\ln\frac{K}{\prod_{i=1}^{n}\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})}}, \tag{28}$$

which corresponds to $L_{\text{LGD}} = \{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{R}\mathbf{x} - \mathbf{y}\| \leq d_{\text{LGD}}\}$ and $L = \{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{R}\mathbf{x} - \mathbf{y}\| \leq d_{\text{PSD}}\}$ respectively. Let $d(\Lambda, \mathbf{y})$ represent the Euclidean distance between the query point $\mathbf{y}$ and lattice $\Lambda$ (i.e., $d(\Lambda, \mathbf{y}) = \|\mathbf{R}\mathbf{x}_{\text{ML}}-\mathbf{y}\| = \|\mathbf{B}\mathbf{x}_{\text{ML}}-\mathbf{c}\|$), then the ML decoding solution $\mathbf{x}_{\text{ML}}$ will be obtained by PSD if $d(\Lambda, \mathbf{y}) \leq d_{\text{PSD}}$, and this leads to

$$K \geq \left(\prod_{i=1}^{n}\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})\right) \cdot e^{2\pi d^2(\Lambda,\mathbf{y})/(\min_i^2|r_{i,i}|)}. \tag{29}$$

Clearly, in PSD, the Gaussian scalar $\rho_{\sigma,\mathbf{y}}(\Lambda)$ is approximated by the product $\prod_{i=1}^{n}\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})$ as an alternative solution. Nevertheless, PSD still follows model probabilistic twin and is capable of achieving the optimal ML decoding at the cost of a larger $K$.

### C. Decoding Trade-off Analysis and Comparison

Next, we investigate the decoding trade-off of the proposed PSD algorithm, where its complexity is measured by the number of visited nodes during the searching.

**Lemma 1.** *In the PSD algorithm, for each parent candidate node $\underline{\widehat{x}_i^j}$ with $K(\underline{\widehat{x}_i^j}) \geq 1$, the number of its saved children candidate nodes at decoding layer $i$ satisfies*

$$K_{\text{save}} \leq K(\underline{\widehat{x}_i^j}) \tag{30}$$

*if* $\sigma \leq \min_i |r_{i,i}|/(2\sqrt{\pi})$.

*Proof:* We start the proof by considering the cases of $1 \leq K(\underline{\widehat{x}_i^j}) < 2$ and $K(\underline{\widehat{x}_i^j}) \geq 2$ respectively.

On one hand, based on the searching threshold in (23),

candidate nodes with $1 \le K(\underline{\widehat{x}_i^j}) < 2$ will be saved if

$$p(\widehat{x}_i^j) \ge \frac{1}{K(\underline{\widehat{x}_i^j})} > \frac{1}{2}. \tag{31}$$

Clearly, because of $\sum_j p(\widehat{x}_i^j) = 1$, there is at most one integer candidate node satisfying (31), implying

$$K_{\text{save}} \le 1 \le K(\underline{\widehat{x}_i^j}) \tag{32}$$

no matter what $\sigma > 0$ is.

On the other hand, when $K(\underline{\widehat{x}_i^j}) \ge 2$, according to the searching threshold in (23), the condition shown in (30) holds if and only if the $\lfloor K(\underline{\widehat{x}_i^j})+1 \rfloor$th closest integer candidate node to $\widetilde{x}_i$ is definitely pruned, that is

$$K(\underline{\widehat{x}_i^j}) p(\widehat{x}_i^{\lfloor K(\underline{\widehat{x}_i^j})+1 \rfloor}) < 1. \tag{33}$$

Then, because the distance $|\widehat{x}_i^j - \widetilde{x}_i|$ is bounded by

$$(j-1) \cdot \frac{1}{2} \le |\widehat{x}_i^j - \widetilde{x}_i| \le j \cdot \frac{1}{2}, \tag{34}$$

(33) can be achieved if

$$K(\underline{\widehat{x}_i^j}) \cdot e^{-\frac{1}{8\sigma_i^2}(\lfloor K(\underline{\widehat{x}_i^j})+1 \rfloor - 1)^2} < \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}). \tag{35}$$

Moreover, according to the following relationship [46]

$$\rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}) \ge e^{-\frac{d^2(\mathbb{Z}, \widetilde{x}_i)}{2\sigma_i^2}} \cdot \rho_{\sigma_i}(\mathbb{Z}) \tag{36}$$

with $d(\mathbb{Z}, \widetilde{x}_i)$ denoting the Euclidean distance between $\widetilde{x}_i$ and its closest integer over $\mathbb{Z}$, (35) holds if

$$K(\underline{\widehat{x}_i^j}) \cdot e^{-\frac{1}{8\sigma_i^2}(\lfloor K(\underline{\widehat{x}_i^j})+1 \rfloor - 1)^2} < e^{-\frac{d^2(\mathbb{Z}, \widetilde{x}_i)}{2\sigma_i^2}} \cdot \rho_{\sigma_i}(\mathbb{Z}) \tag{37}$$

is fulfilled. Because of $0 \le d(\mathbb{Z}, \widetilde{x}_i) \le 1/2$, (37) becomes

$$\sigma^2 < \frac{(\lfloor K(\underline{\widehat{x}_i^j}) + 1 \rfloor - 1)^2 - 1}{8 \ln(K(\underline{\widehat{x}_i^j})/\rho_{\sigma_i}(\mathbb{Z}))} \cdot \|\widehat{\mathbf{b}}_i\|^2. \tag{38}$$

Consequently, it is clear to verify that (38) is satisfied when $\sigma \le \min_i |r_{i,i}|/(2\sqrt{\pi})$. Using the *Jacobi theta function* $\vartheta_3$ [47]

$$\vartheta_3(\nu) = \sum_{n=-\infty}^{+\infty} e^{-\pi \nu n^2} \tag{39}$$

we may write

$$\begin{aligned} \rho_{\sigma_i}(\mathbb{Z}) &= \sum_{\widehat{x}_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma_i^2}\|\widehat{x}_i\|^2} = \vartheta_3(|r_{i,i}|^2/2\pi\sigma^2) \\ &\le \vartheta_3(2) = 1.0039 \approx 1 \end{aligned} \tag{40}$$

for $\sigma \le \min_i |r_{i,i}|/(2\sqrt{\pi})$ because $\vartheta_3(\nu)$ is monotone decreasing with $\nu > 0$. ∎

**Lemma 2.** *In the PSD algorithm, for each parent candidate node $\underline{\widehat{x}_i^j}$ with $K(\underline{\widehat{x}_i^j}) \ge 1$, the summation of search sizes of its saved children candidate nodes at decoding layer $i$ is decreasing*

$$\sum_j K(\widehat{x}_i^j) < K(\underline{\widehat{x}_i^j}) \tag{41}$$

if $\sigma \le \min_i |r_{i,i}|/(2\sqrt{\pi})$.

*Proof:* By (18), for each parent candidate node $\underline{\widehat{x}_i^j}$ with $K(\underline{\widehat{x}_i^j}) \ge 1$, the summation of search sizes of its saved children candidate nodes follows

$$\sum_j K(\widehat{x}_i^j) = K(\underline{\widehat{x}_i^j}) \cdot \sum_j p(\widehat{x}_i^j) < K(\underline{\widehat{x}_i^j}) \cdot \sum_{\widehat{x}_i^j \in \mathbb{Z}} p(\widehat{x}_i^j) = K(\underline{\widehat{x}_i^j}). \tag{42}$$

Here, the inequality holds since partial search sizes would be discarded as their children nodes fail to satisfy the searching threshold. ∎

From Lemma 1 and 2, the complexity of the PSD algorithm can be derived as follows.

**Theorem 2.** *In the PSD algorithm, let $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$, the number of visited nodes is upper bounded by*

$$|S| < nK, \tag{43}$$

*and the number of collected candidate vectors is upper bounded by*

$$|L| < K. \tag{44}$$

*Proof:* According to (30), the number of saved candidate nodes at each layer is upper bounded by the summation of search sizes at the previous layer, namely,

$$K_{\text{save}}^{\text{layer } i} = \sum K_{\text{save}} \le \sum K(\underline{\widehat{x}_i^j}) = K_{\text{search size}}^{\text{layer } i+1}. \tag{45}$$

Then, by (41), it is easy to confirm that the summation of search sizes at each layer is decreasing from layer $n$ to 1, i.e.,

$$K_{\text{search size}}^{\text{layer } 1} < \ldots < K_{\text{search size}}^{\text{layer } n} < K_{\text{search size}}^{\text{layer } n+1} = K. \tag{46}$$

Therefore, the number of visited nodes is upper bounded by

$$|S| = \sum_i K_{\text{save}}^{\text{layer } i} \le \sum_i K_{\text{search size}}^{\text{layer } i+1} < nK. \tag{47}$$

Moreover, since the number of collected searching candidates $|L|$ accounts for $K_{\text{save}}^{\text{layer } 1}$, it is upper bounded by

$$|L| < K, \tag{48}$$

thus completing the proof. ∎

Based on Theorems 1 and 2, the explicit decoding trade-off of the proposed PSD algorithm between decoding performance and complexity can be derived as follows.

