

Effects of the LLL Reduction on the Success Probability of the Babai Point and on the Complexity of Sphere Decoding

Xiao-Wen Chang, Jinming Wen, and Xiaohu Xie

Abstract—A common method to estimate an unknown integer parameter vector in a linear model is to solve an integer least squares (ILS) problem. A typical approach to solving an ILS problem is sphere decoding. To make a sphere decoder faster, the well-known LLL reduction is often used as preprocessing. The Babai point produced by the Babai nearest plane algorithm is a suboptimal solution of the ILS problem. First, we prove that the success probability of the Babai point as a lower bound on the success probability of the ILS estimator is sharper than the lower bound given by Hassibi and Boyd [1]. Then, we show rigorously that applying the LLL reduction algorithm will increase the success probability of the Babai point and give some theoretical and numerical test results. We give examples to show that unlike LLL's column permutation strategy, two often used column permutation strategies SQRD and V-BLAST may decrease the success probability of the Babai point. Finally, we show rigorously that applying the LLL reduction algorithm will also reduce the computational complexity of sphere decoders, which is measured approximately by the number of nodes in the search tree in the literature.

Index Terms—Babai point, complexity, integer least squares (ILS) problem, LLL reduction, sphere decoding, success probability.

I. INTRODUCTION

CONSIDER the following linear model:

$$\mathbf{y} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{v} \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^m$ is an observation vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a deterministic model matrix with full column rank, $\hat{\mathbf{x}} \in \mathbb{Z}^n$ is an unknown integer parameter vector, and $\mathbf{v} \in \mathbb{R}^m$ is a noise vector following the Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ with σ being known. A common method to estimate $\hat{\mathbf{x}}$ in (1) is to solve the following integer least squares (ILS) problem:

$$\min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad (2)$$

Manuscript received April 07, 2012; revised November 02, 2012; accepted March 14, 2013. Date of publication June 20, 2013; date of current version July 10, 2013.

X.-W. Chang and X. Xie are with the School of Computer Science, McGill University, Montreal, QC H3A 2A7, Canada (e-mail: chang@cs.mcgill.ca; xiaohu.xie@mail.mcgill.ca).

J. Wen is with the Department of Mathematics and Statistics, McGill University, Montreal, QC H3A 0B9, Canada (e-mail: jinming.wen@mail.mcgill.ca).

Communicated by R. F. H. Fischer, Associate Editor for Communications.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2013.2253596

whose solution \mathbf{x}^{ILS} is the maximum-likelihood estimator of $\hat{\mathbf{x}}$. The ILS problem is also referred to as the closest point problem in the literature as it is equivalent to finding a point in the lattice $\{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\}$ which is closest to \mathbf{y} .

A typical approach to solving (2) is the discrete search approach, referred to as sphere decoding in communications, such as the Schnorr–Euchner algorithm [2] or its variants, see, e.g., [3] and [4]. To make the search faster, a lattice reduction is performed to transform the given problem to an equivalent problem. A widely used reduction is the LLL reduction proposed by Lenstra *et al.* in [5].

It has been shown that the ILS problem is NP-hard [6], [7]. Solving (2) may become time-prohibitive when \mathbf{A} is ill conditioned, the noise is large, or the dimension of the problem is large [8]. So for some applications, an approximate solution, which can be produced quickly, is computed instead. One often used approximate solution is the Babai point, produced by Babai's nearest plane algorithm [9]. This approximate solution is also the first integer point found by the Schnorr–Euchner algorithm. In communications, a method for finding this approximate solution is referred to as a successive interference cancellation decoder.

In order to verify whether an estimator is good enough for a practical use, one needs to find the probability of the estimator being equal to the true integer parameter vector, which is referred to as success probability [1]. The probability of wrong estimation is referred to as error probability, see, e.g., [10]. If the Babai point is used as an estimator of the integer parameter vector $\hat{\mathbf{x}}$ in (1), certainly it is important to find its success probability, which can easily be computed. Even if one intends to compute the ILS estimator, it is still important to find the success probability of the Babai point. It is very difficult to compute the success probability of the ILS estimator, so lower and upper bounds have been considered to approximate it, see, e.g., [1] and [11]. In [12], it was shown that the success probability of the ILS estimator is the largest among all “admissible” estimators, including the Babai point, which is referred to as a bootstrapping estimator in [12]. The success probability of the Babai point is often used as an approximation to the success probability of the ILS estimator. In general, the higher the success probability of the Babai point, the lower the complexity of finding the ILS estimator by the discrete search approach. In practice, if the success probability of the Babai point is high, say close to 1, then one does not need to spend extra computational time to find the ILS estimator.

Numerical experiments have shown that after the LLL reduction, the success probability of the Babai point increases [13]. But whether the LLL reduction can always improve the success probability of the Babai point is still unknown. In this paper, we will prove that the success probability of the Babai point will become higher after the LLL reduction algorithm is used. It is well known that the LLL reduction can make sphere decoders faster. But to our knowledge there is still no rigorous justification. We will show that the LLL reduction can always decrease the computational complexity of sphere decoders, an approximation to the number of nodes in the search tree given in the literature.

The rest of this paper is organized as follows. In Section II, we introduce the LLL reduction to reduce the ILS problem (2). In Section III, we introduce the Babai point and a formula to compute the success probability of the Babai point, and we show that the success probability of the Babai point is a sharper lower bound on the success probability of ILS estimator compared with the lower bound given in [1]. In Section IV, we rigorously prove that the LLL reduction algorithm improves the success probability of the Babai point. In Section V, we rigorously show that the LLL reduction algorithm reduces the computational complexity of sphere decoders. Finally, we summarize this paper in Section VI.

In this paper, \mathbf{e}_k denotes the k th column of the identity matrix \mathbf{I} . For $\mathbf{x} \in \mathbb{R}^n$, we use $\lfloor \mathbf{x} \rfloor$ to denote its nearest integer vector, i.e., each entry of \mathbf{x} is rounded to its nearest integer (if there is a tie, the one with smaller magnitude is chosen). For a vector \mathbf{x} , $\mathbf{x}_{i:j}$ denotes the subvector of \mathbf{x} formed by entries $i, i+1, \dots, j$. For a matrix \mathbf{A} , $\mathbf{A}_{i:j, i:j}$ denotes the submatrix of \mathbf{A} formed by rows and columns $i, i+1, \dots, j$. The success probabilities of the Babai point and the ILS estimator are denoted by P_B and P_{ILS} , respectively.

II. LLL REDUCTION AND TRANSFORMATION OF THE ILS PROBLEM

Assume that \mathbf{A} in the linear model (1) has the QR factorization

$$\mathbf{A} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where $[\mathbf{Q}_1, \mathbf{Q}_2] \in \mathbb{R}^{m \times m}$ is orthonormal and $\mathbf{R} \in \mathbb{R}^{n \times n}$ is upper triangular. Without loss of generality, we assume the diagonal entries of \mathbf{R} are positive throughout the paper. Define $\tilde{\mathbf{y}} = \mathbf{Q}_1^T \mathbf{y}$. From (1), we have $\tilde{\mathbf{y}} = \mathbf{R}\hat{\mathbf{x}} + \mathbf{Q}_1^T \mathbf{v}$. Because $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, it follows that $\tilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{R}\hat{\mathbf{x}}, \sigma^2 \mathbf{I})$.

With the QR factorization of \mathbf{A} , the ILS problem (2) can be transformed to

$$\min_{\mathbf{x} \in \mathbb{Z}^n} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2^2. \quad (3)$$

One can then apply a sphere decoder such as the Schnorr–Euchner search algorithm [2] to find the solution of (3).

The efficiency of the search process depends on \mathbf{R} . For efficiency, one typically uses the LLL reduction instead of the QR

factorization. After the QR factorization of \mathbf{A} , the LLL reduction [5] reduces the matrix \mathbf{R} in (3) to $\bar{\mathbf{R}}$

$$\bar{\mathbf{Q}}^T \mathbf{R} \mathbf{Z} = \bar{\mathbf{R}} \quad (4)$$

where $\bar{\mathbf{Q}} \in \mathbb{R}^{n \times n}$ is orthonormal, $\mathbf{Z} \in \mathbb{Z}^{n \times n}$ is a unimodular matrix (i.e., $\det(\mathbf{Z}) = \pm 1$), and $\bar{\mathbf{R}} \in \mathbb{R}^{n \times n}$ is upper triangular with positive diagonal entries and satisfies the following conditions:

$$|\bar{r}_{ik}| \leq \frac{1}{2} \bar{r}_{ii}, \quad i = 1, 2, \dots, k-1 \quad (5)$$

$$\delta \bar{r}_{k-1, k-1}^2 \leq \bar{r}_{k-1, k}^2 + \bar{r}_{k, k}^2, \quad k = 2, 3, \dots, n \quad (6)$$

where δ is a constant satisfying $1/4 < \delta \leq 1$. The matrix \mathbf{R} is said to be δ -LLL reduced or simply LLL reduced. Equations (5) and (6) are referred to as the size-reduced condition and the Lovász condition, respectively.

The original LLL algorithm given in [5] can be described in the matrix language. Two types of basic unimodular matrices are implicitly used to update \mathbf{R} so that it satisfies the two conditions. One is the integer Gauss transformations (IGT) matrices and the other is permutation matrices, see below.

To meet the first condition in (5), we can apply an IGT, which has the following form:

$$\mathbf{Z}_{ik} = \mathbf{I} - \zeta \mathbf{e}_i \mathbf{e}_k^T.$$

Applying \mathbf{Z}_{ik} ($i < k$) to \mathbf{R} from the right gives

$$\bar{\mathbf{R}} = \mathbf{R} \mathbf{Z}_{ik} = \mathbf{R} - \zeta \mathbf{R} \mathbf{e}_i \mathbf{e}_k^T.$$

Thus, $\bar{\mathbf{R}}$ is the same as \mathbf{R} , except that $\bar{r}_{jk} = r_{jk} - \zeta r_{ji}$ for $j = 1, \dots, i$. By setting $\zeta = \lfloor r_{ik}/r_{ii} \rfloor$, we ensure $|\bar{r}_{ik}| \leq \bar{r}_{ii}/2$.

To meet the second condition in (6), permutations are needed in the reduction process. Suppose that $\delta r_{k-1, k-1}^2 > r_{k-1, k}^2 + r_{k, k}^2$ for some k . Then, we interchange columns $k-1$ and k of \mathbf{R} . After the permutation, the upper triangular structure of \mathbf{R} is no longer maintained. But we can bring \mathbf{R} back to an upper triangular matrix by using the Gram–Schmidt orthogonalization technique (see [5]) or by a Givens rotation

$$\bar{\mathbf{R}} = \mathbf{G}_{k-1, k}^T \mathbf{R} \mathbf{P}_{k-1, k} \quad (7)$$

where $\mathbf{G}_{k-1, k}$ is an orthonormal matrix and $\mathbf{P}_{k-1, k}$ is a permutation matrix, and

$$\begin{aligned} \bar{r}_{k-1, k-1}^2 &= r_{k-1, k}^2 + r_{k, k}^2 \\ \bar{r}_{k-1, k}^2 + \bar{r}_{k, k}^2 &= r_{k-1, k-1}^2. \end{aligned} \quad (8)$$

Note that the aforementioned operation guarantees $\delta \bar{r}_{k-1, k-1}^2 < \bar{r}_{k-1, k}^2 + \bar{r}_{k, k}^2$ since $\delta \leq 1$. The LLL reduction algorithm is described in Algorithm 1, where the final reduced upper triangular matrix is still denoted by $\bar{\mathbf{R}}$.

