

Article

# Improved GNSS Ambiguity Fast Estimation Reduction Algorithm

**Xinzhong Li** <sup>1</sup>, **Yongliang Xiong** <sup>1,\*</sup>, **Weiwei Chen** <sup>2</sup>, **Shaoguang Xu** <sup>1</sup> and **Rui Zhang** <sup>1</sup>

<sup>1</sup> Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China; lixinzhong@my.swjtu.edu.cn (X.L.); shaoguangxu@home.swjtu.edu.cn (S.X.); zhangry@my.swjtu.edu.cn (R.Z.)

<sup>2</sup> Department of Civil Engineering, Yibin Campus, Chengdu Technological University, Yibin 644000, China; cwwei1@cdtu.edu.cn

\* Correspondence: yl-xiong@swjtu.edu.cn

**Abstract:** The fast and accurate solution of integer ambiguity is the key to achieve GNSS high-precision positioning. Based on the lattice theory of high-dimensional ambiguity solving, the reduction time consumption is much larger than the search time consumption, and it is especially important to improve the efficiency of the lattice basis reduction algorithm. The Householder QR decomposition with minimal column pivoting is utilized to pre-sort the basis vectors and reduce the number of basis vector exchanges during the reduction process by partial size reduction and relaxing the basis vector exchange condition to improve the reduction efficiency of the LLL algorithm. The improved algorithm is validated using simulated and measured data, respectively, and the performance advantages and disadvantages of the improved algorithm are evaluated from the perspectives of the extent of reduction basis orthogonality and the quality of reduction basis size reduction. The results show that the improved LLL algorithm can significantly reduce the number of basis vector exchanges and the reduction time consumption. The HSLLL and PSLLL algorithms with the Siegel condition as the basis vector exchange condition have a better reduction effect, but are slightly less stable. The PLLLR algorithm significantly improves the search ambiguity resolution efficiency, which is conducive to the rapid realization of ambiguity resolution.



**Citation:** Li, X.; Xiong, Y.; Chen, W.; Xu, S.; Zhang, R. Improved GNSS Ambiguity Fast Estimation Reduction Algorithm. *Sensors* **2023**, *23*, 8568. <https://doi.org/10.3390/s23208568>

Academic Editors: Kamil Krasuski and Damian Wierzbicki

Received: 14 September 2023

Revised: 11 October 2023

Accepted: 17 October 2023

Published: 18 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The resolution of integer ambiguity has a significant impact on the high-precision navigation and positioning results of the carrier phase. The LAMBDA method is currently recognized as the most theoretically rigorous, most efficiently solved, and most widely used ambiguity solving method [1]. It is based on the integer least squares model and reduces the search space by reducing the correlation between the variance components of the ambiguity in order to improve the search efficiency [2]. In addition, many scholars have carried out a lot of fruitful research on the decorrelation algorithm: Liu et al. proposed an approach to united ambiguity decorrelation from the perspective of LU decomposition [3]; Xu proposed a decorrelation algorithm for inverse integer Cholesky decomposition using a pre-sorting strategy [4,5]; Zhou proposed the (inverse) paired Cholesky integer transformation algorithm using upper and lower triangular Cholesky decomposition [6,7]; and Chang et al. improved the LAMBDA algorithm by using a greedy algorithm and lazy transformation strategy, and proposed the MLAMBDA algorithm [8].

Ambiguity resolution is an integer least-squares problem, which is essentially equivalent to the closest vector problem (CVP) in lattice theory, which is an NP-hard problem [9,10]. In order to obtain the nearest vector, it is usually necessary to reduce the basis vector. That is, the integer Gaussian transform is utilized to reduce the correlation of the basis vector, and the basis vector is sorted according to a certain criterion in order to obtain the shortest

possible reduced basis. Among them, the LLL reduction algorithm is the most popular [11]. Therefore, the related methods of ambiguity resolution can be placed in the framework of lattice theory, thus injecting new vitality into the in-depth study of integer ambiguity resolution. Hassib et al. first introduced the LLL reduction algorithm to GNSS ambiguity resolution [12]. Grafarend introduced the solution principle of the LLL algorithm and carried out a detailed data analysis [13]. L.Z. Lou proposed to improve the original LLL algorithm by using a new judgment criterion in response to the defect of iterative non-convergence in the LLL algorithm [14]. Since the LLL algorithm introduces a large rounding error during the rounding process, Z.P. Liu et al. proposed an improved LLL algorithm based on overall matrix rounding [15]. R.H. Yang et al. improved the LLL algorithm by reordering the Gram–Schmidt orthogonal basis [16]. L. Fan and K. Xie improved the LLL algorithm from the perspective of reducing the length of the reduction basis vector [17,18]. Jazaeri et al. compared and analyzed the performance differences between the LAMBDA algorithm and the LLL algorithm [19]. Ling and Howgrave-Graham pointed out that the core of the LLL algorithm lies in the basis vector exchange by analyzing the characteristics of the size reduction and basis vector exchange in the LLL algorithm, and based on this, they proposed the ELLL (Effective LLL) algorithm with partial size reduction [20]. Xie et al. analyzed the effectiveness of the size reduction in the LLL algorithm and proposed the PLLL (Partial LLL) algorithm that selectively performs the size reduction of a column vector in response to the large truncation error of the ELLL algorithm [21]. L.G. Lu et al. improved the LLL algorithm using the greedy selection of the basis vector and partial column vector reduction to reduce the computational complexity of the LLL algorithm [22]. H. Lv et al. improved the original LLL algorithm by using delayed size reduction and partial size reduction in order to reduce the redundant size reduction during the reduction process [23]. Li et al. improved the LLL algorithm based on the Householder transform by using a symmetric pivoting strategy [24].