**Theorem 3.** *With $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$, the initial search size $K$ of solving CVP by the PSD algorithm is upper bounded by*

$$K \le e^{\frac{2\pi d^2(\Lambda, \mathbf{y})}{\min_i^2 |r_{i,i}|}}, \tag{49}$$

*which corresponds to the upper bound on the number of visited nodes*

$$|S| < n \cdot e^{\frac{2\pi d^2(\Lambda, \mathbf{y})}{\min_i^2 |r_{i,i}|}}. \tag{50}$$

*Proof:* Given the Euclidean distance $d(\Lambda, \mathbf{y})$ between the lattice $\Lambda$ and the query point $\mathbf{y}$, then CVP will be solved by setting $d_{\text{PSD}} = d(\Lambda, \mathbf{y})$, which corresponds to

$$\sigma \sqrt{2 \ln \frac{K}{\prod_{i=1}^n \rho_{\sigma_{n-i+1}, \widetilde{x}_{n-i+1}}(\mathbb{Z})}} = d(\Lambda, \mathbf{y}). \tag{51}$$

TABLE I
PERFORMANCE AND COMPLEXITY OF VARIOUS DECODING SCHEMES.

| | Decoding Radius | Complexity | Number of Visited Nodes |
|---|---|---|---|
| Klein Sampling [23] | $\sqrt{\log_n(Ke^{-2})}\cdot\min_i\lvert r_{i,i}\rvert$ | $O(Kn^2)$ | $\lvert S\rvert = nK$ |
| Randomized Sampling [24] | $\sqrt{\log_\varrho(Ke^{-2n/\varrho})}\cdot\min_i\lvert r_{i,i}\rvert,\ \varrho>1$ | $O(Kn^2)$ | $\lvert S\rvert = nK$ |
| Derandomized Sampling [25] | $\sqrt{\log_\rho(2Ke^{-2n/\rho})}\cdot\min_i\lvert r_{i,i}\rvert$ | unbounded | unbounded |
| IMHK Sampling [30] | $\sqrt{(\ln\frac{K}{\log(1/\epsilon)\cdot\prod_{i=1}^n\rho_{\sigma_{n-i+1},\bar{x}_{n-i+1}}(\mathbb{Z})})/(2\pi)}\cdot\min_i\lvert r_{i,i}\rvert$ | $O(Kn^2)$ | $\lvert S\rvert = nK$ |
| PSD based on Klein's sampling probability | $\sqrt{(\ln\frac{K}{\prod_{i=1}^n\rho_{\sigma_{n-i+1},\bar{x}_{n-i+1}}(\mathbb{Z})})/(2\pi)}\cdot\min_i\lvert r_{i,i}\rvert$ | less than $O(Kn^2)$ | $\lvert S\rvert < nK$ |
| PSD based on LGD | $\sqrt{(\ln\frac{K}{\rho_{\sigma,\mathbf{y}}(\Lambda)})/(2\pi)}\cdot\min_i\lvert r_{i,i}\rvert$ | N/A | N/A |

Moreover, by letting $\sigma=\min_i|r_{i,i}|/(2\sqrt{\pi})$, it follows that

$$\begin{aligned}K&=\left(\prod_{i=1}^n\rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})\right)\cdot e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}}\\&\le\left(\prod_{i=1}^n\rho_{\sigma_{n-i+1}}(\mathbb{Z})\right)\cdot e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}}\qquad(52)\\&=\left(\prod_{i=1}^n\vartheta_3(|r_{n-i+1,n-i+1}|^2/2\pi\sigma^2)\right)\cdot e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}}\\&\le\left(\prod_{i=1}^n\vartheta_3(2)\right)\cdot e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}}\\&\approx e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}},\qquad(53)\end{aligned}$$

where the inequality in (52) recalls the following relationship

$$\rho_{\sigma,\mathbf{y}}(\Lambda)\le\rho_\sigma(\Lambda),\qquad(54)$$

and the equality only holds when $\mathbf{y}\in\Lambda$ [37]. Then, using (43), the complexity is upper bounded by

$$|S|<n\cdot e^{\frac{2\pi d^2(\Lambda,\mathbf{y})}{\min_i^2|r_{i,i}|}},\qquad(55)$$

completing the proof. ■

To make it clear, the comparisons over various decoding schemes about decoding radius (i.e., $d$) and the number of visited nodes (i.e., $|S|$) are summarized in Table I. Note that $K$ in sampling decoding serves as the number of sampling times while $K$ in the proposed PSD algorithm is the initial search size. Nevertheless, their decoding radii and complexities can be well characterized by $K$ for a fair comparison. As can be seen clearly, under the same value of $K$, PSD outperforms sampling decoding schemes due to the larger decoding radius and the lower complexity cost, namely, $d_{\text{PSD}}>d_{\text{IMHK, Randomized, Klein}}$ and $|S|_{\text{PSD}}<nK=|S|_{\text{IMHK, Randomized, Klein}}$. More specifically, given the fact that the computational complexity of randomized sampling decoding is $O(Kn^2)$ (i.e., with fixed $nK$ visited nodes), the computational complexity of PSD is much smaller than it (i.e., with $|S|<nK$ visited nodes) by removing the randomness, which is upper bounded by $O(Kn^2)$. To make it more specific, we express the computational complexity of the proposed PSD algorithm as

$$\mathcal{C}_{\text{PSD}}<\mathcal{C}_{\text{RSD}}=O(Kn^2).\qquad(56)$$

Meanwhile, since the number of visited nodes in PSD is upper bounded by $|S|<nK$, we can easily bound the computational complexity per visited node as

$$\mathcal{C}_{\text{per node}}<O(n).\qquad(57)$$

Nevertheless, we have to point out that such a bound is rather loose while the actual complexity per node could be much smaller. This is because the upper bound $|S|<nK$ that we use is quite loose, which can be verified in the simulation results. To this end, finding a more rigorous bound on $\mathcal{C}_{\text{per node}}$ is one of our works in future.

We point out that PSD only performs the decoding based on Klein's sampling probability $P_{\text{Klein}}(\mathbf{x})$ rather than lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x})$. To make it specific, the Gaussian scalar $\rho_{\sigma,\mathbf{y}}(\Lambda)$ with $\sigma=\min_i|r_{i,i}|/(2\sqrt{\pi})$ in MIMO systems is presented in Fig. 4 by Monte Carlo methods, where $\mathbf{x}\in\mathcal{X}^n$ belongs to QAM modulation. As can be seen clearly, due to the Gaussian scalar $\rho_{\sigma,\mathbf{y}}(\Lambda)<1$, great decoding potential can be exploited by deterministic searching over the lattice Gaussian distribution, leading to a much larger decoding radius $d_{\text{LGD}}$ than that of PSD. Accordingly, this corresponds to the complexity upper bound $K\le\rho_{\sigma,\mathbf{y}}(\Lambda)\cdot e^{2\pi d^2(\Lambda,\mathbf{y})/\min_i^2|r_{i,i}|}$ for solving CVP, which is much lower than that of PSD in (49). Note that $\rho_{\sigma,\mathbf{y}}(\Lambda)$ improves along with the increment of SNR. This is because the received signal $\mathbf{y}$ is getting close to the lattice $\Lambda=\mathbf{Rx}$ as the effect of noise is constrained accordingly.

Here, the Lenstra-Lenstra-Lovász (LLL) reduction can be applied as a preprocessing stage for the proposed PSD algorithm, which effectively improves $\min_i|r_{i,i}|$ (i.e., $\min_i\|\widehat{\mathbf{b}}_i\|$) through the matrix transformation (also reducing $\max_i|r_{i,i}|$ at the same time) [5], [48]. Although LLL reduction is applied to increase the decoding radius, it is easy to check that the complexity by means of the number of visited nodes in PSD still obeys the upper bound $|S|<nK$. Similarly, the upper bound $|L|<K$ for the number of collected candidate vectors holds as well, thus leading to a better decoding trade-off. On

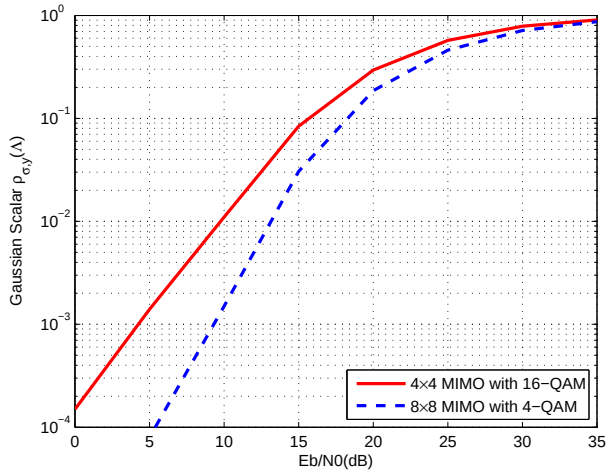


Fig. 4. The Gaussian scalar $\rho_{\sigma,\mathbf{y}}(\Lambda)$ in various uncoded MIMO systems.

the other hand, the computational complexity of LLL reduction is $O(n^3 \log n)$ [49], and the complex version of LLL can be applied for the further complexity reduction [50]. Therefore, LLL reduction can be applied as a preprocessing stage for the proposed PSD algorithm.