Algorithm 1 LLL Reduction

- 1: compute the QR factorization: $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$;
- 2: set $\mathbf{Z} = \mathbf{I}_n, k = 2$;
- 3: **while** $k \leq n$ **do**
- 4: apply IGT $\mathbf{Z}_{k-1,k}$ to reduce $r_{k-1,k}$: $\mathbf{R} = \mathbf{R}\mathbf{Z}_{k-1,k}$;
- 5: update \mathbf{Z} : $\mathbf{Z} = \mathbf{Z}\mathbf{Z}_{k-1,k}$;
- 6: **if** $\delta r_{k-1,k-1}^2 > r_{k-1,k}^2 + r_{kk}^2$ **then**
- 7: permute and triangularize \mathbf{R} : $\mathbf{R} = \mathbf{G}_{k-1,k}^T \mathbf{R}\mathbf{P}_{k-1,k}$;
- 8: update \mathbf{Z} : $\mathbf{Z} = \mathbf{Z}\mathbf{P}_{k-1,k}$;
- 9: $k = k - 1$, when $k > 2$;
- 10: **else**
- 11: **for** $i = k - 2, \dots, 1$ **do**
- 12: apply IGT \mathbf{Z}_{ik} to reduce r_{ik} : $\mathbf{R} = \mathbf{R}\mathbf{Z}_{ik}$;
- 13: update \mathbf{Z} : $\mathbf{Z} = \mathbf{Z}\mathbf{Z}_{i,k}$;
- 14: **end for**
- 15: $k = k + 1$;
- 16: **end if**
- 17: **end while**

After the LLL reduction (4), the ILS problem (3) is then transformed to

$$\min_{\mathbf{z} \in \mathbb{Z}^n} \|\bar{\mathbf{y}} - \bar{\mathbf{R}}\mathbf{z}\|_2^2, \quad (9)$$

where $\bar{\mathbf{y}} = \bar{\mathbf{Q}}^T \tilde{\mathbf{y}}$ and $\mathbf{z} = \mathbf{Z}^{-1}\mathbf{x}$.

The LLL reduction is a powerful preprocessing tool that allows us to reduce the complexity of search process for finding the ILS solution, see, e.g., [1] and [3].

III. SUCCESS PROBABILITY OF THE BABAI POINT AND A LOWER BOUND

The Babai (integer) point $\mathbf{x}^B \in \mathbb{Z}^n$ found by the Babai nearest plane algorithm [9] is defined as follows:

$$\begin{aligned} c_n &= \tilde{y}_n / r_{nn}, \quad x_n^B = \lfloor c_n \rfloor, \\ c_i &= (\tilde{y}_i - \sum_{j=i+1}^n r_{ij} x_j^B) / r_{ii}, \quad x_i^B = \lfloor c_i \rfloor \end{aligned} \quad (10)$$

for $i = n - 1, \dots, 1$. Note that the entries of \mathbf{x}^B are determined from the last to the first. The Babai point \mathbf{x}^B is actually the first integer point found by the Schnorr–Euchner search algorithm [2] for solving (3).

In the following, we give a formula for the success probability of the Babai point. The formula is equivalent to the one given by Teunissen in [14], which considers a variant form of the ILS

problem (2). But our proof is easier to follow than that given in [14].

Theorem 1: Suppose $\tilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{R}\hat{\mathbf{x}}, \sigma^2\mathbf{I})$ in the ILS problem (3). Let P_B denote the success probability of the Babai point \mathbf{x}^B given in (10), i.e., $P_B = \Pr(\mathbf{x}^B = \hat{\mathbf{x}})$. Then

$$P_B = \prod_{i=1}^n \phi(r_{ii}), \quad \phi(\zeta) = \sqrt{\frac{2}{\pi}} \int_0^{\zeta/(2\sigma)} \exp(-\frac{1}{2}t^2) dt. \quad (11)$$

Proof: By the chain rule of conditional probabilities

$$\begin{aligned} P_B &= \Pr(\mathbf{x}^B = \hat{\mathbf{x}}) = \Pr\left(\bigcap_{i=1}^n (x_i^B = \hat{x}_i)\right) = \Pr(x_n^B = \hat{x}_n) \\ &\quad \times \prod_{i=1}^{n-1} \Pr(x_i^B = \hat{x}_i | x_{i+1}^B = \hat{x}_{i+1}, \dots, x_n^B = \hat{x}_n). \end{aligned} \quad (12)$$

Since $\tilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{R}\hat{\mathbf{x}}, \sigma^2\mathbf{I})$, we have

$$\begin{aligned} \tilde{y}_n &\sim \mathcal{N}(r_{nn}\hat{x}_n, \sigma^2) \\ \tilde{y}_i &\sim \mathcal{N}(r_{ii}\hat{x}_i + \sum_{j=i+1}^n r_{ij}\hat{x}_j, \sigma^2), \quad i = n - 1, \dots, 1. \end{aligned}$$

Thus, from (10), we have

$$c_n \sim \mathcal{N}(\hat{x}_n, \sigma^2/r_{nn}^2)$$

and if $x_{i+1}^B = \hat{x}_{i+1}, \dots, x_n^B = \hat{x}_n$

$$c_i \sim \mathcal{N}(\hat{x}_i, \sigma^2/r_{ii}^2).$$

Then, it follows that

$$\begin{aligned} \Pr(x_n^B = \hat{x}_n) &= \Pr(|c_n - \hat{x}_n| \leq 1/2) \\ &= \frac{1}{\sqrt{2\pi} \frac{\sigma}{r_{nn}}} \int_{-0.5}^{0.5} \exp(-\frac{t^2}{2(\frac{\sigma}{r_{nn}})^2}) dt \\ &= \frac{2}{\sqrt{2\pi}} \int_0^{r_{nn}/(2\sigma)} \exp(-\frac{1}{2}t^2) dt = \phi(r_{nn}). \end{aligned}$$

Similarly, we can obtain

$$\Pr(x_i^B = \hat{x}_i | x_{i+1}^B = \hat{x}_{i+1}, \dots, x_n^B = \hat{x}_n) = \phi(r_{ii}).$$

Then, from (12), we can conclude that (11) holds. \square

Since P_B in (11) depends on \mathbf{R} , sometimes we also write P_B as $P_B(\mathbf{R})$.

The success probability P_{ILS} of the ILS estimator depends on its Voronoi cell [1] and it is difficult to compute it because the shape of Voronoi cell is complicated. In [1], a lower bound $F(d_{\min}^2/(4\sigma^2), n)$ is proposed to approximate it, where d_{\min} is the length of the shortest lattice vector, i.e., $d_{\min} = \min_{\mathbf{0} \neq \mathbf{x} \in \mathbb{Z}^n} \|\mathbf{R}\mathbf{x}\|_2$, and F is the cumulative distribution function of chi-square distribution. However, no polynomial-time algorithm has been found to compute d_{\min} . To overcome this problem, Hassibi and Boyd [1] proposed a more practical lower bound $F(r_{\min}^2/(4\sigma^2), n)$, where $r_{\min} \equiv \min_i r_{ii}$. Note that P_B is also a lower bound on P_{ILS}

(see [12]). The following result shows that P_B is sharper than $F(r_{\min}^2/(4\sigma^2), n)$.

Theorem 2: $F\left(\frac{r_{\min}^2}{4\sigma^2}, n\right) \leq P_B$.

Proof: Let $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. Thus, u_1, u_2, \dots, u_n are i.i.d. and $\sum_{i=1}^n u_i^2$ follows the chi-squared distribution with degree n . Let events $E = \{\sum_{i=1}^n u_i^2 \leq r_{\min}^2/(4\sigma^2)\}$ and $E_i = \{u_i^2 \leq r_{ii}^2/(4\sigma^2)\}$ for $i = 1, 2, \dots, n$. Since $r_{\min} \leq r_{ii}$, $E \subseteq \bigcap_{i=1}^n E_i$. Thus

$$\begin{aligned} F\left(\frac{r_{\min}^2}{4\sigma^2}, n\right) &= \Pr(E) \leq \Pr\left(\bigcap_{i=1}^n E_i\right) = \prod_{i=1}^n \Pr(E_i) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \int_{-r_{ii}/(2\sigma)}^{r_{ii}/(2\sigma)} \exp\left(-\frac{1}{2}t^2\right) dt \\ &= \prod_{i=1}^n \phi(r_{ii}) = P_B. \end{aligned}$$

□

In the following, we give an example to show that $F(r_{\min}^2/(4\sigma^2), n)$ can be much smaller than P_B .

Example 1: Let $\mathbf{R} = \begin{bmatrix} 0.001 & 0 \\ 0 & 10 \end{bmatrix}$ and $\sigma = 0.5$. By simple calculations, we obtain $F(r_{\min}^2/(4\sigma^2), n)/P_B = 1/1596$. Although this is a contrived example, where the signal-to-noise ratio is small, it shows that P_B can be much sharper than $F(r_{\min}^2/(4\sigma^2), n)$ as a lower bound on P_{ILS} .

IV. ENHANCEMENT OF P_B BY THE LLL REDUCTION

In this section, we rigorously prove that column permutations and size reductions in the LLL reduction process given in Algorithm 1 enhance (not strictly) the success probability P_B of the Babai point. We give simulations to show that unlike LLL's column permutation strategy, two often used column permutation strategies SQRD [15] and V-BLAST [16] may decrease the success probability of the Babai point. We will also discuss how the parameter δ affects the enhancement and give some upper bounds on P_B after the LLL reduction.

A. Effects of the LLL Reduction on P_B

Suppose that we have the QRZ factorization (4), where $\bar{\mathbf{Q}}$ is orthonormal, \mathbf{Z} is unimodular, and $\bar{\mathbf{R}}$ is upper triangular with positive diagonal entries (we do not assume that $\bar{\mathbf{R}}$ is LLL reduced unless we state otherwise). Then, with $\bar{\mathbf{y}} = \bar{\mathbf{Q}}^T \tilde{\mathbf{y}}$ and $\mathbf{z} = \mathbf{Z}^{-1} \mathbf{x}$, the ILS problem (3) can be transformed to (9). For (9), we can also define its corresponding Babai point \mathbf{z}^B . This Babai point can be used as an estimator of $\hat{\mathbf{z}} \equiv \mathbf{Z}^{-1} \hat{\mathbf{x}}$, or equivalently, $\mathbf{Z}\mathbf{z}^B$ can be used as an estimator of $\hat{\mathbf{x}}$. In (3), $\tilde{\mathbf{y}} \sim \mathcal{N}(\bar{\mathbf{R}}\hat{\mathbf{x}}, \sigma^2 \mathbf{I})$. It is easy to verify that in (9), $\bar{\mathbf{y}} \sim \mathcal{N}(\bar{\mathbf{R}}\hat{\mathbf{z}}, \sigma^2 \mathbf{I})$. In the following, we look at how the success probability of the Babai point changes after some specific transformation is used to $\bar{\mathbf{R}}$.