At the same time, relevant scholars have also conducted in-depth research on the correlation between the LAMBDA decorrelation algorithm and LLL reduction algorithm in lattice theory, as well as the performance evaluation index of lattice basis reduction assisting ambiguity resolution. J.N. Liu et al. theoretically proved the equivalence between the decorrelation and the lattice basis reduction [9]. Lannes further proved the equivalence of the LAMBDA decorrelation algorithm and the LLL algorithm [25]. Borno et al. pointed out through theoretical analysis that simple integer Gaussian transformations do not affect the efficiency of the search for integer ambiguity [26]. Jazaeri et al. analyzed the relationship between commonly used reduction performance evaluation indexes (condition number and orthogonal defect) and search efficiency, and pointed out that the above indexes could not accurately measure the efficiency of an accelerated ambiguity search for lattice basis reduction methods [27]. L.G. Lu et al. compared and analyzed the performance of the LAMBDA decorrelation algorithm and the LLL reduction algorithm under different decomposition methods. They categorized and generalized the common decorrelation (reduction) evaluation indexes from a geometric perspective. It is further illustrated that different evaluation indexes are not directly related to the search efficiency of ambiguity [28].

In view of this, this paper proposes new improved algorithms on the basis of LLL and PLLL algorithms for the characteristics of ambiguity resolution, and appropriately relaxes the exchange conditions of the basis vector in order to reduce the reduction time consumption and improve the computational efficiency of ambiguity resolution. The effectiveness of the improved algorithms and the reduction performance are verified by simulation and measured data.

## 2. Methods and Improvement Strategies

The following describes the notations to be used in this paper. The set of real and integer formed by n-dimensional vectors are denoted by  $\mathbb{R}^n$  and  $\mathbb{Z}^n$ , respectively. MATLAB notation is used to represent submatrices. Specifically, if  $A = (a_{i,j}) \in \mathbb{R}$ , then  $A(i,:)$  denotes

the  $i$ -th row,  $A(:, j)$  denotes the  $j$ -th column, and  $A(i_1 : i_2, j_1 : j_2)$  the submatrix formed by rows  $i_1$  to  $i_2$  and columns  $j_1$  to  $j_2$ . For the element  $(i, j)$  of  $A$ , it is denoted by  $a_{i,j}$  or  $A(i, j)$ .

### 2.1. Integer Least Squares Model

The GNSS observation equation is [29,30]:

$$\mathbf{y} = \mathbf{A}\mathbf{a} + \mathbf{B}\mathbf{b} + \mathbf{e} \quad (1)$$

where  $\mathbf{a}$  is the carrier phase ambiguity parameter,  $\mathbf{b}$  is the baseline parameter of the component to be estimated,  $\mathbf{e}$  is the observation noise,  $\mathbf{y}$  is the carrier phase and pseudorange observation, and  $\mathbf{A}$  and  $\mathbf{B}$  are the design matrix.

Using the least squares criterion [31,32], it can be shown that

$$\min \|\mathbf{y} - \mathbf{A}\mathbf{a} - \mathbf{B}\mathbf{b}\|_{Q_y}^2, \mathbf{a} \in \mathbb{Z}^m, \mathbf{b} \in \mathbb{R}^n \quad (2)$$

where  $\|\cdot\| = (\cdot)^T Q_y^{-1}(\cdot)$ ,  $Q_y$  is the variance covariance matrix of observation  $\mathbf{y}$ . Considering that the ambiguity parameter is an integer, Equation (2) can be further decomposed as:

$$\|\mathbf{y} - \mathbf{A}\mathbf{a} - \mathbf{B}\mathbf{b}\|_{Q_y}^2 = \|\hat{\mathbf{e}}\|_{Q_y}^2 + \|\hat{\mathbf{a}} - \mathbf{a}\|_{Q_{\hat{a}}}^2 + \|\hat{\mathbf{b}}(\mathbf{a}) - \mathbf{b}\|_{Q_{\hat{b}|\hat{a}}}^2 \quad (3)$$