### D. Generalization by Candidate Protection

An important question about the proposed PSD algorithm is the choice of the probability threshold $P_{\text{threshold}} = 1/K$, which should be lower than $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}})$ (i.e., $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}}) \geq P_{\text{threshold}}$). Clearly, if $P_{\text{threshold}} = 1/K$ is well chosen, then $\mathbf{x}_{\text{ML}}$ will be efficiently returned. Unfortunately, a precise choice of $P_{\text{threshold}} = 1/K$ is difficult in practice. What's more, PSD only works if $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}}) \geq P_{\text{threshold}}$ while no eligible candidate vector will be outputted if $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}}) < P_{\text{threshold}}$, rendering it only eligible for a sufficiently large search size $K$. Undoubtedly, this severely restricts the applications of PSD. To this end, we generalize PSD with arbitrary choice of $K \geq 1$ (i.e., $1 \geq P_{\text{threshold}} > 0$) via the designed mechanism named as *candidate protection*, where flexible performance between suboptimal decoding (i.e., $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}}) < P_{\text{threshold}}$) and optimal decoding (i.e., $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x}_{\text{ML}}) \geq P_{\text{threshold}}$) can be achieved by adjusting $K$.

Specifically, among candidate nodes with small search size $K(\widehat{x}_i^j)$, candidate protection aims to rescue the most valuable candidate vector along the searching branch, and the searching solution consists of the closest candidate nodes $\widehat{x}_{i's}^1$ in the rest of layers tends to be the most reliable choice. Specifically, as for candidate node $\widehat{x}_i^j$ with small search size

$$2 > K(\widehat{x}_i^j) \geq 1, \tag{58}$$

candidate protection is activated to obtain the closest integer nodes $\widehat{x}_{i-1}^1, \ldots, \widehat{x}_1^1$ in the rest of searching layers, which directly yields a candidate vector $\widehat{\mathbf{x}}$:

$$\widehat{\mathbf{x}} = [\overbrace{\underbrace{\widehat{x}_1^1, \ldots, \widehat{x}_{i-1}^1}_{\text{candidate protection } 2>K(\cdot)\geq 1}, \widehat{x}_i^j, \underbrace{\widehat{x}_{i+1}^j, \ldots, \widehat{x}_n^j}_{K(\cdot)\geq 2}}^{\longleftarrow \text{decoding order}}]^T. \tag{59}$$

For a better understanding, Fig. 5 illustrates the operations of candidate protection.

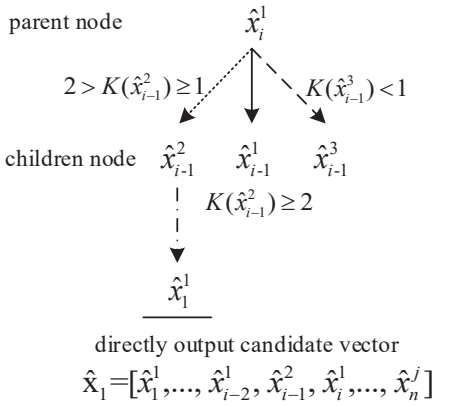


Fig. 5. The illustration of candidate protection, where node $\widehat{x}_{i-1}^2$ invokes candidate protection to directly output a candidate vector $\widehat{\mathbf{x}}$ to set $L$.

We point out that the searching threshold $K(\widehat{x}_i^j) \geq 1$ is smoothly compatible with candidate protection as the latter tries to activate a few candidate nodes discarded by the former. Intuitively, the proposed candidate protection extends the initial search size from $K > 1$ to $K \geq 1$, and it is easy to verify that the decoding performance of Babai's nearest plane algorithm (i.e., SIC) will be achieved when $K = 1$. More specifically, candidate protection can be simply carried out through Babai's nearest plane algorithm since $[\widehat{x}_1^1, \ldots, \widehat{x}_n^1]^T$ is just the decoding result of it.

**Remark 1.** *For PSD with candidate protection, flexible decoding performance can be achieved from Babai's nearest plane algorithm (i.e., $K = 1$) and ML decoding (i.e., $K = e^{2\pi d^2(\Lambda,\mathbf{y})/\min_i^2 |r_{i,i}|}$).*

Another way to evaluate the decoding performance of the proposed PSD algorithm is based on the decoding radius $d_{\text{PSD}}$. In particular, the gain in squared decoding radius of PSD over Babai's nearest plane algorithm is defined as follows in [24]:

$$\begin{aligned} G \triangleq \frac{d_{\text{PSD}}^2}{d_{\text{SIC}}^2} &= \frac{2}{\pi} \cdot \ln \frac{K}{\prod_{i=1}^n \rho_{\sigma_{n-i+1},\widetilde{x}_{n-i+1}}(\mathbb{Z})} \\ &\geq \frac{2}{\pi} \cdot \ln \frac{K}{\prod_{i=1}^n \rho_{\sigma_{n-i+1}}(\mathbb{Z})} \approx \frac{2}{\pi} \cdot \ln K \end{aligned} \tag{60}$$

for $\sigma \leq \min_i |r_{i,i}|/(2\sqrt{\pi})$, where the decoding radius of Babai's nearest plane algorithm is known as $d_{\text{SIC}} = \min_i |r_{i,i}|/2$. Therefore, we have $K \leq e^{\frac{\pi}{2}G}$, which reveals the trade-off between $K$ and $G$. For fixed performance gain $G$, the PSD algorithm has polynomial complexity with respect to the system dimension $n$, which offers a valuable way to guide the choice of $K$.

To summarize, at each searching layer, PSD with candidate protection operates in the following two steps:

- *Calculate the search size $K(\widehat{x}_i^j)$ by (18).*
- *Obtain candidate nodes $\widehat{x}_i^j$ by (21). If $2 > K(\widehat{x}_i^j) \geq 1$, invoke Babai's nearest plane algorithm to directly return a decoding candidate vector $\widehat{\mathbf{x}}$.*

Overall, an illustration of the proposed PSD algorithm is

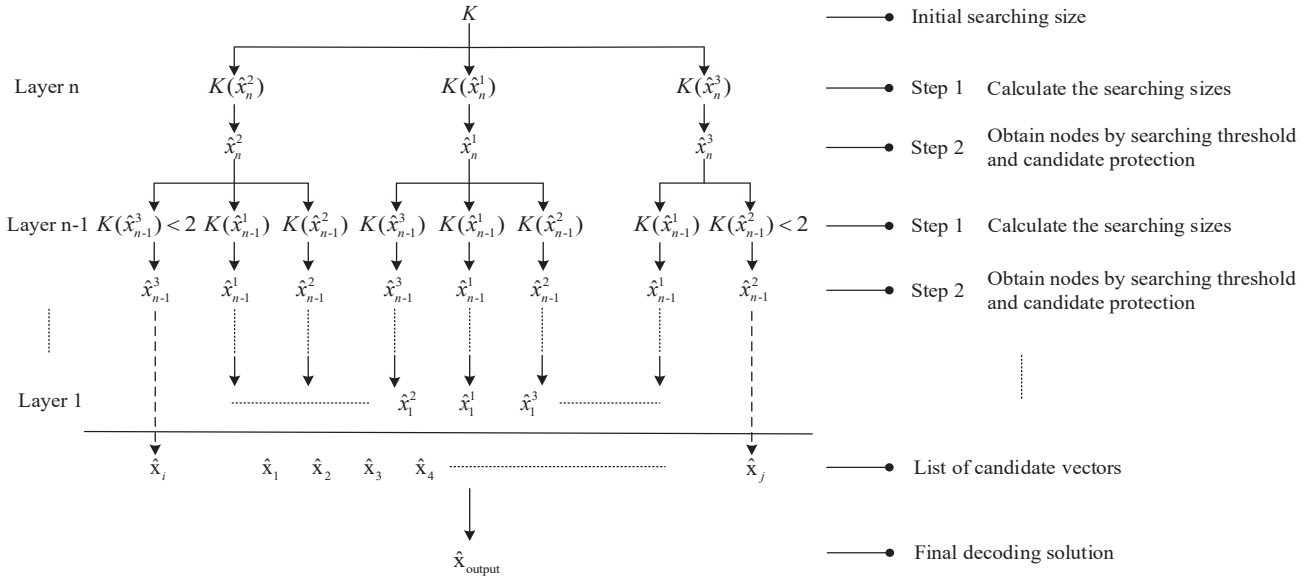


Fig. 6. Illustration of the proposed PSD algorithm with normalized probability and candidate protection, where $K(\widehat{x}_i^j) \geq 1$. The dashed lines stemmed from $K(\widehat{x}_i^j) < 2$ denote the closest candidate nodes $\widehat{x}_{i-1}^1, \ldots, \widehat{x}_1^1$ in the rest of the layers, which are retained to directly yield a decoding candidate vector $\widehat{\mathbf{x}}$.

presented in Fig. 6 with more details. Furthermore, the proposed PSD algorithm is outlined in Algorithm 2 for a better understanding.

Interestingly, even with candidate protection, the complexity $|S|$ as well as the number of collected candidate vectors $|L|$ in PSD still maintains the same upper bound as before.

**Theorem 4.** *Given the initial search size $K \geq 1$, the number of candidate vectors collected by PSD with candidate protection is upper bounded by*

$$|L| < K \tag{61}$$

*with the number of visited nodes bounded by*

$$|S| < nK \tag{62}$$

*for $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$.*

*Proof:* Theoretically, the collected candidate vectors $\widehat{\mathbf{x}}$ come from searching threshold and candidate protection respectively. For notational simplicity, here we represent the search size $K(x_i^j)$ in two different ways: $2 > K(x_i^{\text{protection}}) \geq 1$ and $K(x_i^{\text{searching}}) \geq 2$.