The following result shows that if the Lovász condition (6) is not satisfied, after a column permutation and triangularization, the success probability of the Babai point increases.

Lemma 1: Suppose that $\delta r_{k-1, k-1}^2 > r_{k-1, k}^2 + r_{kk}^2$ for some k for the $\bar{\mathbf{R}}$ matrix in the ILS problem (3). After the permutation of columns $k-1$ and k and triangularization, $\bar{\mathbf{R}}$ becomes $\bar{\bar{\mathbf{R}}}$, i.e., $\bar{\bar{\mathbf{R}}} = \mathbf{G}_{k-1, k}^T \bar{\mathbf{R}} \mathbf{P}_{k-1, k}$ (see (7)). With $\bar{\bar{\mathbf{y}}} = \mathbf{G}_{k-1, k}^T \bar{\mathbf{y}}$ and $\mathbf{z} =$

$\mathbf{P}_{k-1, k}^{-1} \mathbf{x}$, (3) can be transformed to (9). Denote $\hat{\mathbf{z}} \equiv \mathbf{P}_{k-1, k}^{-1} \hat{\mathbf{x}}$. Then, the Babai point \mathbf{z}^B has a success probability greater than or equal to the Babai point \mathbf{x}^B , i.e.,

$$\Pr(\mathbf{x}^B = \hat{\mathbf{x}}) \leq \Pr(\mathbf{z}^B = \hat{\mathbf{z}}) \quad (13)$$

where the equality holds if and only if $r_{k-1, k} = 0$.

Proof: By Theorem 1, what we need to show is the following inequality:

$$\prod_{i=1}^n \phi(r_{ii}) \leq \prod_{i=1}^n \phi(\bar{r}_{ii}). \quad (14)$$

Since $\bar{r}_{ii} = r_{ii}$ for $i \neq k-1, k$, we only need to show

$$\phi(r_{k-1, k-1}) \phi(r_{kk}) \leq \phi(\bar{r}_{k-1, k-1}) \phi(\bar{r}_{kk})$$

which is equivalent to

$$\begin{aligned} &\int_0^{\frac{r_{k-1, k-1}}{2\sigma}} \exp\left(-\frac{1}{2}t^2\right) dt \int_0^{\frac{r_{kk}}{2\sigma}} \exp\left(-\frac{1}{2}t^2\right) dt \\ &\leq \int_0^{\frac{\bar{r}_{k-1, k-1}}{2\sigma}} \exp\left(-\frac{1}{2}t^2\right) dt \int_0^{\frac{\bar{r}_{kk}}{2\sigma}} \exp\left(-\frac{1}{2}t^2\right) dt. \end{aligned} \quad (15)$$

Since $\mathbf{G}_{k-1, k}$ is orthonormal and $\mathbf{P}_{k-1, k}$ is a permutation matrix, the absolute value of the determinant of the submatrix $\bar{\mathbf{R}}_{k-1: k, k-1: k}$ is unchanged, i.e., we have

$$r_{k-1, k-1} r_{kk} = \bar{r}_{k-1, k-1} \bar{r}_{kk}. \quad (16)$$

Let

$$a = \frac{r_{k-1, k-1}}{2\sigma} \frac{r_{kk}}{2\sigma} = \frac{\bar{r}_{k-1, k-1}}{2\sigma} \frac{\bar{r}_{kk}}{2\sigma} \quad (17)$$

$$f(\zeta) = \ln \int_0^\zeta \exp\left(-\frac{1}{2}t^2\right) dt + \ln \int_0^{a/\zeta} \exp\left(-\frac{1}{2}t^2\right) dt. \quad (18)$$

Note that $f(\zeta) = f(a/\zeta) = f(\max\{\zeta, a/\zeta\})$. Then, (15) is equivalent to

$$f\left(\frac{\max\{r_{k-1, k-1}, r_{kk}\}}{2\sigma}\right) \leq f\left(\frac{\max\{\bar{r}_{k-1, k-1}, \bar{r}_{kk}\}}{2\sigma}\right). \quad (19)$$

Obviously, if $r_{k-1, k} = 0$, then the equality in (19) holds since in this case

$$\frac{\max\{r_{k-1, k-1}, r_{kk}\}}{2\sigma} = \frac{\max\{\bar{r}_{k-1, k-1}, \bar{r}_{kk}\}}{2\sigma}.$$

So we only need to show if $r_{k-1, k} \neq 0$, then the strict inequality in (19) holds. In the following, we assume $r_{k-1, k} \neq 0$.

From $\delta r_{k-1, k-1}^2 > r_{k-1, k}^2 + r_{kk}^2$ and (8), we can conclude that

$$r_{kk}, \bar{r}_{k-1, k-1}, \bar{r}_{kk} < r_{k-1, k-1}.$$

Then, with (17), it follows that

$$\begin{aligned} &\frac{\max\{r_{k-1, k-1}, r_{kk}\}}{2\sigma} = \frac{r_{k-1, k-1}}{2\sigma} \\ &> \frac{\max\{\bar{r}_{k-1, k-1}, \bar{r}_{kk}\}}{2\sigma} \geq \sqrt{a}. \end{aligned}$$

Thus, to show the strict inequality in (19) holds, it suffices to show that when $\zeta > \sqrt{a}$, $f(\zeta)$ is a strict monotonically decreasing function or equivalently $f'(\zeta) < 0$.

From (18)

$$f'(\zeta) = \frac{\exp(-\frac{1}{2}\zeta^2)}{\int_0^\zeta \exp(-\frac{1}{2}t^2)dt} - \frac{\frac{a}{\zeta^2} \exp(-\frac{(a/\zeta)^2}{2})}{\int_0^{a/\zeta} \exp(-\frac{1}{2}t^2)dt}$$

$$= \frac{1}{\zeta} \left(g(\zeta) - g\left(\frac{a}{\zeta}\right) \right)$$

where $g(\zeta) = \frac{\zeta \exp(-\frac{1}{2}\zeta^2)}{\int_0^\zeta \exp(-\frac{1}{2}t^2)dt}$. Note that $\zeta > \sqrt{a}, \zeta > a/\zeta$.

Thus, in order to show $f'(\zeta) < 0$ for $\zeta > \sqrt{a}$, we need only to show that $g(\zeta)$ is a strict monotonically decreasing function or equivalently $g'(\zeta) < 0$ when $\zeta > 0$.

Simple calculations give

$$g'(\zeta) = \frac{\exp(-\frac{1}{2}\zeta^2)}{(\int_0^\zeta \exp(-\frac{1}{2}t^2)dt)^2}$$

$$\times \left[(1 - \zeta^2) \int_0^\zeta \exp(-\frac{1}{2}t^2)dt - \zeta \exp(-\frac{1}{2}\zeta^2) \right].$$

If $1 - \zeta^2 \leq 0$ and $\zeta > 0$, then obviously $g'(\zeta) < 0$. If $1 - \zeta^2 > 0$ and $\zeta > 0$, since $\exp(-\frac{1}{2}t^2) \leq 1$

$$(1 - \zeta^2) \int_0^\zeta \exp(-\frac{1}{2}t^2)dt \leq \zeta(1 - \zeta^2) < \zeta \exp(-\frac{1}{2}\zeta^2)$$

where the second inequality can easily be verified. Thus, again $g'(\zeta) < 0$ when $\zeta > 0$, completing the proof. \square

Now we make some remarks. The aforementioned proof shows that $f(\zeta)$ for $\zeta \geq \sqrt{a}$ reaches its maximum when $\zeta = \sqrt{a}$. Thus, if $\bar{r}_{k-1,k-1} = \bar{r}_{kk}$ or equivalently

$$r_{k-1,k}^2 + r_{kk}^2 = r_{k-1,k-1}r_{kk}$$

P_B will increase most. For a more general result, see Lemma 4 and the remark after it.

In Lemma 1, there is no requirement that $r_{k-1,k}$ should be size reduced. The question we would like to ask here is do size reductions in the LLL reduction algorithm affect P_B ? From (11), we observe that P_B only depends on the diagonal entries of \mathbf{R} . Thus, size reductions *alone* will not change P_B . However, if a size reduction can bring changes to the diagonal entries of \mathbf{R} after a permutation, then it will likely affect P_B . Therefore, all the size reductions on the off-diagonal entries above the super-diagonal have no effect on P_B . But the size reductions on the superdiagonal entries may affect P_B . There are a few different situations, which we will discuss below.

Suppose that the Lovász condition (6) holds for a specific k . If (6) does not hold any more after the size reduction on $r_{k-1,k}$, then columns $k-1$ and k of \mathbf{R} are permuted by the LLL reduction algorithm and according to Lemma 1, P_B strictly increases or keeps unchanged if and only if the size reduction makes $r_{k-1,k}$ zero (this occurs if $r_{k-1,k}$ is a multiple of $r_{k-1,k-1}$ before the reduction). If (6) still holds after the size reduction on $r_{k-1,k}$, then this size reduction does not affect P_B .

Suppose that the Lovász condition (6) does not hold for a specific k . Then, by Lemma 1, P_B increases after a permutation and triangularization. If the size reduction on $r_{k-1,k}$ is performed before the permutation, we show in the next lemma that P_B increases further.