and

$$\left. \begin{aligned} \hat{\mathbf{e}} &= \mathbf{y} - \mathbf{A}\hat{\mathbf{a}} - \mathbf{B}\hat{\mathbf{b}} \\ \hat{\mathbf{b}}(\mathbf{a}) &= \hat{\mathbf{b}} - Q_{\hat{b}\hat{a}} Q_{\hat{a}}^{-1}(\hat{\mathbf{a}} - \mathbf{a}) \\ Q_{\hat{b}|\hat{a}} &= Q_{\hat{b}} - Q_{\hat{b}\hat{a}} Q_{\hat{a}}^{-1} Q_{\hat{a}\hat{b}} \end{aligned} \right\} \quad (4)$$

where  $\hat{\mathbf{a}}$  is the ambiguity float solution,  $\hat{\mathbf{b}}$  is the baseline component corresponding to the ambiguity float solution, and  $\hat{\mathbf{b}}(\mathbf{a})$  is the baseline component corresponding to the ambiguity fixed solution. Since  $\mathbf{b}$  in Equation (1) is a real vector, the third term to the right of Equation (3) should be zero. So when  $\mathbf{b} = \hat{\mathbf{b}}(\mathbf{a})$  and  $\|\hat{\mathbf{a}} - \mathbf{a}\|_{Q_{\hat{a}}}^2$  takes the minimum value,  $\|\mathbf{y} - \mathbf{A}\mathbf{a} - \mathbf{B}\mathbf{b}\|_{Q_y}^2$  takes the minimum value. Therefore, the minimization problem of Equation (2) is transformed into:

$$\min \|\hat{\mathbf{a}} - \mathbf{a}\|_{Q_{\hat{a}}}^2 = \min (\hat{\mathbf{a}} - \mathbf{a})^T Q_{\hat{a}}^{-1} (\hat{\mathbf{a}} - \mathbf{a}), \mathbf{a} \in \mathbb{Z}^m \quad (5)$$

The Cholesky decomposition of  $Q_{\hat{a}}$ , namely,

$$Q_{\hat{a}} = G^T G \quad (6)$$

where  $G$  is the upper triangular matrix.

Substituting Equation (6) into Equation (5) gives

$$\min \|G^{-T}(\hat{\mathbf{a}} - \mathbf{a})\|^2 = \min \|\mathbf{y} - G^{-T}\mathbf{a}\|^2 \quad (7)$$

where  $\mathbf{y} = G^{-T}\hat{\mathbf{a}}$  is a constant.

Equation (7) is also known as the nearest vector problem in lattice theory [33]. In order to obtain the integer solution of the ambiguity rapidly, the decorrelation process is usually used to reduce the correlation between the variance components in  $Q$ , which improves the efficiency of the search for the ambiguity in Equation (5).

## 2.2. LLL Algorithm Based on QR Decomposition

Let  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n \in \mathbb{R}^n$  be a set of linearly independent basis vectors and the lattice  $L(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n)$  represent the set consisting of all linear combinations of integers of  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n$ , i.e.:

$$L(\mathbf{G}) = \left\{ \sum_{i=1}^n x_i \mathbf{g}_i, x_i \in \mathbb{Z} (1 \leq \forall i \leq n) \right\} \quad (8)$$

where  $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n\}$  is called a set of basis of the lattice,  $L(\mathbf{G})$  is the lattice generated by  $\mathbf{G}$ , and  $x_i$  is the combinatorial coefficient of  $\mathbf{g}_i$ .

The classical LLL algorithm implements the reduction on the basis of Gram–Schmidt Orthogonalization (GSO) [11]. Schmidt Orthogonalization is performed on the basis matrix  $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n]$ :

$$\mathbf{G} = \mathbf{G} * \mathbf{U} = [\mathbf{g}_1^*, \mathbf{g}_2^*, \dots, \mathbf{g}_n^*] \begin{bmatrix} 1 & u_{1,2} & \cdots & u_{1,n} \\ 0 & 1 & \cdots & u_{2,n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

In the formula,  $\mathbf{G}^* = [\mathbf{g}_1^*, \mathbf{g}_2^*, \dots, \mathbf{g}_n^*]$  and  $\mathbf{g}_i^* = \mathbf{g}_i - \sum_{j=1}^{i-1} u_{j,i} \mathbf{g}_j^*$ ,  $\mathbf{U} = [u_{j,i}]$  are the unit upper triangular matrix and satisfy  $u_{j,i} = \langle \mathbf{g}_i, \mathbf{g}_j^* \rangle / \|\mathbf{g}_j^*\|^2$ ,  $1 \leq j < i \leq n$ . Matrix  $\mathbf{G}^*$  and  $\mathbf{U}$  satisfy the following two reduction conditions:

$$\left. \begin{array}{l} |u_{j,i}| \leq \frac{1}{2} \quad 1 \leq j < i \leq n \\ \delta \|\mathbf{g}_{i-1}^*\|^2 \leq \|\mathbf{g}_i^*\|^2 + u_{i,i-1}^2 \|\mathbf{g}_{i-1}^*\|^2 \quad \frac{1}{4} < \delta \leq 1 \end{array} \right\} \quad (10)$$

Call  $\mathbf{G}$  the LLL reduction basis parameterized by  $\delta$ . The first equation is called the size reduction and the second equation is the basis vector exchange.