In particular, the summation of the search sizes at each layer is decreasing, which can be expressed as

$$\begin{aligned} K = K(\underline{x_n^j}) &> \sum K(x_n^{\text{protection}}) + \sum K(x_n^{\text{searching}}) \\ &> \sum K(x_n^{\text{protection}}) + \sum K(x_{n-1}^{\text{protection}}) + \sum K(x_{n-1}^{\text{searching}}) \\ &> \cdots \\ &> \sum_{i=2}^{n} \left[ \sum K(x_i^{\text{protection}}) \right] + \sum K(x_2^{\text{searching}}). \end{aligned} \tag{63}$$

Based on candidate protection, only one decoding candidate vector will be saved for each $K(x_i^{\text{protection}})$, $2 \leq i \leq n$, which means the number of collected candidate vectors generated by candidate protection from searching layer $n$ to 2 is bounded by

$$|L_{\text{protection}}| \leq \sum_{i=2}^{n} \left[ \sum K(x_i^{\text{protection}}) \right]. \tag{64}$$

Besides, the number of candidate vectors survived from the searching threshold corresponds to the number of saved candidate nodes at layer $i = 1$, i.e., $K_{\text{save}}^{\text{layer }1}$, which is upper bounded by

$$|L_{\text{searching}}| = K_{\text{save}}^{\text{layer }1} \leq \sum K(x_2^{\text{searching}}) \tag{65}$$

according to (45). Therefore, based on (63), (64) and (65), we have

$$|L| = |L_{\text{searching}}| + |L_{\text{protection}}| < K. \tag{66}$$

Consequently, as all the visited nodes are taken into account to generate $|L|$ decoding candidate vectors, the number of visited nodes is bounded as

$$|S| < n|L| < nK, \tag{67}$$

completing the proof. ■

Note that the initial search size $K$ only offers a complexity upper bound (i.e., $|S| < nK$) while the real complexity could be much less than this. More importantly, the decoding performance of PSD in terms of decoding radius can be explicitly evaluated along with the increment of complexity, which is meaningful especially in high-dimensional systems.

Regarding the complexity, as shown in Algorithm 2, given a sufficient search size $K$, the number of elementary operations (additions, subtractions, and multiplications) for calculating (7), (13) and (18) for a visited node $\widehat{x}_i^j$ at searching layer $1 \leq i \leq n$ are $2(n-i)+1$, 15 and 1 respectively, which leads to $2(n-i)+17$ in total. After that, the judgement will be made about whether to save the node $\widehat{x}_i^j$ based on the value of $K(\widehat{x}_i^j)$. Here, $j = 3$ is considered in the calculation while the demonstration later in Section IV will show it is sufficient. Note that for a small size $K$, the mechanism of

**Algorithm 2** Probabilistic Searching Decoding Algorithm

**Input:** $K, \mathbf{R}, \mathbf{y}, \sigma = \min_i |r_{i,i}|/(2\sqrt{\pi}), L = \emptyset$
**Output:** $\mathbf{R}\mathbf{x} \in \Lambda$

invoke **Function 1** with $i = n$ to decode layer by layer
add all the candidates $\widehat{\mathbf{x}}$'s generated by **Function 1** to $L$
output $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}\in L} \|\mathbf{y} - \mathbf{R}\mathbf{x}\|$ as the decoding solution

candidate protection will be invoked, which means a lower number of elementary operations for each visited node. For example, when $K = 1$, the proposed PSD algorithm just accounts for SIC decoding. Therefore, a different search size $K$ corresponds to a different number of elementary operations for each visited node. Nevertheless, we can see that the complexity of each visited node is upper bounded by $2n + 15$ while the total complexity of the proposed PSD algorithm is less than $O(Kn^2)$ according to (56) and (62) in Theorem 4.

Similar to the traditional sphere decoding, the proposed PSD algorithm also employs the tree-search structure. However, the difference between these sphere decoding-based works [7]–[12], [51], [52] and the proposed PSD algorithm can mainly be found in the following two aspects. On one hand, the proposed PSD algorithm performs the enumeration in terms of probability over Klein's sampling probability, which serves as an approximation of the lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{y}}(\mathbf{x})$. For this reason, PSD belongs to a deterministic sampling decoding strategy with respect to the lattice Gaussian distribution. As shown in Table I, it outperforms all the previous sampling decoding schemes with a better decoding trade-off [23]–[25], [30]. On the other hand, the introduced parameters $\sigma$ and $K$ (as shown in (26), $1/K$ denotes the probability threshold) from the lattice Gaussian distribution offer a more interpretable way to reveal the decoding trade-off of PSD, where the clear complexity upper bound is accessible given the decoding radius. To the best of our knowledge, the probabilistic tree-search based sphere decoding algorithms may also carry out the searching based on other probability criteria, but the sampling probability from the lattice Gaussian distribution shown in (12) has not been involved. Meanwhile, compared to these existing methods, a clear mathematical characterization about the performance and complexity is provided by the proposed PSD algorithm, which is flexibly adjusted by the initial search size $K$.

Nevertheless, we point out that PSD is performed in a sequential order from $x_n$ to $x_1$, which does not allow parallel implementation. By contrast, recent works on parallel sphere schemes have been proposed in [51]–[54]. By adopting a massively parallel design into the nonlinear processing, they provide a promising solution for the problem of MIMO detection in 6G [55]. This could be useful for the development of sampling decoding, and how to incorporate it with sampling decoding will be an interesting direction for future work.

## IV. Optimization and Complexity Reduction

In this section, further optimization and complexity reduction methods are given to make the proposed PSD algorithm well suited to the finite state space of $\mathbf{x}$ in implementation.

**Function 1** Searching at layer $i$ given $[\widehat{x}_n, \ldots, \widehat{x}_{i+1}]$

compute $\widetilde{x}_i$ according to (7)
compute probability $p(\widehat{x}_i^j)$ by (13) with $j \in [1, 2, 3]$
compute search size $K(\widehat{x}_i^j)$ according to (18)
**for** each specific integer candidate $\widehat{x}_i^j$ **do**
  **if** $K(\widehat{x}_i^j) < 1$ **then**
  prune $\widehat{x}_i^j$ from the tree-search decoding
  **else**
  save $\widehat{x}_i^j$ to form the decoding result $[\widehat{x}_n, \ldots, \widehat{x}_{i+1}, \widehat{x}_i^j]$
    **if** $2 > K(\widehat{x}_i^j) \geq 1$ **then**
    decode the rest of layers by SIC to get a candidate $\widehat{\mathbf{x}}$
    **else if** $K(\widehat{x}_i^j) \geq 2$ **then**
      **if** $i = 1$ **then**
      output the candidate $\widehat{\mathbf{x}}$
      **else**
      invoke **Function 1** to decode the next layer $i-1$
      **end if**
    **end if**
  **end if**
**end for**

### A. Optimization with respect to $\sigma$

For the proposed PSD algorithm, the choice of the standard deviation $\sigma$ is recommended as $\min_i |r_{i,i}|/(2\sqrt{\pi})$ so that $K \geq 1$ is adjustable to provide a tractable and flexible decoding trade-off. However, the assumption $\mathbf{x} \in \mathbb{Z}^n$ should, in a practical MIMO detection application, be replaced by a finite state space, i.e., $\mathbf{x} \in \mathcal{X}^n$. Therefore, it is possible to further optimize $\sigma$ by this relaxation for a better decoding performance.

Specifically, let $\sigma = \frac{\min_i |r_{i,i}|}{\sqrt{2\log\alpha}}$ with $\alpha > 1$. Then $\alpha$ becomes the parameter to be considered. Moreover, with $\sigma = \frac{\min_i |r_{i,i}|}{\sqrt{2\log\alpha}}$, it has been demonstrated in [23] that

$$\prod_{i=1}^{n} \rho_{\sigma_i,\widetilde{x}_i}(\mathbb{Z}) \leq e^{\frac{2n}{\alpha}(1+O(\alpha^{-3}))}, \tag{68}$$

where the term $O(\alpha^{-3})$ in (68) could be negligible if $\alpha$ is large. Assume $\alpha$ satisfies this weak condition; by relaxation, (28) can be expressed as

$$e^{-\frac{2n}{\alpha}} \cdot \alpha^{-\|\mathbf{R}\mathbf{x}-\mathbf{y}\|^2/\min_i r_{i,i}^2} \geq \frac{1}{K}, \tag{69}$$

which leads to the following decoding radius

$$d_{\text{opt}} = \min_i r_{i,i} \cdot \sqrt{\log_\alpha(Ke^{-2n/\alpha})}. \tag{70}$$

In order to exploit the decoding potential, parameter $\alpha$ can be optimized to maximize the above decoding radius. Hence, setting the derivative of $\log_\alpha(Ke^{-2n/\alpha})$ with respect to $\alpha$ be zero, the optimum $\alpha_{\text{o}}$ given the initial search size $K$ can be determined by

$$K = (e\alpha_{\text{o}})^{2n/\alpha_{\text{o}}}. \tag{71}$$

From (71), it is easy to check that the optimum $\alpha_{\text{o}}$ monotonically decreases with the increment of $K$, which means the choice of $\sigma = \frac{\min_i |r_{i,i}|}{\sqrt{2\log\alpha_{\text{o}}}}$ should be improved with the increase of $K$ as well. Note that such an optimization about $\sigma$ is only a

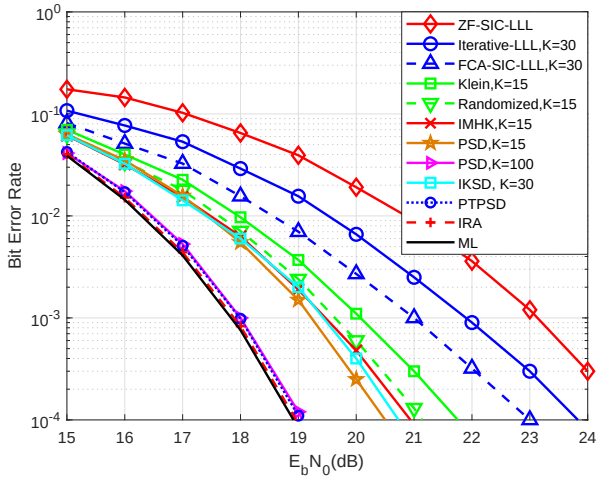


Fig. 7. Bit error rate versus average SNR per bit for the uncoded $12 \times 12$ MIMO system using 64-QAM.