Lemma 2: Suppose that in the ILS problem (3) \mathbf{R} satisfies $\delta r_{k-1,k-1}^2 > r_{k-1,k}^2 + r_{kk}^2$ and $|r_{k-1,k}| > r_{k-1,k-1}/2$ for some k . Let $\bar{\mathbf{R}}, \bar{\mathbf{y}}, \mathbf{z}$, and $\hat{\mathbf{z}}$ be defined as in Lemma 1. Suppose a size reduction on $r_{k-1,k}$ is performed first and then after the permutation of columns $k-1$ and k and triangularization, \mathbf{R} becomes $\hat{\mathbf{R}}$, i.e., $\hat{\mathbf{R}} = \hat{\mathbf{G}}_{k-1,k}^T \mathbf{R} \mathbf{Z}_{k-1,k} \mathbf{P}_{k-1,k}$. Let $\hat{\mathbf{y}} = \hat{\mathbf{G}}_{k-1,k}^T \bar{\mathbf{y}}$ and $\mathbf{w} = \mathbf{P}_{k-1,k}^{-1} \mathbf{Z}_{k-1,k}^{-1} \mathbf{x}$; then, (3) is transformed to $\min_{\mathbf{w} \in \mathbb{Z}^n} \|\hat{\mathbf{y}} - \hat{\mathbf{R}}\mathbf{w}\|_2$. Denote $\hat{\mathbf{w}} = \mathbf{P}_{k-1,k}^{-1} \mathbf{Z}_{k-1,k}^{-1} \hat{\mathbf{x}}$. Then, the Babai point \mathbf{w}^B corresponding to the new transformed ILS problem has a success probability greater than or equal to the Babai point \mathbf{z}^B , i.e.,

$$\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) \leq \Pr(\mathbf{w}^B = \hat{\mathbf{w}}) \tag{20}$$

where the equality holds if and only if

$$|r_{k-1,k-1}r_{k-1,k}| = r_{k-1,k}^2 + r_{kk}^2. \tag{21}$$

Proof: Obviously (20) is equivalent to

$$\phi(\bar{r}_{k-1,k-1})\phi(\bar{r}_{kk}) \leq \phi(\hat{r}_{k-1,k-1})\phi(\hat{r}_{kk})$$

which, by the proof of Lemma 1, is also equivalent to

$$f\left(\frac{\max\{\bar{r}_{k-1,k-1}, \bar{r}_{kk}\}}{2\sigma}\right) \leq f\left(\frac{\max\{\hat{r}_{k-1,k-1}, \hat{r}_{kk}\}}{2\sigma}\right)$$

where f is defined in (18). Since $f(\zeta)$ has been showed to be strict monotonically decreasing when $\zeta > \sqrt{a}$, what we need to show is that

$$\max\{\bar{r}_{k-1,k-1}, \bar{r}_{kk}\} \geq \max\{\hat{r}_{k-1,k-1}, \hat{r}_{kk}\} \tag{22}$$

where the equality holds if and only if (21) holds.

Since $|r_{k-1,k}| > r_{k-1,k-1}/2$

$$\bar{r}_{k-1,k-1} = \sqrt{r_{k-1,k}^2 + r_{kk}^2} > \sqrt{r_{k-1,k-1}^2/4 + r_{kk}^2}$$

$$\bar{r}_{kk} = \frac{r_{k-1,k-1}r_{kk}}{\sqrt{r_{k-1,k}^2 + r_{kk}^2}} < \frac{r_{k-1,k-1}r_{kk}}{\sqrt{r_{k-1,k-1}^2/4 + r_{kk}^2}}.$$

But $\sqrt{r_{k-1,k-1}^2/4 + r_{kk}^2} \geq \frac{r_{k-1,k-1}r_{kk}}{\sqrt{r_{k-1,k-1}^2/4 + r_{kk}^2}}$; thus

$$\max\{\bar{r}_{k-1,k-1}, \bar{r}_{kk}\} = \bar{r}_{k-1,k-1}.$$

Suppose that after the size reduction, $r_{k-1,k}$ becomes $\tilde{r}_{k-1,k}$. Note that

$$\hat{r}_{k-1,k-1} = \sqrt{\tilde{r}_{k-1,k}^2 + r_{kk}^2} < \sqrt{r_{k-1,k}^2 + r_{kk}^2} = \bar{r}_{k-1,k-1}.$$

Thus, it follows from (22) what we need to prove is that $\hat{r}_{kk} \leq \bar{r}_{k-1,k-1}$ or equivalently

$$\hat{r}_{kk} \leq \sqrt{r_{k-1,k}^2 + r_{kk}^2} \tag{23}$$

and the equality holds if and only if (21) holds.

By the conditions given in the lemma

$$|r_{k-1,k}| < r_{k-1,k-1} < 2|r_{k-1,k}|.$$

Thus

$$\begin{aligned}\tilde{r}_{k-1,k} &= r_{k-1,k} - \lfloor r_{k-1,k}/r_{k-1,k-1} \rfloor r_{k-1,k-1} \\ &= r_{k-1,k} - \text{sign}(r_{k-1,k})r_{k-1,k-1}.\end{aligned}$$

Now, we consider two cases $r_{k-1,k} > 0$ and $r_{k-1,k} < 0$ separately. If $r_{k-1,k} > 0$, then

$$\begin{aligned}\hat{r}_{kk} &= \frac{r_{k-1,k-1}r_{kk}}{\hat{r}_{k-1,k-1}} = \frac{r_{k-1,k-1}r_{kk}}{\sqrt{\tilde{r}_{k-1,k}^2 + r_{kk}^2}} \\ &= \frac{r_{k-1,k-1}r_{kk}}{\sqrt{(r_{k-1,k} - r_{k-1,k-1})^2 + r_{kk}^2}}.\end{aligned}$$

Thus, to show (23), it suffices to show that

$$\frac{r_{k-1,k-1}r_{kk}}{\sqrt{(r_{k-1,k} - r_{k-1,k-1})^2 + r_{kk}^2}} \leq \sqrt{r_{k-1,k}^2 + r_{kk}^2}.$$

Simple algebraic manipulations shows that the aforementioned inequality is equivalent to

$$(r_{k-1,k-1}r_{k-1,k} - r_{k-1,k}^2 - r_{kk}^2)^2 \geq 0$$

which certainly holds. And obviously, the equality in (23) holds if and only if

$$r_{k-1,k-1}r_{k-1,k} = r_{k-1,k}^2 + r_{kk}^2.$$

If $r_{k-1,k} < 0$, we can similarly prove that (23) holds and the equality holds if and only if

$$-r_{k-1,k-1}r_{k-1,k} = r_{k-1,k}^2 + r_{kk}^2$$

completing the proof. \square

Here, we make a remark about the equality (21). From the proof of Lemma 2, we see that if (21) holds, then the equality in (23) holds, thus $\hat{r}_{kk} = \bar{r}_{k-1,k-1}$. But the absolute value of the determinant of the submatrix $\mathbf{R}_{k-1:k,k-1:k}$ is unchanged by the size reduction; we must have $\hat{r}_{k-1,k-1} = \bar{r}_{kk}$. Thus, if (21) holds, the effect of the size reduction on $r_{k-1,k}$ is to make $\bar{r}_{k-1,k-1}$ and \bar{r}_{kk} permuted; therefore, the success probability P_B is not changed by the size reduction. Here, we give an example.

Example 2: Let $\mathbf{R} = \begin{bmatrix} 5 & 4 \\ 0 & 2 \end{bmatrix}$. Then, it is easy to verify that $\bar{\mathbf{R}} = \begin{bmatrix} 2\sqrt{5} & 2\sqrt{5} \\ 0 & \sqrt{5} \end{bmatrix}$ and $\hat{\mathbf{R}} = \begin{bmatrix} \sqrt{5} & -\sqrt{5} \\ 0 & 2\sqrt{5} \end{bmatrix}$. From the diagonal entries of $\bar{\mathbf{R}}$ and $\hat{\mathbf{R}}$, we can conclude that the success probabilities of the two Babai points corresponding to $\bar{\mathbf{R}}$ and $\hat{\mathbf{R}}$ are equal.

From Lemmas 1 and 2, we immediately obtain the following results.

Theorem 3: Suppose that the ILS problem (3) is transformed to the ILS problem (9), where $\bar{\mathbf{R}}$ is obtained by Algorithm 1. Then

$$\Pr(\mathbf{x}^B = \hat{\mathbf{x}}) \leq \Pr(\mathbf{z}^B = \hat{\mathbf{z}})$$

where the equality holds if and only if no column permutation occurs during the LLL reduction process or whenever two consecutive columns, say $k-1$ and k , are permuted, $r_{k-1,k}$ is a

multiple of $r_{k-1,k-1}$ (before the size reduction on $r_{k-1,k}$ is performed). Any size reductions on the superdiagonal entries of \mathbf{R} that are immediately followed by a column permutation during the LLL reduction process will enhance the success probability of the Babai point. All other size reductions have no effect on the success probability of the Babai point.

Now, we make some remarks. Note that the LLL reduction is not unique. Two different LLL reduction algorithms may produce different \mathbf{R} 's. In Algorithm 1, when the Lovász condition for two consecutive columns is not satisfied; then, a column permutation takes places to ensure the Lovász condition to be satisfied. If an algorithm that computes the LLL reduction does not do permutations as Algorithm 1 does, e.g., the algorithm permutes two columns that are not consecutive or permutes two consecutive columns but the corresponding Lovász condition is not satisfied after the permutation, then we cannot guarantee this specific LLL reduction will increase P_B .

It is interesting to note that [17] showed that all the size reductions on the off-diagonal entries above the superdiagonal of \mathbf{R} have no effect on the residual norm of the Babai point. Here, we see that those size reductions are not useful from another perspective.

If we do not do size reductions in Algorithm 1, the algorithm will do only column permutations. We refer to this column permutation strategy as LLL-permute. The column permutation strategies SQRD [15] and V-BLAST [16] are often used for solving box-constrained ILS problems (see [4] and [18]). In the following, we give simple numerical test results to see how the four methods (SQRD, V-BLAST, LLL-permute with $\delta = 1$ and LLL with $\delta = 1$) affect P_B .

We performed our MATLAB simulations for the following two cases.

- 1) Case 1. $\mathbf{A} = \text{randn}(n, n)$, where $\text{randn}(n, n)$ is a MATLAB built-in function to generate a random $n \times n$ matrix, whose entries follow the normal distribution $\mathcal{N}(0, 1)$.
- 2) Case 2. $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, \mathbf{U}, \mathbf{V} are random orthogonal matrices obtained by the QR factorization of random matrices generated by $\text{randn}(n, n)$ and \mathbf{D} is a $n \times n$ diagonal matrix with $d_{ii} = 10^{3(n/2-i)/(n-1)}$.

In the tests for each case for a fixed n , we gave 200 runs to generate 200 different \mathbf{A} 's. For $n = 20$, Figs. 1 and 2 display the average success probabilities of the Babai points corresponding to various reduction or permutation strategies over 200 runs versus $\sigma = 0.05 : 0.05 : 0.4$, for Cases 1 and 2, respectively. In both figures, "QR" means the QR factorization is used, giving $\Pr(\mathbf{x}^B = \hat{\mathbf{x}})$.

From Figs. 1 and 2, we can see that on average the LLL reduction improves P_B much more significantly than the other three, V-BLAST performs better than LLL-permute and SQRD, and LLL-permute and SQRD have similar performance. We observed the same phenomenon when we changed the dimensions of \mathbf{A} .