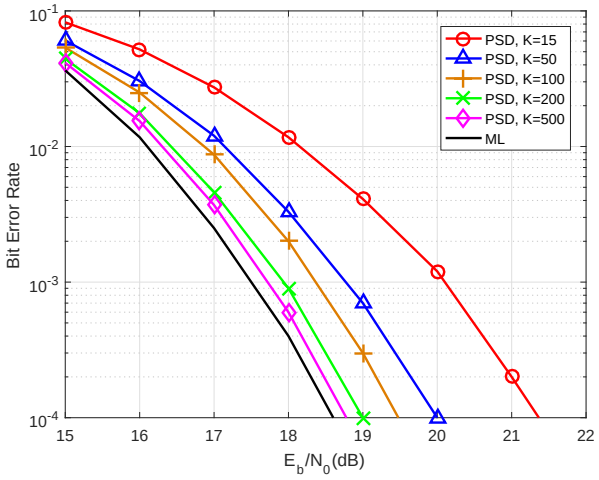


Fig. 8. Bit error rate versus average SNR per bit for the uncoded $16 \times 16$ MIMO system using 64-QAM.

compromise by relaxation, and $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$ is still a better choice for $\mathbf{x} \in \mathbb{Z}^n$.

### B. Complexity Reduction

In principle, sufficient candidate nodes $\widehat{x}_i^j$ with $j = 1, 2, 3, \ldots$ for each parent node $\widehat{x}_i^j$ should be taken into account given the searching threshold. However, in practice, only limited candidate nodes need to be considered, and we now investigate the required size of index $j$.

From (19), the normalized probability of the $j$th candidate node at searching layer $i$ is written as

$$p(x_i^j) = \begin{cases} e^{-\frac{1}{2\sigma_i^2}((j-1)/2+d)^2} / \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}) & \text{when } j \text{ is odd,} \\ e^{-\frac{1}{2\sigma_i^2}(\frac{j}{2}-d)^2} / \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}) & \text{when } j \text{ is even,} \end{cases} \tag{72}$$

where $\frac{1}{2} \geq d = |x_i^1 - \widetilde{x}_i| \geq 0$. Therefore, the summation of the normalized probability of the first $2N$ candidate nodes with respect to $\widetilde{x}_i$ can be expressed as

$$P_{2N} = \sum_{j=1}^{N} \left( e^{-\frac{1}{2\sigma_i^2}(j-1+d)^2} + e^{-\frac{1}{2\sigma_i^2}(j-d)^2} \right) / \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}). \tag{73}$$

Because of $\sum_{\widehat{x}_i^j \in \mathbb{Z}} p(\widehat{x}_i^j) = 1$, with $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$ the normalized probability associated with nodes other than those $2N$ candidate nodes can be derived as

$$\begin{aligned} 1 - P_{2N} &= \sum_{j \geq N+1} \left( e^{-\frac{1}{2\sigma_i^2}(j-1+d)^2} + e^{-\frac{1}{2\sigma_i^2}(j-d)^2} \right) / \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}) \\ &< \sum_{j \geq N+1} 2 \cdot e^{-\frac{1}{2\sigma_i^2}(j-1)^2} / \rho_{\sigma_i, \widetilde{x}_i}(\mathbb{Z}) \\ &< \sum_{j \geq N+1} 2 \cdot e^{-\frac{1}{2\sigma_i^2}[(j-1)^2 - \frac{1}{4}]} / \rho_{\sigma_i}(\mathbb{Z}) \\ &\approx \sum_{j \geq N+1} 2 \cdot e^{-2\pi[(j-1)^2 - \frac{1}{4}]} \\ &= O\left( e^{-2\pi N^2} \right), \end{aligned} \tag{74}$$

which implies that the tail bound (74) decays exponentially fast ($e^{2\pi} \gg 1$).

**Remark 2.** *From (74), with $\sigma = \min_i |r_{i,i}|/(2\sqrt{\pi})$, only limited number of children candidate nodes need to be considered due to the negligible probability $p(x_i^j), j > 3$.*

Therefore, $j = 3$ is recommended for the complexity reduction in the proposed PSD algorithm unless the initial search size $K$ is sufficiently large. This is also well suited to practical cases with finite state space $\mathbf{x} \in \mathcal{X}^n$.

## V. Simulation

In this section, the performance and complexity of the proposed PSD algorithm are evaluated in MIMO detection. Specifically, given the system model in (2), the $i$th entry of the transmitted signal $\mathbf{x}$, denoted as $x_i$, is a modulation symbol taken independently from an $M$-QAM constellation $\mathcal{X}$ with Gray mapping. Meanwhile, we assume a flat fading environment, where the square channel matrix $\mathbf{B}$ contains uncorrelated complex Gaussian fading gains with unit variance and remains constant over each frame duration. Let $E_b$ represents the average energy per bit at the receiver. Then the signal-to-noise ratio (SNR) $E_b/N_0 = n/(\log_2(M)\sigma_w^2)$ where $M$ is the modulation order and $\sigma_w^2$ is the noise variance. Besides, PSD is enhanced by LLL reduction and MMSE augmentation, as well as optimizing $\sigma$ through choosing $\alpha_o$ via (71). As a fair comparison, all the other decoding schemes are also strengthened by LLL reduction. Meanwhile, the sampling decoding schemes are also enhanced by MMSE augmentation [24].

Fig. 7 shows the bit error rate (BER) of the proposed PSD algorithm compared with other decoding schemes in a $12 \times 12$ uncoded MIMO system with 64-QAM. Here, lattice-reduction-aided SIC (i.e., Babai's nearest plane) decoding serves as a performance baseline while ML decoding is implemented by the Schnorr-Euchner (SE) strategy. Meanwhile, the increasing radii algorithm (IRA) in [56], the probabilistic tree pruning sphere decoding (PTPSD) in [57] with pruning probability

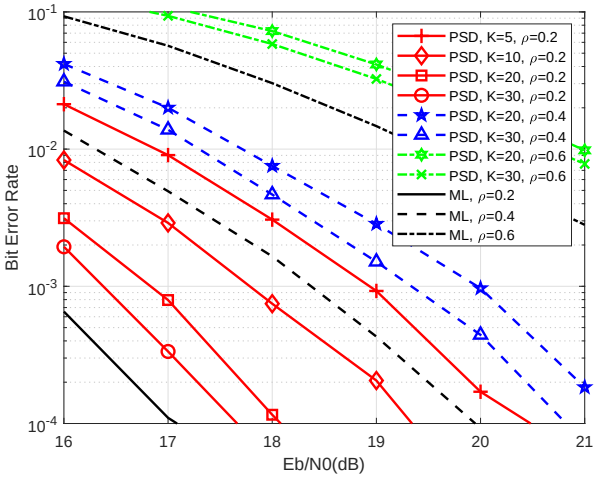


Fig. 9. Bit error rate versus average SNR per bit for the uncoded $10 \times 10$ MIMO system using 16-QAM.

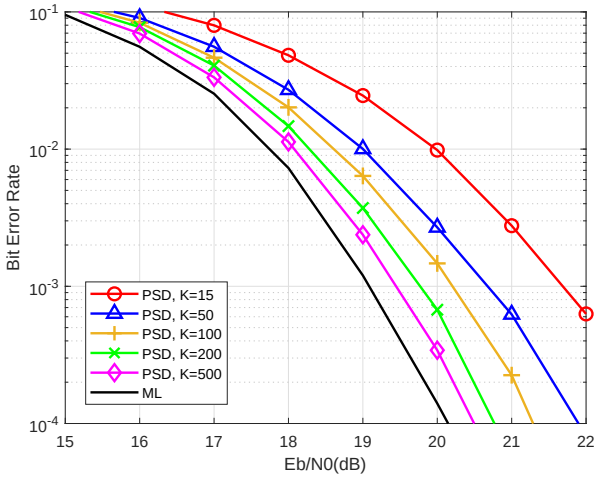


Fig. 10. Bit error rate versus average SNR per bit for the uncoded $16 \times 16$ MIMO system using 64-QAM under local scattering spatial correlation model.

$P_\epsilon = 0.1$ and the improved K-best sphere decoder (IKSD) from [58] are also shown as a comparison. Note that the nonlinear SIC detection has a better decoding performance than these traditional schemes such as ZF and MMSE. Clearly, compared to the fixed candidates algorithm (FCA) in [59] and iterative list decoding in [60] with 30 samples, sampling decoding algorithms such as Klein's sampling decoding [23], randomized sampling decoding [24] and IMHK sampling decoding [30] offer not only the improved BER performance but also the promise of smaller sample size $K$. It is clear that PSD outperforms all the sampling decoding schemes for the same value of $K$, which agrees with the results shown in Table I. More importantly, the complexity cost of PSD is much less than those of sampling decoding schemes, which is illustrated in Fig. 11, Fig. 13 and Fig. 14 in detail. Observe that with $K = 100$, the performance of PSD suffers negligible loss compared with ML. Therefore, with a moderate $K$, near-ML performance can be achieved.

In order to show the performance comparison with different initial search sizes $K$, Fig. 8 is given to illustrate the BER performance of PSD in a $16 \times 16$ uncoded system with 64-QAM. According to (28), a larger $K$ leads to a larger decoding radius, which corresponds to a better decoding performance. More specifically, as shown in (23), a larger $K$ naturally corresponds to a looser searching threshold, which allows more candidate vectors to be obtained. Therefore, as can be seen clearly, with the increment of $K$, the BER performance improves gradually to the ML decoding performance. It is interesting to see that in Fig. 7 near-ML decoding performance can be achieved with $K = 100$ while in Fig. 8 near-ML decoding performance requires $K = 500$. This is because the larger system dimension has a deeper tree-structure to search, which requires a higher initial search size $K$ of the proposed PSD algorithm to explore. Note that according to Theorem 5, the number of visited nodes and the number of collected candidate vectors are upper bounded by $|S| < nK$ and $|L| < K$ respectively, and the complexity increment with respect to $K$ is mild as expected, thus resulting in a promising trade-off between performance and complexity.

Fig. 9 shows the bit error rate (BER) of the proposed PSD algorithm under correlated channels in a $10 \times 10$ uncoded MIMO system with 16-QAM. Specifically, the correlated channel matrix is given by $\mathbf{R}_{\text{cor}}^{\frac{1}{2}}\mathbf{H}\mathbf{T}_{\text{cor}}^{\frac{1}{2}}$. Here, $\mathbf{R}_{\text{cor}} \in \mathbb{C}^{n\times n}$ is the receive correlation matrix, $\mathbf{T}_{\text{cor}} \in \mathbb{C}^{n\times n}$ is the transmit correlation matrix, and $\mathbf{H} \in \mathbb{C}^{n\times n}$ is an independent, identically distributed (i.i.d.) complex Gaussian with zero-mean and unit variance elements. Without loss of generality, we assume in this work that the antennas at both transmitter and receiver sides are equally separated while the correlation matrices $\mathbf{R}_{\text{cor}}$ and $\mathbf{T}_{\text{cor}}$ follow the model in [61], i.e.,

$$\mathbf{R}_{\text{cor}} = \mathbf{T}_{\text{cor}} = \begin{bmatrix} 1 & \rho & \rho^4 & \cdots & \rho^{(n-1)^2} \\ \rho & 1 & \rho & \cdots & \vdots \\ \rho^4 & \rho & 1 & \cdots & \rho^4 \\ \vdots & \vdots & \vdots & \ddots & \rho \\ \rho^{(n-1)^2} & \cdots & \rho^4 & \rho & 1 \end{bmatrix},$$

where the normalized correlation coefficient $\rho$ is employed to adjust the degree of correlation. Note that a totally uncorrelated scenario corresponds to $\rho = 0$ while a fully correlated scenario implies $\rho = 1$. Here, we set $\rho = 0.2$, which results in weak correlated channels. As can clearly be seen, the proposed PSD algorithm works as usual under correlated channels. This is in accordance with the fact that PSD is designed given the random matrix $\mathbf{B}$, making it suitable for various scenarios of MIMO systems. As expected, with the increment of $K$, the detection performance of the proposed PSD improves gradually, implying a flexible and tractable detection trade-off. On the other hand, with the increase of $\rho$, it is clear to see that the underlying channel matrix becomes more correlated so that the detection of both ML and PSD detection continues to deteriorate. Nevertheless, the performance gap between PSD and ML detection still decreases accordingly with the increment of $K$. For a better illustration, the performance curves of PSD with $K = 20, 30$ and ML decoding under

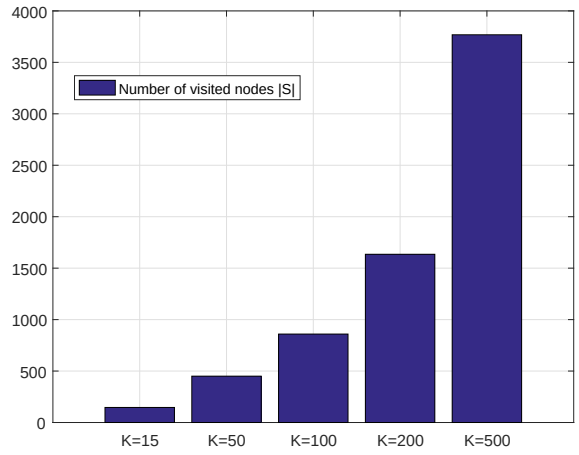


Fig. 11. Number of visited nodes $|S|$ versus initial search size $K$ for $16 \times 16$ uncoded MIMO using 64-QAM at SNR per bit = 17dB.

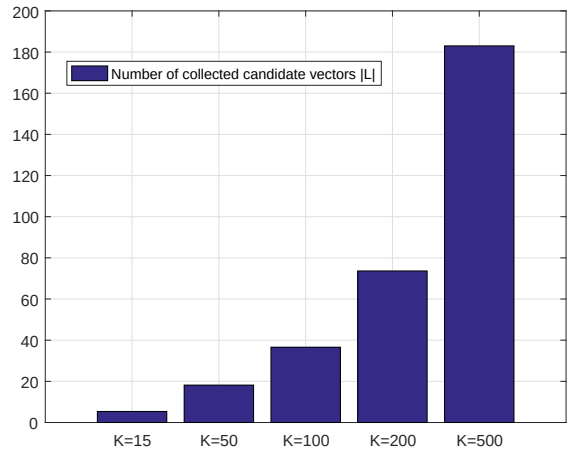


Fig. 12. Number of collected candidate vectors $|L|$ versus initial search size $K$ for $16 \times 16$ uncoded MIMO using 64-QAM at SNR per bit = 17dB.

$\rho = 0.4$ and $\rho = 0.6$ are also added.

To further illustrate the proposed PSD algorithm in correlated channel models, Fig. 10 is presented to show the bit error rate (BER) of the proposed PSD algorithm in a $16 \times 16$ uncoded MIMO system with 64-QAM under the local scattering spatial correlation model in [62]. In particular, in the local scattering spatial correlation model, the uniformly distributed deviations are applied with $\theta \sim U[-\sqrt{3}\sigma_\varphi, \sqrt{3}\sigma_\varphi]$, and we set the nominal angle $\varphi = 10^\circ$ with angular standard deviation (ASD) $\sigma_\varphi = 30^\circ$, where the correlation matrices are normalized with matrix trace equal to $n = 16$. As can be seen clearly in Fig. 10, the BER performance of the proposed PSD algorithm improves gradually with the increment of $K$ due to the larger decoding radius derived in (28), which leads to a flexible detection performance adjusted by $K$. Meanwhile, compared to the i.i.d. Gaussian channel model shown in Fig. 8, we can observe that both the BER performance of ML and PSD degrade accordingly. This is due to the fact that a higher degree of channel correlation is introduced by the local scattering spatial correlation model. Note that the traditional ZF detection is able to achieve the ML detection performance if the channel matrix is orthogonal. Nevertheless, PSD still achieves near-ML performance by improving $K$.

In Fig. 11, the comparison regarding the average numbers of visited nodes number $|S|$ obtained by PSD for $16 \times 16$ uncoded MIMO systems using 64-QAM is given. Note that the $16 \times 16$ uncoded MIMO detection corresponds to CVP with dimension $n = 32$. As a comparison, the number of visited nodes of traditional sampling decoding schemes is $|S|_{\text{IMHK}} = |S|_{\text{randomized}} = |S|_{\text{Klein}} = nK$. On the contrary, as shown in Theorem 4, the number of visited nodes of the proposed PSD algorithm is upper bounded by $|S| < nK$. Specifically, with the increment of $K$, $|S|$ improves gradually as more qualified candidate vectors are obtained by searching threshold and candidate protection. Clearly, even with the optimized $\sigma$ by relaxation, $|S|$ is always much smaller than $nK$ (i.e., $|S| < nK = |S|_{\text{IMHK, Randomized, Klein}}$), which enables a more efficient decoding than these traditional sampling decoding schemes. Meanwhile, in Fig. 12, the collected candidate vectors $|L|$ obtained by the proposed PSD for $16 \times 16$ uncoded MIMO systems using 64-QAM is given as well. Specifically, with the increment of $K$, $|L|$ improves gradually as more qualified candidate vectors are obtained by searching threshold and candidate protection. Clearly, even with the optimized $\sigma$ by relaxation, both $|S|$ and $|L|$ are always much smaller than the $nK$ and $K$ respectively.