Figs. 1 and 2 indicate that on average SQRD and V-BLAST increase P_B . However, unlike LLL-permute, both SQRD and V-BLAST may decrease P_B sometimes. Table I gives the number of runs out of 200 in which SQRD and V-BLAST decrease P_B for various σ and n . From the table, we can see

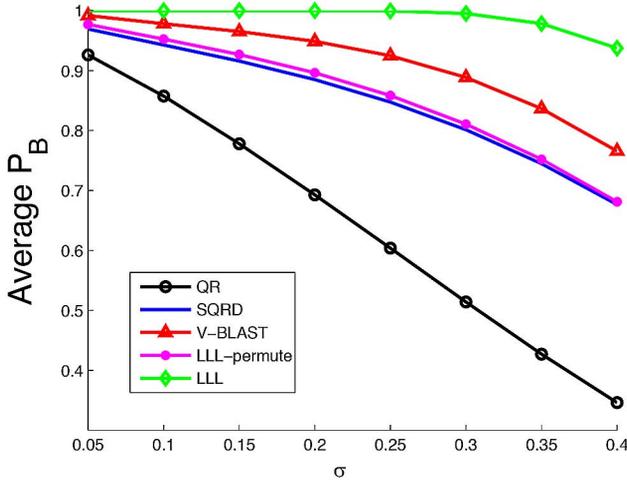
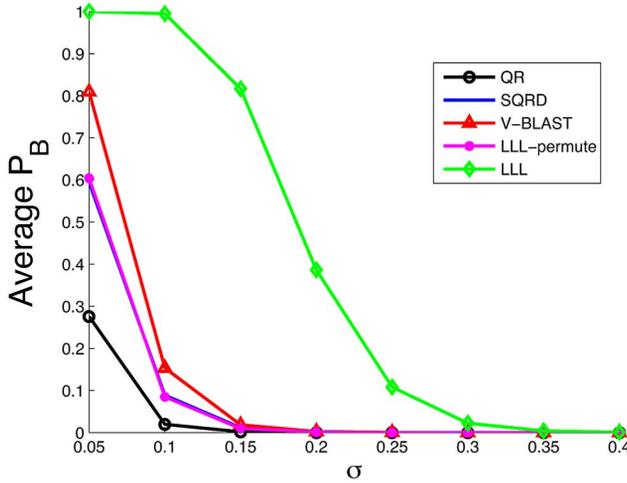

 Fig. 1. Average success probability versus σ for Case 1, $n = 20$.

 Fig. 2. Average success probability versus σ for Case 2, $n = 20$.

 TABLE I
 NUMBER OF RUNS OUT OF 200 IN WHICH P_B DECREASES

Methods	$n \backslash \sigma$	Case 1			Case 2		
		0.1	0.2	0.3	0.1	0.2	0.3
SQRD	10	9	10	6	13	8	5
	20	12	11	7	6	2	1
	30	16	14	11	0	1	1
	40	15	9	5	0	0	0
V-BLAST	10	0	0	0	2	6	7
	20	0	0	0	0	0	0
	30	0	0	0	0	0	0
	40	0	0	0	0	0	0

that for both Cases 1 and 2, the chance that SQRD decreases P_B is much larger than V-BLAST and when σ increases, the chance that SQRD decreases P_B tends to decrease. For Case 2, when n increases, the chance that SQRD decreases P_B tends to decrease, but this phenomenon is not seen for Case 1.

B. Effects of δ on the Enhancement of P_B

Suppose that \mathbf{R}_1 and \mathbf{R}_2 are obtained by applying Algorithm 1 to \mathbf{A} with $\delta = \delta_1$ and $\delta = \delta_2$, respectively and $\delta_1 < \delta_2$.

A natural question is what is the relation between $P_B(\mathbf{R}_1)$ and $P_B(\mathbf{R}_2)$? In the following, we try to address this question. First, we give a result for $n = 2$.

Theorem 4: Suppose that \mathbf{R}_1 and \mathbf{R}_2 are obtained by applying Algorithm 1 to $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $\delta = \delta_1$ and $\delta = \delta_2$, respectively, and $\delta_1 < \delta_2$. If $n = 2$, then

$$P_B(\mathbf{R}_1) \leq P_B(\mathbf{R}_2). \quad (24)$$

Proof: Note that only two columns are involved in the reduction process and the value of δ only determines when the process should terminate. In the reduction process, the upper triangular matrix \mathbf{R} either first becomes δ_1 -LLL reduced and then becomes δ_2 -LLL reduced after some more permutations or becomes δ_1 -LLL reduced and δ_2 -LLL reduced at the same time. Therefore, by Lemma 1, the conclusion holds. \square

However, the inequality (24) in Theorem 4 may not hold when $n \geq 3$. In fact, for any given $n \geq 3$, we can give an example to illustrate this.

Example 3: Let δ_1 and δ_2 satisfy $1/4 < \delta_1 < \delta_2 \leq 1$ and $\delta_2 < \delta_1^2 + 1/4$. Let η and θ satisfy $\delta_1 < \eta < \delta_2$ and $0 < \theta < \frac{1}{2}\sqrt{\delta_1(\eta - \delta_1)}$. Let

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & \sqrt{\eta} & \theta \\ 0 & 0 & \delta_1 \end{bmatrix}. \quad (25)$$

Note that \mathbf{R} is size reduced already.

Suppose that we apply Algorithm 1 with $\delta = \delta_1$ to \mathbf{R} , leading to \mathbf{R}_1 . The first two columns of \mathbf{R} do not permute as the Lovász condition holds. However, the Lovász condition does not hold for the last two columns and a permutation is needed. Then by Lemma 1, we must have $P_B(\mathbf{R}_1) > P_B(\mathbf{R})$.

Applying Algorithm 1 with $\delta = \delta_2$ to \mathbf{R} , we obtain

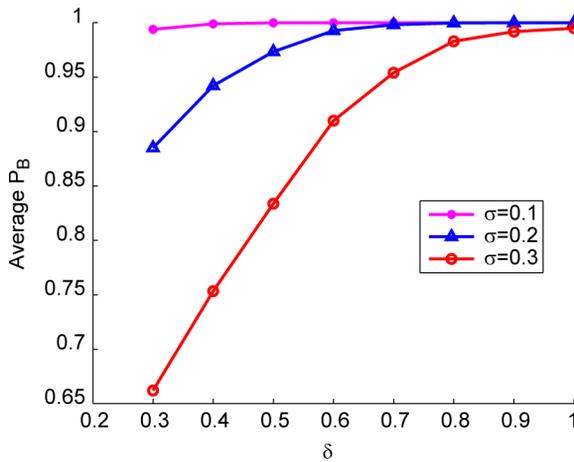
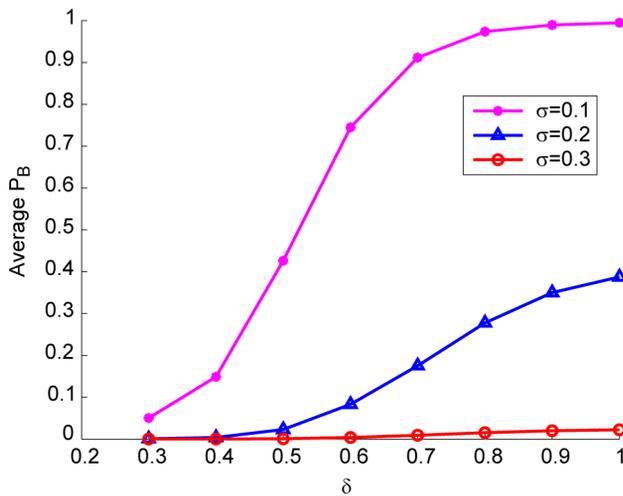
$$\mathbf{R}_2 = \begin{bmatrix} \sqrt{\eta} & 0 & \theta \\ 0 & 1 & 1/2 \\ 0 & 0 & \delta_1 \end{bmatrix}$$

whose diagonal entries are the same as those of \mathbf{R} with a different order. Then, we have $P_B(\mathbf{R}_2) = P_B(\mathbf{R})$. Therefore, $P_B(\mathbf{R}_1) > P_B(\mathbf{R}_2)$.

With $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ given in (25), we define \mathbf{A} as $\mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-3} \end{bmatrix} \in \mathbb{R}^{n \times n}$, it is easy to show that we still have $P_B(\mathbf{R}_1) > P_B(\mathbf{R}_2)$, where \mathbf{R}_1 and \mathbf{R}_2 were obtained by applying Algorithm 1 to \mathbf{A} with $\delta = \delta_1$ and $\delta = \delta_2$, respectively.

Although the aforementioned example shows that larger δ may not guarantee to produce higher P_B when $n \geq 3$, we can expect that the chance that $P_B(\mathbf{R}_1) \leq P_B(\mathbf{R}_2)$ is much higher than the chance that $P_B(\mathbf{R}_1) > P_B(\mathbf{R}_2)$. Here, we give an explanation. If \mathbf{R}_1 is not δ_2 -LLL reduced, applying Algorithm 1 with $\delta = \delta_2$ to \mathbf{R}_1 produces $\bar{\mathbf{R}}_1$ with $P_B(\bar{\mathbf{R}}_1) \geq P_B(\mathbf{R}_1)$. Although $\bar{\mathbf{R}}_1$ may not be equal to \mathbf{R}_2 , we can expect that the difference between these two δ_2 -LLL reduced matrices is small. Thus it is likely that $P_B(\mathbf{R}_2) \approx P_B(\bar{\mathbf{R}}_1) \geq P_B(\mathbf{R}_1)$.

Here, we give numerical results to show how δ affects P_B (i.e., $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$). We used the matrices defined in Cases 1 and 2 of Section IV-A. As earlier, in the tests for each case, we gave 200 runs to generate 200 different \mathbf{A} 's for a fixed n . For $n = 20$, Figs. 3 and 4 display the average $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ over 200

Fig. 3. Average P_B after the LLL reduction for Case 1, $n = 20$.Fig. 4. Average P_B after the LLL reduction for Case 2, $n = 20$.TABLE II
SUCCESS PROBABILITY $\Pr(\mathbf{x}^B = \hat{\mathbf{x}})$

	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
Case 1	0.839	0.661	0.477
Case 2	1.85×10^{-2}	1.95×10^{-4}	5.56×10^{-6}

runs versus $\delta = 0.3 : 0.1 : 1.0$ for Cases 1 and 2, respectively. The three curves in both figures correspond to $\sigma = 0.1, 0.2, 0.3$. For comparisons, we give the corresponding $\Pr(\mathbf{x}^B = \hat{\mathbf{x}})$ in Table II.

From Table II and Figs. 3 and 4, we can see that the LLL reduction has a significant effect on improving P_B . Figs. 3 and 4 show that as δ increases, on average P_B increases too, in particular for large σ . But we want to point out that we also noticed that sometimes a larger δ resulted in a smaller P_B in the tests. Table III gives the exact number of runs out of those 200 runs in which P_B decreases when δ increases from t to $t + 0.1$ for $t = 0.3 : 0.1 : 0.9$. From Table III, we can see that most of the time P_B does not decrease when δ increases. We would like to point out that in our numerical tests, we tried various dimension size n for the two test cases and observed the same phenomena.