Fig. 13 shows the complexity comparison (in flops) of the proposed PSD algorithm with other decoding schemes in different system dimensions, where the flops evaluation scenario that we use comes from [63]. Clearly, in the uncoded MIMO system with 64-QAM, PSD needs a much lower number of flops than other decoding schemes under the same size $K$. This benefit comes from the adaptation of the tree-structure searching with limited number of visited nodes, which reduces the computation in sampling procedures by removing all repetitions and unnecessary calculations. Specifically, the flops cost of PSD with $K = 50$ is less than that of randomized sampling decoding with $K = 15$. More importantly, with the increase of $K$, the decoding performance improves gradually but the complexity increment is mild. This is different from the traditional sphere decoding schemes like IRA and PTPSD as their complexities grow rapidly with the increase of the system dimension. Consequently, better BER performance and a lower complexity requirement make PSD very promising for MIMO detection.

Following the same scenario in Fig. 13, as a complement to illustrate the computational cost, Fig. 14 is given to show the complexity comparison in average elapsed running times. In particular, the uncoded MIMO system takes 64-QAM at SNR per bit = 17dB, and the simulation is conducted by MATLAB R2019a on a single computer, with an Intel Core i7 processor at 2.7GHz, a RAM of 8GB and Windows 10 Enterprise Service Pack operating system. As can be seen clearly, the average elapsed running time of SIC-LLL decoding scheme increases slightly with the increase of the system dimension. On the contrary, the optimal ML decoding from

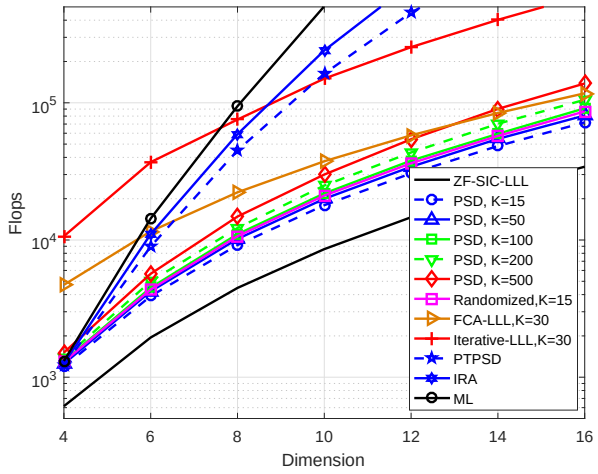


Fig. 13. Complexity comparison in flops for the uncoded MIMO system using 64-QAM at SNR per bit = 17dB.

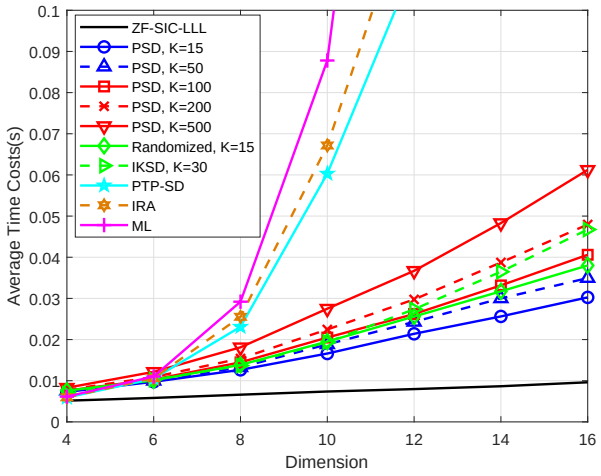


Fig. 14. Complexity comparison in average time cost for the uncoded MIMO system using 64-QAM at SNR per bit = 17dB.

[9] takes an exponentially increasing average elapsed running time. Although a part of the complexity of ML decoding can be reduced by IRA and PTPSD, the complexities of these traditional sphere decoding schemes increase rapidly, making them unaffordable especially in high dimensional systems. As expected, under the same $K$, PSD has a lower average elapsed running time than randomized sampling decoding and IKSD algorithm, making it suitable to implement in MIMO systems.

## VI. Conclusions

In this paper, to overcome the randomness in sampling decoding, a probabilistic searching decoding (PSD) algorithm is proposed based on the lattice Gaussian distribution, which achieves a better decoding performance and less complexity cost in MIMO systems. Based on the standard deviation introduced by model probabilistic twin, the search space of decoding is significantly reduced while a searching threshold is designed to facilitate efficient decoding. Meanwhile, by fully taking advantages of the extra degree of freedom, the explicit decoding trade-off between performance and complexity of PSD is also derived. Moreover, the mechanism of candidate protection is proposed, which generalizes PSD to provide a flexible performance between suboptimal and ML decoding. Finally, further optimization and complexity reduction methods are also given for the proposed PSD algorithm.

## References


[1] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[2] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Globecom*, Taipei, China, Nov. 2002, pp. 424–428.

[3] M. Taherzadeh, A. Mobasher, and A. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. Inform. Theory*, vol. 53, pp. 4801–4805, Dec. 2007.

[4] J. Jalden and P. Elia, "DMT optimality of LR-aided linear decoders for a general class of channels, lattice designs, and system models," *IEEE Trans. Inform. Theory*, vol. 56, no. 10, pp. 4765–4780, Oct. 2010.

[5] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2795–2808, Jun. 2011.

[6] R. Kannan, "Improved algorithms for integer programming and related lattice problems," in *Proc ACM Symp. Theory of Computing*, Boston, Apr. 1983, pp. 193–206.

[7] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[8] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, pp. 2806–2818, Aug. 2005.

[9] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389–2401, Oct. 2003.

[10] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2299 – 2311, Nov. 2004.

[11] J. Jalden, L. Barbero, B. Ottersten, and J. Thompson, "The error probability of the fixed-complexity sphere decoder," *IEEE Trans. Signal Process.*, vol. 57, pp. 2711–2720, Jul. 2009.

[12] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-best MIMO signal detector design and vlsi implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 328–337, March 2007.

[13] L. Luzzi, D. Stehlé, and C. Ling, "Decoding by embedding: correct decoding radius and DMT optimality," *IEEE Trans. Inform. Theory*, vol. 59, no. 5, pp. 2960–2973, 2013.

[14] N. T. Nguyen, K. Lee, and H. Dai, "QR-decomposition-aided Tabu search detection for large MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4857–4870, 2019.

[15] A. K. Sah and A. K. Chaturvedi, "Sequential and global likelihood ascent search-based detection in large MIMO systems," *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 713–725, 2018.

[16] M. K. Izadinasab and O. Damen, "Bridging the gap between MMSE-DFE and optimal detection of MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 220–231, 2020.

[17] G. He, X. Zhang, and Z. Liang, "Algorithm and architecture of an efficient MIMO detector with cross-level parallel tree-search," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 467–479, 2020.

[18] H.-Y. Lu, L.-P. Chang, and H.-S. Hung, "Partial tree search assisted symbol detection for massive MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 319–13 327, 2020.

[19] B. Hassibi, M. Hansen, A. Dimakis, H. Alshamary, and W. Xu, "Optimized Markov Chain Monte Carlo for signal detection in MIMO systems: An analysis of the stationary distribution and mixing time," *IEEE Transactions on Signal Processing*, vol. 62, no. 17, pp. 4436–4450, Sep. 2014.

[20] T. Datta, N. Kumar, A. Chockalingam, and B. Rajan, "A novel Monte Carlo sampling based receiver for large-scale uplink multiuser MIMO systems," *IEEE Transactions on Vehicular Technology,*, vol. 62, no. 7, pp. 3019–3038, Sep. 2013.

[21] P. Aggarwal and X. Wang, "Multilevel sequential Monte Carlo algorithms for MIMO demodulation," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 750–758, Feb. 2007.

[22] J. Choi, "An MCMC-MIMO detector as a stochastic linear system solver using successive overrelexation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1445–1455, Feb. 2016.

[23] P. Klein, "Finding the closest lattice vector when it is unusually close," in *ACM-SIAM Symp. Discr. Algorithms*, 2000, pp. 937–941.

[24] S. Liu, C. Ling, and D. Stehlé, "Decoding by sampling: A randomized lattice algorithm for bounded distance decoding," *IEEE Trans. Inform. Theory*, vol. 57, pp. 5933–5945, Sep. 2011.

[25] Z. Wang, S. Liu, and C. Ling, "Decoding by sampling - Part II: Derandomization and soft-output decoding," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4630–4639, Nov. 2013.

[26] R. Chen, J. Liu, and X. Wang, "Convergence analysis and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. on Signal Process.*, vol. 50, no. 2, pp. 255–270, 2002.

[27] Z. Wang, C. Ling, and G. Hanrot, "Markov chain Monte Carlo algorithms for lattice Gaussian sampling," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, USA, Jun. 2014, pp. 1489–1493.

[28] H. Zhu, B. Farhang-Boroujeny, and R.-R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods," *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 669–672, 2005.

[29] Z. Wang and C. Ling, "On the geometric ergodicity of Metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 738–751, Feb. 2018.

[30] ——, "Lattice Gaussian sampling by Markov chain Monte Carlo: Bounded distance decoding and trapdoor sampling," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3630–3645, June 2019.

[31] Z. Wang, L. Liu, and C. Ling, "Sliced lattice Gaussian sampling: Convergence improvement and decoding optimization," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2599–2612, 2021.

[32] Y. Xia and D. P. Mandic, "Augmented performance bounds on strictly linear and widely linear estimators with complex data," *IEEE Trans. on Signal Process.*, vol. 66, no. 2, pp. 507–514, Jan 2018.

[33] L. Dai, X. Gao, X. Su, S. Han, C. I, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss-Seidel method for uplink multiuser large-scale MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4839–4845, Oct. 2015.

[34] A. Lu, X. Gao, Y. R. Zheng, and C. Xiao, "Low complexity polynomial expansion detector with deterministic equivalents of the moments of channel Gram matrix for massive MIMO uplink," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 586–600, Feb. 2016.

[35] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, "Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 902–915, Oct. 2014.

[36] F. Jin, Q. Liu, H. Liu, and P. Wu, "A low complexity signal detection scheme based on improved Newton iteration for massive MIMO systems," *IEEE Communications Letters*, vol. 23, no. 4, pp. 748–751, 2019.

[37] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," in *Proc. Ann. Symp. Found. Computer Science*, Rome, Italy, Oct. 2004, pp. 372–381.

[38] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehlé, "Semantically secure lattice codes for the Gaussian wiretap channel," *IEEE Trans. Inform. Theory*, vol. 60, no. 10, pp. 6399–6416, Oct. 2014.

[39] C. Ling and J.-C. Belfiore, "Achieiving the AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inform. Theory*, vol. 60, no. 10, pp. 5918–5929, Oct. 2014.

[40] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Ann. ACM Symp. Theory of Comput.*, Victoria, Canada, 2008, pp. 197–206.

[41] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Time*, American Mathematical Society, 2008.

[42] N. Stephens-Davidowitz, "Discrete Gaussian sampling reduces to CVP and SVP," submitted for publication. [Online]. Available: http://arxiv.org/abs/1506.07490.

[43] Z. Wang, Y. Huang, and S. Lyu, "Lattice-reduction-aided Gibbs algorithm for lattice Gaussian sampling: Convergence enhancement and decoding optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4342–4356, 2019.

[44] Z. Wang, "Markov chain Monte Carlo methods for lattice Gaussian sampling: Convergence analysis and enhancement," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 6711–6724, 2019.

[45] Z. Wang, S. Lyu, Y. Xia, and Q. Wu, "Expectation propagation-based sampling decoding: Enhancement and optimization," *IEEE Transactions on Signal Processing*, vol. 69, pp. 195–209, 2021.

[46] D. Aharonov and O. Regev, "Lattice problems in NP ∩ coNP." *J. ACM*, vol. 52, no. 5, pp. 749–765, 2005.

[47] J. H. Conway and N. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1998.

[48] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Math. Annalen*, vol. 261, pp. 515–534, 1982.

[49] C. Ling, W. H. Mow, and N. Howgrave-Graham, "Reduced and Fixed-Complexity variants of the LLL algorithm for communications," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 1040–1050, 2013.

[50] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2701–2710, 2009.

[51] K. Nikitopoulos, G. Georgis, C. Jayawardena, D. Chatzipanagiotis, and R. Tafazolli, "Massively parallel tree search for high-dimensional sphere decoders," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2309–2325, 2019.

[52] C. Jayawardena and K. Nikitopoulos, "G-multisphere: Generalizing massively parallel detection for non-orthogonal signal transmissions," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1227–1239, 2020.

[53] C. Husmann, R. Tafazolli, and K. Nikitopoulos, "Antipodal detection and decoding for large multi-user MIMO with reduced base-station antennas," in *2018 IEEE Globecom Workshops (GC Workshops)*, 2018, pp. 1–6.

[54] K. N. C. Husmann, G. Georgis and K. Jamieson, "Flexcore: Massively parallel and flexible processing for large MIMO access points," in *USENIX NSDI*, 2017, pp. 197–211.

[55] K. Nikitopoulos, "Massively parallel, nonlinear processing for 6G: Potential gains and further research challenges," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 81–87, 2022.

[56] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2661–2675, 2007.

[57] B. Shim and I. Kang, "Sphere decoding with a probabilistic tree pruning," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4867–4878, 2008.

[58] S. Han, T. Cui, and C. Tellambura, "Improved K-best sphere detection for uncoded and coded MIMO systems," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 472–475, 2012.

[59] W. Zhang and X. Ma, "Low-complexity soft-output decoding with lattice-reduction-aided detectors," *IEEE Trans. Commun.*, vol. 58, no. 9, pp. 2621–2629, Sep. 2010.

[60] T. Shimokawa and T. Fujino, "Iterative lattice reduction aided MMSE list detection in MIMO system," in *Proc. IEEE International Conference on Advanced Technologies for Communications*, Oct. 2008, pp. 50–54.

[61] B. Costa, A. Mussi, and T. Abrao, "MIMO detectors under correlated channels," *Semina: Cincias Exatas e Tecnolgicas*, vol. 37, p. 3, 03 2016.

[62] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017. [Online]. Available: http://dx.doi.org/10.1561/2000000093

[63] H. Qian, "Counting the floating point operations (FLOPS)," *MATLAB Central File Exchange*, no. 50608, June, 2015.


**Zheng Wang** (Member, IEEE) received the B.S. degree in electronic and information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, and the M.S. degree in communications from University of Manchester, Manchester, U.K., in 2010. He received the Ph.D degree in communication engineering from Imperial College London, UK, in 2015.

Since 2021, he has been an Associate Professor in the School of Information and Engineering, Southeast University, Nanjing, China. From 2015 to 2016 he served as a Research Associate at Imperial College London, UK. From 2016 to 2017 he was an senior engineer with Radio Access Network R&D division, Huawei Technologies Co.. From 2017 to 2020 he was an Associate Professor at the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His current research interests include massive MIMO systems, machine learning and data analytics over wireless networks, and lattice theory for wireless communications.

**Cong Ling** received the bachelors and masters degrees from the Nanjing Institute of Communications Engineering, China, in 1995 and 1997 respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2005. He is currently a Reader (equivalent to Professor/Associate Professor) with the Electrical and Electronic Engineering Department, Imperial College London. He is also a member of the Academic Centre of Excellence in Cyber Security Research, Imperial College, and an affiliated member of the Institute of Security Science and Technology, Imperial College. Before joining Imperial College, he had been on the faculties of the Nanjing Institute of Communications Engineering and Kings College. He visited The Hong Kong University of Science and Technology as a Hong Kong Telecom Institute of Information Technology (HKTIIT) fellow in 2009.

He has been an Associate Editor (in multiterminal communications and lattice coding) of IEEE Transactions on Communications, and an Associate Editor of IEEE Transactions on Vehicular Technology and on the program committees of several international conferences, including IEEE Information Theory Workshop, Globecom, and ICC.

**Shi Jin** (Senior Member, IEEE) received the B.S. degree in communications engineering from the Guilin University of Electronic Technology, Guilin, China, in 1996, the M.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, and the Ph.D. degree in information and communications engineering from Southeast University, Nanjing, in 2007. From June 2007 to October 2009, he was a Research Fellow with University College London, Adastral Park Research Campus, London, U.K. He is currently with the Faculty of the National Mobile Communications Research Laboratory, Southeast University. His research interests include space time wireless communications, random matrix theory, and information theory. He and his coauthors have been awarded the 2011 IEEE Communications Society Stephen O. Rice Prize Paper Award in the field of communication theory and the 2010 Young Author Best Paper Award by the IEEE Signal Processing Society. He served as an Associate Editor for the IEEE Transactions on Wireless Communications, IEEE Communications Letters, and IET Communications.

**Yongming Huang** (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, China, in 2007.

Since March 2007, he has been a Faculty with the School of Information Science and Engineering, Southeast University, where he is currently a Full Professor. During 2008-2009, he visited the Signal Processing Lab, Royal Institute of Technology, Stockholm, Sweden. He has authored or coauthored more than 200 peer-reviewed papers, and holds more than 80 invention patents. His research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He was an Associate Editor for the IEEE Transactions on Signal Processing and a Guest Editor of the IEEE Journal Selected Areas in Communications. He is currently an Editor-at-Large of the IEEE Open Journal of the Communications Society and an Associate Editor for the IEEE Wireless Communications Letters.

**Feifei Gao** (M'09-SM'14-F'20) received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China in 2002, the M.Sc. degree from McMaster University, Hamilton, ON, Canada in 2004, and the Ph.D. degree from National University of Singapore, Singapore in 2007. Since 2011, he joined the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. Prof. Gao's research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence assisted communications. He has authored/coauthored more than 150 refereed IEEE journal papers and more than 150 IEEE conference proceeding papers that are cited more than 12400 times in Google Scholar. Prof. Gao has served as an Editor of IEEE Transactions on Wireless Communications, IEEE Journal of Selected Topics in Signal Processing (Lead Guest Editor), IEEE Transactions on Cognitive Communications and Networking, IEEE Signal Processing Letters (Senior Editor), IEEE Communications Letters (Senior Editor), IEEE Wireless Communications Letters, and China Communications. He has also served as the symposium co-chair for 2019 IEEE Conference on Communications (ICC), 2018 IEEE Vehicular Technology Conference Spring (VTC), 2015 IEEE Conference on Communications (ICC), 2014 IEEE Global Communications Conference (GLOBECOM), 2014 IEEE Vehicular Technology Conference Fall (VTC), as well as Technical Committee Members for more than 50 IEEE conferences.