TABLE III
NUMBER OF RUNS IN WHICH P_B DECREASES WHEN δ INCREASES

$\delta \backslash \sigma$	Case 1			Case 2		
	0.1	0.2	0.3	0.1	0.2	0.3
0.3—0.4	8	9	10	9	10	11
0.4—0.5	10	9	8	10	11	11
0.5—0.6	13	14	13	12	11	11
0.6—0.7	19	18	16	17	18	20
0.7—0.8	2	10	12	12	13	14
0.8—0.9	3	11	9	15	18	19
0.9—1.0	1	13	8	16	19	22

C. Some Upper Bounds on P_B After the LLL Reduction

We have shown that the LLL reduction by Algorithm 1 can enhance the success probability of the Babai point. A natural question is how much is the enhancement? If the LLL reduction has been computed by Algorithm 1, then we can easily obtain the ratio $\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) / \Pr(\mathbf{x}^B = \hat{\mathbf{x}})$ by using the formula given in (11). If we only know the R-factor of the QR factorization of \mathbf{A} , usually it is impossible to know the ratio exactly. However, we will derive some bounds on $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$, which involve only the R-factor of the QR factorization of \mathbf{A} . From these bounds, one can immediately obtain bounds on the ratio.

Before giving an upper bound on $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$, we give the following result, see, e.g., [19, Th. 6].

Lemma 3: Let \mathbf{R} be the R-factor of the QR factorization of \mathbf{A} and let $\mathbf{R}^{(p)}$ be the upper triangular matrix after the p th column permutation and triangularization in the LLL reduction process by Algorithm 1; then, for $i = 1, 2, \dots, n$

$$\begin{aligned} & \min\{r_{ii}, r_{i+1, i+1}, \dots, r_{nn}\} \\ & \leq r_{ii}^{(p)} \leq \max\{r_{11}, r_{22}, \dots, r_{ii}\}. \end{aligned} \quad (26)$$

When the LLL reduction process finishes, the diagonal entries of the upper triangular matrix certainly satisfy (26). Then, using the second inequality in (26), we obtain the following result from (11).

Theorem 5: Suppose that the ILS problem (3) is transformed to the ILS problem (9) after the LLL reduction by Algorithm 1. The success probability of the Babai point for the ILS problem (9) satisfies

$$\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) \leq \prod_{i=1}^n \phi(\gamma_i) \quad (27)$$

where $\gamma_i = \max\{r_{11}, r_{22}, \dots, r_{ii}\}$.

In the following, we give another upper bound on the success probability of the Babai point, which is invariant to the unimodular transformation to \mathbf{R} . The result was essentially obtained in [20], but our proof is much simpler.

Lemma 4: Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be an upper triangular matrix with positive diagonal entries; then

$$\prod_{i=1}^n \phi(r_{ii}) \leq \phi^n \left(\left(\prod_{i=1}^n r_{ii} \right)^{1/n} \right) \quad (28)$$

where the equality holds if and only if all the diagonal entries of \mathbf{R} are equal.

Proof: Let $h(\xi) = \ln(\phi(\exp(\xi)))$ and $v_i = \ln r_{ii}$ for $i = 1, \dots, n$. Define $v = \frac{1}{n} \sum_{i=1}^n v_i = \frac{1}{n} \ln(\prod_{i=1}^n r_{ii})$. To prove (28), it suffices to show that

$$\frac{1}{n} \sum_{i=1}^n h(v_i) \leq h(v). \quad (29)$$

It is easy to verify that

$$h''(\xi) = \frac{1}{2\sigma} \exp(\xi) g' \left(\frac{1}{2\sigma} \exp(\xi) \right)$$

where $g(\cdot)$ was defined in the proof of Lemma 1. According to the proof of Lemma 1, $g'(\zeta) < 0$ for $\zeta > 0$. Thus, $h''(\xi) < 0$, i.e., $h(\xi)$ is a strictly concave function. Therefore, (29) must hold and the equality holds if and only if all v_i are equal, or equivalently all r_{ii} are equal. \square

Suppose that the ILS problem (3) is transformed to the ILS problem (9) after the LLL reduction by Algorithm 1. Then, $\det(\bar{\mathbf{R}}) = \det(\mathbf{R}) = \prod_{i=1}^n r_{ii}$. Thus, by Lemma 4, we have

$$\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) = \prod_{i=1}^n \phi(\bar{r}_{ii}) \leq \phi^n \left(\left(\prod_{i=1}^n r_{ii} \right)^{1/n} \right). \quad (30)$$

The upper bound is reachable if and only if all the diagonal entries of $\bar{\mathbf{R}}$ are equal to $\det^{1/n}(\mathbf{R})$. If the gap between the largest diagonal entry and the smallest diagonal entry of $\bar{\mathbf{R}}$ is large, the upper bound in (30) will not be tight. In the following, we give an improved upper bound.

Theorem 6: Under the same assumption as in Theorem 5, if there exist indices i_1, i_2, \dots, i_l such that

$$M_k \leq m_{k+1}, \quad k = 1, \dots, l \quad (31)$$

where

$$M_k = \max\{r_{i_{k-1}+1, i_{k-1}+1}, r_{i_{k-1}+2, i_{k-1}+2}, \dots, r_{i_k, i_k}\}$$

$$m_{k+1} = \min\{r_{i_k+1, i_k+1}, r_{i_k+2, i_k+2}, \dots, r_{i_{k+1}, i_{k+1}}\}$$

with $i_0 = 0$ and $i_{l+1} = n$, then

$$\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) \leq \prod_{k=1}^{l+1} \phi^{i_k - i_{k-1}}(\nu_k) \leq \phi^n(\nu) \quad (32)$$

where

$$\nu_k = \left(\prod_{j=i_{k-1}+1}^{i_k} r_{jj} \right)^{1/(i_k - i_{k-1})}, \quad \nu = \left(\prod_{j=1}^n r_{jj} \right)^{1/n}.$$

Proof: Partition \mathbf{R} as follows:

$$\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{l+1}]$$

where the diagonal entries of \mathbf{R} which are in block $\mathbf{R}_k \in \mathbb{R}^{n \times (i_k - i_{k-1})}$ are $r_{i_{k-1}+1, i_{k-1}+1}, r_{i_{k-1}+2, i_{k-1}+2}, \dots, r_{i_k, i_k}$ for $k = 1, \dots, l + 1$. The condition (31) is to ensure that in the LLL reduction process by Algorithm 1, there are no column permutations between \mathbf{R}_k s. Now, we prove this claim. Suppose that Algorithm 1 has just finished the operations on

\mathbf{R}_2 and is going to work on \mathbf{R}_3 . At this moment, $[\mathbf{R}_1, \mathbf{R}_2]$ is LLL reduced. In the LLL reduction of $[\mathbf{R}_1, \mathbf{R}_2]$, no column permutation between the last column of \mathbf{R}_1 and the first column of \mathbf{R}_2 occurred. In fact, by (26) in Lemma 3 and the inequality $M_1 \leq m_2$ from (31), after a permutation, say the p th permutation, in the LLL reduction of $[\mathbf{R}_1, \mathbf{R}_2]$ by Algorithm 1

$$r_{i_1, i_1}^{(p)} \leq \max\{r_{11}, \dots, r_{i_1, i_1}\}$$

$$\leq \min\{r_{i_1+1, i_1+1}, \dots, r_{i_2, i_2}\} \leq r_{i_1+1, r_1+1}^{(p)}.$$

Thus, for any δ satisfying $1/4 < \delta \leq 1$, the Lovász condition (6) is satisfied for columns i_1 and $i_1 + 1$ and no permutation between these two columns would occur. Now, the algorithm goes to work on the first column of \mathbf{R}_3 . Again, we can similarly show that no column permutation between the last column of \mathbf{R}_2 and the first column of \mathbf{R}_3 will occur, so the algorithm will not go back to \mathbf{R}_2 . The algorithm continues and whenever the current block is LLL reduced, it goes to next block and will not come back to the previous block. Then, by applying the result given in (30) for each block \mathbf{R}_k , we obtain the first inequality in (32). The second inequality in (32) is obtained immediately by applying Lemma 4. \square

If indices i_k for $k = 1, \dots, l$ defined in Theorem 6 do not exist, we assume $l = 0$; then, the first inequality in (32) still holds as its right-hand side is just $\phi^n(\nu)$.

We now show how to find these indices if they exist. It is easy to verify that (31) is equivalent to

$$\max\{M_1, \dots, M_k\} \leq \min\{m_{k+1}, \dots, m_{l+1}\} \quad (33)$$

for $k = 1, \dots, l$. Define two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n-1}$ as follows: $u_1 = r_{11}, u_i = \max\{r_{11}, \dots, r_{ii}\} = \max\{u_{i-1}, r_{ii}\}$ for $i = 2, \dots, n - 1$; $v_{n-1} = r_{nn}, v_i = \min\{r_{i+1, i+1}, \dots, r_{nn}\} = \min\{r_{i+1, i+1}, v_{i+1}\}$. Then, (33) is equivalent to

$$u_{i_k} \leq v_{i_k}, \quad k = 1, \dots, l.$$

Thus, we can compare the entries of \mathbf{u} and \mathbf{v} from the first to the last to obtain all indices i_k . It is easy to observe that the total cost is $O(n)$.

Let β_1, β_2 , and β_3 denote the three upper bounds on $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ given in (27) and (32), respectively, i.e.,

$$\beta_1 = \prod_{i=1}^n \phi(\gamma_i), \quad \beta_2 = \prod_{k=1}^{l+1} \phi^{i_k - i_{k-1}}(\nu_k), \quad \beta_3 = \phi^n(\nu).$$

In the following, we first give some special examples to compare β_1, β_2 , and β_3 .

Example 4: Let $\mathbf{R} = \begin{bmatrix} 1/\eta & \times \\ 0 & \eta^2 \end{bmatrix}$, where $0 < \eta < 1$ and \times is any real number. Then

$$\beta_1 = \phi^2(1/\eta), \quad \beta_2 = \beta_3 = \phi^2(\sqrt{\eta}).$$

By the definition of $\phi(\zeta)$ given in (11), $\phi(1/\eta) \rightarrow 1$ and $\phi(\sqrt{\eta}) \rightarrow 0$ when $\eta \rightarrow 0$. Thus, when η is very small, β_2 and β_3 are much sharper than β_1 .

TABLE IV
AVERAGE P_B AND BOUNDS FOR CASE 1, $n = 20$

σ	QR	LLL	β_1	β_2	β_3
0.05	0.93242	1.00000	1.00000	1.00000	1.00000
0.10	0.84706	1.00000	1.00000	1.00000	1.00000
0.15	0.75362	0.99999	1.00000	1.00000	1.00000
0.20	0.66027	0.99966	1.00000	0.99984	0.99984
0.25	0.56905	0.99815	1.00000	0.99891	0.99891
0.30	0.48130	0.99289	1.00000	0.99645	0.99645
0.35	0.39864	0.97589	0.99999	0.98849	0.98849
0.40	0.32279	0.93432	0.99997	0.96319	0.96319

Example 5: Let

$$\mathbf{R} = \begin{bmatrix} \eta/3 & \times & \times & \times \\ 0 & \eta & \times & \times \\ 0 & 0 & 1/\eta^3 & \times \\ 0 & 0 & 0 & \eta/2 \end{bmatrix}, \quad 0 < \eta < 1$$

where \times is any real number. Then

$$\beta_1 = \phi(\eta/3)\phi(\eta)\phi^2(1/\eta^3),$$

$$\beta_2 = \phi(\eta/3)\phi^3(\sqrt[3]{1/(2\eta)}), \quad \beta_3 = \phi^4(\sqrt[4]{1/6}).$$

From the definition of $\phi(\zeta)$, we see that when $\eta \rightarrow 0$

$$\beta_1 \rightarrow 0, \quad \beta_2 \rightarrow 0, \quad \beta_1/\beta_2 \rightarrow 0, \quad \beta_2/\beta_3 \rightarrow 0.$$

Therefore, when η is very small, β_1 is much sharper than β_2 , which is also much sharper than β_3 .

Now, we use more general examples to compare the three upper bounds and also compare them with $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$. In addition to Cases 1 and 2 given in Section IV-A, we also tested the following case.

Case 3. $\mathbf{A} = \mathbf{QR}$, where \mathbf{Q} is a random orthogonal matrix obtained by the QR factorization of a random matrix generated by $\text{randn}(n, n)$ and \mathbf{R} is an $n \times n$ upper triangular matrix with r_{ii}^2 following the χ^2 distribution with freedom degree i and with r_{ij} ($j > i$) following the normal distribution $\mathcal{N}(0, 1)$.

Case 3 is motivated by Case 1. In Case 1, the entries of the R-factor of the QR factorization of \mathbf{A} have the same distributions as the entries of \mathbf{R} in Case 3, except that the freedom degree for r_{ii}^2 is $n - i + 1$, see [21, p99].

In the numerical experiments, for a given n and for each case, we gave 200 runs to generate 200 different \mathbf{A} 's.

All the six tables given below display the average values of $\Pr(\mathbf{x}^B = \hat{\mathbf{x}})$ (corresponding to QR), $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ (corresponding to LLL with $\delta = 1$), β_1, β_2 , and β_3 . For each case, we give two tables. In the first table, n is fixed and σ varies, and in the second table, n varies and σ is fixed. In Tables V and IX, σ was fixed to be 0.4, while in Table VII, σ was fixed to be 0.1. We used different values of σ for these three tables so that $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ is neither close to 0 nor close to 1; otherwise, the bounds would not be much interesting.

For Case 1, from Tables IV and V, we observe that the upper bounds β_2 and β_3 are sharper than the upper bound β_1 , especially when n is small, and the former are good approximations to $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$.

For Case 2, from Table VI, we observe that the upper bound β_1 is extremely loose when σ is large, and β_2 and β_3 are much

TABLE V
AVERAGE P_B AND BOUNDS FOR CASE 1, $\sigma = 0.4$

n	QR	LLL	β_1	β_2	β_3
5	0.37181	0.52120	0.92083	0.55777	0.56437
10	0.33269	0.73310	0.99634	0.75146	0.75146
15	0.30324	0.87116	0.99967	0.89076	0.89076
20	0.32896	0.94211	0.99999	0.97004	0.97004
25	0.31439	0.95364	1.00000	0.98993	0.98993
30	0.32649	0.96961	1.00000	0.99752	0.99752
35	0.34107	0.97361	1.00000	0.99939	0.99939
40	0.32538	0.97579	1.00000	0.99980	0.99980

TABLE VI
AVERAGE P_B AND BOUNDS FOR CASE 2, $n = 20$

σ	QR	LLL	β_1	β_2	β_3
0.05	0.27379	1.00000	1.00000	1.00000	1.00000
0.10	0.01864	0.99490	1.00000	0.99939	0.99939
0.15	0.00161	0.82023	1.00000	0.89650	0.89650
0.20	0.00019	0.38963	1.00000	0.46930	0.46930
0.25	0.00003	0.10896	1.00000	0.13462	0.13462
0.30	0.00001	0.02248	1.00000	0.02738	0.02738
0.35	0.00000	0.00411	1.00000	0.00489	0.00489
0.40	0.00000	0.00074	1.00000	0.00086	0.00086

TABLE VII
AVERAGE P_B AND BOUNDS FOR CASE 2, $\sigma = 0.1$

n	QR	LLL	β_1	β_2	β_3
5	0.06157	0.75079	0.99984	0.83688	0.83688
10	0.05522	0.98875	1.00000	0.99344	0.99344
15	0.03069	0.99670	1.00000	0.99860	0.99860
20	0.01865	0.99486	1.00000	0.99939	0.99939
25	0.01149	0.97374	1.00000	0.99963	0.99963
30	0.00562	0.88945	1.00000	0.99973	0.99973
35	0.00324	0.76654	1.00000	0.99978	0.99978
40	0.00175	0.68623	1.00000	0.99981	0.99981

TABLE VIII
AVERAGE P_B AND BOUNDS FOR CASE 3, $n = 20$

σ	QR	LLL	β_1	β_2	β_3
0.05	0.91780	0.92401	0.92450	0.92471	1.00000
0.10	0.85132	0.86372	0.87017	0.86856	1.00000
0.15	0.77339	0.79087	0.80902	0.79945	1.00000
0.20	0.68615	0.70836	0.74366	0.72379	1.00000
0.25	0.59499	0.62040	0.67610	0.64530	0.99986
0.30	0.50466	0.53153	0.60831	0.56704	0.99837
0.35	0.41858	0.44528	0.54164	0.49161	0.99038
0.40	0.33919	0.36432	0.47679	0.42031	0.96432

TABLE IX
AVERAGE P_B AND BOUNDS FOR CASE 3, $\sigma = 0.4$

n	QR	LLL	β_1	β_2	β_3
5	0.35057	0.37086	0.47342	0.38878	0.53300
10	0.35801	0.38542	0.49866	0.42252	0.75949
15	0.32379	0.35068	0.47865	0.40583	0.90613
20	0.34612	0.37149	0.49066	0.44551	0.96841
25	0.35252	0.37865	0.48907	0.44248	0.99232
30	0.32538	0.35542	0.46208	0.43224	0.99708
35	0.33183	0.35421	0.46524	0.42288	0.99933
40	0.32196	0.34759	0.45264	0.41220	0.99975

sharper for all those σ . From Table VII, we see that when n becomes larger, the upper bounds β_2 and β_3 become worse, although they are still sharper than β_1 . Tables VI and VII show that β_2 is equal to β_3 . Actually, it is indeed true.

For Case 3, from Tables VIII and IX, we observe that the success probability of the Babai point improves after the LLL reduction, but not as much as Cases 1 and 2. We also observe that β_2 is sharper than β_1 , both are much sharper than β_3 , and β_2 is a reasonable approximation to $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$.

Based on the numerical experiments and Theorem 6, we suggest taking $\min\{\beta_1, \beta_2\}$ as an upper bound on $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ in practice.

Although the upper bound $\min\{\beta_1, \beta_2\}$ is a good approximation to $\Pr(\mathbf{z}^B = \hat{\mathbf{z}})$ in the earlier numerical tests, we want to point out that this upper bound can be very loose. Here is a contrived example: suppose all the off-diagonal entries of \mathbf{R} in Example 5 are zero. Then

$$\Pr(\mathbf{x}^B = \hat{\mathbf{x}}) = \Pr(\mathbf{z}^B = \hat{\mathbf{z}}) = \phi(\eta/3)\phi(\eta)\phi(1/\eta^3)\phi(\eta/2).$$

Thus, when $\eta \rightarrow 0$, $\Pr(\mathbf{z}^B = \hat{\mathbf{z}}) / \min\{\beta_1, \beta_2\} \rightarrow 0$.

V. REDUCTION OF THE SEARCH COMPLEXITY BY THE LLL REDUCTION

In this section, we rigorously show that applying the LLL reduction algorithm given in Algorithm 1 can reduce the computational complexity of sphere decoders, which is measured approximately by the number of nodes in the search tree.

The complexity results of sphere decoders given in the literature are often about the complexity of enumerating all integer points in the search region

$$\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2 \leq \beta \quad (34)$$

where β is a constant called the search radius. A typical measure of the complexity is the number of nodes enumerated by sphere decoders, which we denote by ζ .

For $i = n, n-1, \dots, 1$, define E_i as follows:

$$E_i = |\{\mathbf{x}_{i:n} \in \mathbb{Z}^{n-i+1} : \|\tilde{\mathbf{y}}_{i:n} - \mathbf{R}_{i:n,i:n}\mathbf{x}_{i:n}\|_2 \leq \beta\}| \quad (35)$$

where $|\cdot|$ denotes the number of elements in the set. As given in [22], E_i can be estimated as follows:

$$E_i \approx \frac{V_{n-i+1} \beta^{n-i+1}}{|\det(\mathbf{R}_{i:n,i:n})|} = \frac{V_{n-i+1} \beta^{n-i+1}}{r_{ii}r_{i+1,i+1} \cdots r_{nn}} \quad (36)$$

where V_{n-i+1} denotes the volume of an $(n-i+1)$ -dimensional unit Euclidean ball.

This estimation would become the expected value to E_i if $\tilde{\mathbf{y}}_{i:n}$ is uniformly distributed over a Voroni cell of the lattice generated by $\mathbf{R}_{i:n,i:n}$. Then, we have (see, e.g., [23, Sec. 3.2] and [24])

$$\zeta = \sum_{i=1}^n E_i \approx \hat{\zeta}(\mathbf{R}) \equiv \sum_{i=1}^n \frac{V_{n-i+1} \beta^{n-i+1}}{r_{ii}r_{i+1,i+1} \cdots r_{nn}}. \quad (37)$$

In practice, when a sphere decoder such as the Schnorr–Euchner algorithm is used in the search process, after an integer point is found, β will be updated to shrink the search region. But ζ or $\hat{\zeta}$ here does not take this into account for the sake of simplicity.

The following result shows that if the Lovász condition (6) is not satisfied, after a column permutation and triangularization, the complexity $\hat{\zeta}(\mathbf{R})$ decreases.

Lemma 5: Suppose that $\delta r_{k-1,k-1}^2 > r_{k-1,k}^2 + r_{kk}^2$ for some k for the \mathbf{R} matrix in the ILS problem (3). After the permutation

of columns $k-1$ and k and triangularization, \mathbf{R} becomes $\bar{\mathbf{R}}$, i.e., $\bar{\mathbf{R}} = \mathbf{G}_{k-1,k}^T \mathbf{R} \mathbf{P}_{k-1,k}$ (see (7)). Then, the complexity $\hat{\zeta}(\mathbf{R})$ of the search process decreases after the transformation, i.e.,

$$\hat{\zeta}(\mathbf{R}) > \hat{\zeta}(\bar{\mathbf{R}}). \quad (38)$$

Proof: Since $\bar{r}_{ii} = r_{ii}$ for $i \neq k-1, k$, $\bar{r}_{k-1,k-1}\bar{r}_{kk} = r_{k-1,k-1}r_{kk}$, and $\bar{r}_{kk} > r_{kk}$, we have

$$\begin{aligned} \hat{\zeta}(\mathbf{R}) - \hat{\zeta}(\bar{\mathbf{R}}) &= \sum_{i=1}^n \frac{V_{n-i+1} \beta^{n-i+1}}{r_{ii}r_{i+1,i+1} \cdots r_{nn}} - \sum_{i=1}^n \frac{V_{n-i+1} \beta^{n-i+1}}{\bar{r}_{ii}\bar{r}_{i+1,i+1} \cdots \bar{r}_{nn}} \\ &= \frac{V_{n-k+1} \beta^{n-k+1}}{r_{kk}r_{k+1,k+1} \cdots r_{nn}} - \frac{V_{n-k+1} \beta^{n-k+1}}{\bar{r}_{kk}\bar{r}_{k+1,k+1} \cdots r_{nn}} \\ &= \left(\frac{1}{r_{kk}} - \frac{1}{\bar{r}_{kk}} \right) \frac{V_{n-k+1} \beta^{n-k+1}}{r_{k+1,k+1} \cdots r_{nn}} > 0 \end{aligned}$$

completing the proof. \square

Suppose the Lovász condition (6) does not hold for a specific k and furthermore $|r_{k-1,k}| > r_{k-1,k-1}/2$. The next lemma, which is analogous to Lemma 2, shows that the size reduction on $r_{k-1,k}$ performed before the permutation can decrease the complexity $\hat{\zeta}(\mathbf{R})$ further.

Lemma 6: Suppose that in the ILS problem (3) \mathbf{R} satisfies $\delta r_{k-1,k-1}^2 > r_{k-1,k}^2 + r_{kk}^2$ and $|r_{k-1,k}| > r_{k-1,k-1}/2$ for some k . Let $\hat{\mathbf{R}}$ be defined as in Lemma 5. Suppose a size reduction on $r_{k-1,k}$ is performed first and then after the permutation of columns $k-1$ and k and triangularization, $\hat{\mathbf{R}}$ becomes $\hat{\hat{\mathbf{R}}}$, i.e., $\hat{\hat{\mathbf{R}}} = \hat{\mathbf{G}}_{k-1,k}^T \hat{\mathbf{R}} \mathbf{Z}_{k-1,k} \mathbf{P}_{k-1,k}$. Then

$$\hat{\zeta}(\bar{\mathbf{R}}) > \hat{\zeta}(\hat{\hat{\mathbf{R}}}). \quad (39)$$

Proof: By the same argument given in the proof of Lemma 5, we have

$$\hat{\zeta}(\bar{\mathbf{R}}) - \hat{\zeta}(\hat{\hat{\mathbf{R}}}) = \left(\frac{1}{\bar{r}_{kk}} - \frac{1}{\hat{r}_{kk}} \right) \frac{V_{n-k+1} \beta^{n-k+1}}{r_{k+1,k+1} \cdots r_{nn}}.$$

To show (39), we need only to prove $\bar{r}_{kk} < \hat{r}_{kk}$. Since $\bar{r}_{k-1,k-1}\bar{r}_{kk} = \hat{r}_{k-1,k-1}\hat{r}_{kk}$ and $\hat{r}_{k-1,k-1} < \bar{r}_{k-1,k-1}$ (see the proof of Lemma 2), we have $\bar{r}_{kk} < \hat{r}_{kk}$, completing the proof. \square

From Lemmas 5 and 6, we immediately obtain the following result.

Theorem 7: Suppose that the ILS problem (3) is transformed to the ILS problem (9), where $\bar{\mathbf{R}}$ is obtained by Algorithm 1. Then

$$\hat{\zeta}(\mathbf{R}) \geq \hat{\zeta}(\bar{\mathbf{R}})$$

where the equality holds if and only if no column permutation occurs during the LLL reduction process. Any size reductions on the superdiagonal entries of \mathbf{R} , which is immediately followed by a column permutation during the LLL reduction process, will reduce the complexity $\hat{\zeta}$. All other size reductions have no effect on $\hat{\zeta}$.

The result on the effect of the size reductions is consistent with a result given in [25], which shows that all the size reductions on the off-diagonal entries above the superdiagonal of \mathbf{R} and the size reductions on the superdiagonal entries of \mathbf{R} that

are not followed by column permutations have no effect on the search speed of the Schnorr–Euchner algorithm for finding the ILS solution.

Like Theorem 4 in Section IV-B, we can show that when $n = 2$, larger δ will decrease the complexity $\hat{\zeta}$ more, but when $n \geq 3$, it may not be true, although our simulation results indicated that usually it is true.

In Section IV-C, we gave some upper bounds on the success probability of the Babai point after the LLL reduction. Here, we can use (26) to give a lower bound on the complexity $\hat{\zeta}$ after the LLL reduction. To save space, we will not give any details.

VI. SUMMARY AND FUTURE WORK

We have shown that the success probability P_B of the Babai point will increase and the complexity $\hat{\zeta}$ of sphere decoders will decrease if the LLL reduction algorithm given in Algorithm 1 is applied for lattice reduction. We have also discussed how the parameter δ in the LLL reduction affects P_B and $\hat{\zeta}$. Some upper bounds on P_B after the LLL reduction have been presented. In addition, we have shown that P_B is a better lower bound on the success probability of ILS estimator than the lower bound given in [1].

The implementation of LLL reduction is not unique. The KZ reduction [26] is also an LLL reduction. But the KZ conditions are stronger than the LLL conditions. Whether some implementations of the KZ reduction can always increase P_B and decrease $\hat{\zeta}$ and whether the improvement is more significant compared with the regular LLL reduction algorithm given in Algorithm 1 will be studied in the future.

In this paper, we assumed the model matrix \mathbf{A} is deterministic. If \mathbf{A} is a random matrix following some distribution, what is the formula of P_B ? what is the expected value of the search complexity? and how does the LLL reduction affect them? These questions are for future studies.

ACKNOWLEDGMENT

We are grateful to Robert Fischer and the referees for their valuable and thoughtful suggestions. We would also like to thank Damien Stehlé for helpful discussions and for providing a reference.

REFERENCES

- [1] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Trans. Signal Process.*, vol. 46, no. 11, pp. 2938–2952, Nov. 1998.
- [2] C. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, pp. 181–191, 1994.
- [3] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [4] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
- [5] A. Lenstra, H. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.
- [6] P. van Emde Boas, Another NP-complete partition problem and the complexity of computing short vectors in a lattice University of Amsterdam, Dept. Math., Amsterdam, The Netherlands, 1981, Tech. Rep. 81-04.
- [7] D. Micciancio, "The hardness of the closest vector problem with preprocessing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1212–1215, Mar. 2001.
- [8] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.
- [9] L. Babai, "On Lovasz lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.
- [10] J. Jaldén, L. Barbero, B. Ottersten, and J. Thompson, "The error probability of the fixed-complexity sphere decoder," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2711–2720, Jul. 2009.
- [11] P. Xu, "Voronoi cells, probabilistic bounds, and hypothesis testing in mixed integer linear models," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3122–3138, Jul. 2006.
- [12] P. J. G. Teunissen, "An optimality property of integer least-squares estimator," *J. Geodesy*, vol. 73, no. 11, pp. 587–593, 1999.
- [13] Y. H. Gan and W. H. Mow, "Novel joint sorting and reduction technique for delay-constrained LLL-aided MIMO detection," *IEEE Signal Process. Lett.*, vol. 15, pp. 194–197, 2008.
- [14] P. J. G. Teunissen, "Success probability of integer GPS ambiguity rounding and bootstrapping," *J. Geodesy*, vol. 72, no. 10, pp. 606–612, 1998.
- [15] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electron. Lett.*, vol. 37, no. 22, pp. 1348–1350, Oct. 2001.
- [16] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 11, pp. 1841–1852, Nov. 1999.
- [17] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding," in *Proc. IEEE Int. Symp. Inf. Theory*, 2007, pp. 196–200.
- [18] X.-W. Chang and Q. Han, "Solving box-constrained integer least squares problems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 277–287, Jan. 2008.
- [19] P. Q. Nguyen and D. Stehlé, "An LLL algorithm with quadratic complexity," *SIAM J. Comput.*, vol. 39, no. 3, pp. 874–903, 2009.
- [20] P. J. G. Teunissen, "An invariant upperbound for the GNSS bootstrap ambiguity success-rate," *J. Global Position. Syst.*, vol. 2, no. 1, pp. 13–17, 2003.
- [21] R. I. Muirhead, *Aspects of Multivariate Statistical Theory*. New York, NY, USA: Wiley, 1982.
- [22] J. M. W. P. M. Gruber, *Handbook of Convex Geometry*. Amsterdam, The Netherlands: North-Holland, 1993.
- [23] W. Abediseid, Efficient lattice decoders for the linear Gaussian vector channel: Performance & complexity analysis Ph.D. dissertation, University of Waterloo, Dept. Elect. Comput. Eng., Waterloo, ON, Canada, 2011.
- [24] D. Seethaler, J. Jaldén, C. Studer, and H. Bölcskei, "On the complexity distribution of sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5754–5768, Sep. 2011.
- [25] X. Xie, X.-W. Chang, and M. Al Borno, "Partial LLL reduction," *Proc. IEEE GLOBECOM*, p. 5, 2011.
- [26] A. Korkine and G. Zolotareff, "Sur les formes quadratiques," *Mathematische Annalen*, vol. 6, pp. 366–389, 1873.

Xiao-Wen Chang is an Associate Professor in the School of Computer Science at McGill University. He obtained his B.Sc. and M.Sc. in Computational Mathematics from Nanjing University (1986, 1989) and his Ph.D. in Computer Science from McGill University (1997). His research interests are in the area of scientific computing, with particular emphasis on numerical linear algebra and its applications. Currently he is mainly interested in parameter estimation methods, including integer least squares, and as well as their applications in communications, signal processing and satellite-based positioning and wireless localization. He has published about fifty papers in refereed journals.

Jinming Wen received his Bachelor degree in Information and Computing Science from Jilin Institute of Chemical Technology, Jilin, China, in 2008 and his M.Sc. degree in Pure Mathematics from the Mathematics Institute of Jilin University, Jilin, China, in 2010. He is currently pursuing a Ph.D. in The Department of Mathematics and Statistics, McGill University, Montreal. His research interests are in the area of integer least squares problems and their applications in communications and signal processing.

Xiaohu Xie received his Bachelor degree in Computer Science and Technology from Wuhan University of Technology, Wuhan, China, in 2007 and his M.Sc. degree in Computer Science and Technology from Wuhan University of Technology, Wuhan, China, in 2009. He is currently pursuing a Ph.D. in The School of Computer Science, McGill University, Montreal. Currently his research focuses on the theories and algorithms for integer least squares problems.