In fact, in order to improve the float accuracy of the lattice basis reduction, the LLL reduction algorithm based on QR decomposition is usually used [34]. The following decomposition is performed on the basis matrix  $\mathbf{G}$ :

$$\mathbf{G} = \mathbf{Q}\mathbf{R} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n] \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ 0 & r_{2,2} & \cdots & r_{2,n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{n,n} \end{bmatrix} \quad (11)$$

In the equation,  $\mathbf{Q}$  is the orthogonal matrix and  $\mathbf{q}_i = \mathbf{g}_i / \|\mathbf{g}_i^*\|$ ,  $\mathbf{R} = [r_{j,i}]$  is the upper triangular matrix and satisfies  $u_{j,i} = r_{j,i} / r_{j,j}$  and  $\|\mathbf{b}_j^*\| = r_{j,j}$ .

Thus, the reduction condition of Equation (8) can be rewritten as:

$$\left. \begin{array}{l} \left| \frac{r_{j,i}}{r_{j,j}} \right| \leq \frac{1}{2} \quad 1 \leq j < i \leq n \\ \delta r_{i-1,i-1}^2 \leq r_{i,i}^2 + r_{i-1,i}^2 \quad \frac{1}{4} < \delta \leq 1 \end{array} \right\} \quad (12)$$

Equation (12) is the reduction condition of the LLL algorithm based on QR decomposition.

In order to satisfy the above reduction conditions, it is usually necessary to construct a transformation matrix for reduction operation.

1. Size reduction: in order to realize the first condition in Equation (10), construct the unimodular matrix  $\mathbf{Z}_{j,i} = \mathbf{I}_n - [r_{j,i} / r_{j,j}]_{\text{int}} \mathbf{e}_j \mathbf{e}_i^T$  ( $[\bullet]_{\text{int}}$  represents the rounding operator), use its right multiplication by the basis matrix  $\mathbf{G}$  to realize the size reduction of the corresponding element, and, at the same time, update the upper triangular matrix  $\mathbf{R}$ .

2. Basis vector exchange (Lovasz condition): if the second condition in Equation (10) is not satisfied, the exchange matrix  $P_{i-1,i}$  is constructed to swap the order of  $g_{i-1}$  and  $g_i$ , and the matrix  $R$  is updated to re-triangularize it.

### 2.3. Improved LLL Algorithm

#### 2.3.1. Householder QR Decomposition Based on Minimum Column Pivoting

The original LLL algorithm performs a QR decomposition of the basis matrix based on GSO. The Householder QR decomposition has lower computational complexity and better numerical stability compared to GSO [35]. Partial LLL reduction algorithms utilize the Householder QR decomposition with minimum column pivoting instead of the regular Householder QR decomposition. In general, the number of basis vector exchanges is a key factor affecting the time consumption of the whole LLL reduction, and it is possible to reduce the number of basis vector exchanges if matrix  $R$  of QR decomposition can be made closer to the LLL reduction basis. It can be obtained from Equation (12) that

$$\left(\delta - \frac{1}{4}\right)r_{i-1,i-1}^2 \leq r_{j,j}^2 \quad \frac{1}{4} < \delta \leq 1 \quad (13)$$

In order to make it easier for matrix  $R$  to satisfy Equation (13), the minimum column pivoting strategy selects the columns that minimize  $|r_{j,j}|$  to be exchanged. In the  $j$ -th step of the QR decomposition, find column  $i$  of  $G(j : n, j : n)$  which has the shortest length and exchange the  $i$ -th column of  $G$  with the  $j$ -th column. The off-diagonal element  $G(j+1 : n, j)$  is then eliminated by the Householder transformation. By using the minimum column pivoting strategy, the Householder QR expression is:

$$Q^T GP = R \quad (14)$$

where  $P \in \mathbb{Z}^n$  is the exchange matrix and  $Q^T = H_n H_{n-1} \cdots H_1$  is the product of  $n$  Householder transformations.

#### 2.3.2. Partial Size Reduction

It has been theoretically demonstrated in the literature that simple size reduction does not affect the number of candidate points for the ambiguity search, and that basis vector exchange is the real goal of lattice basis reduction to accelerate the ambiguity search. In the LLL algorithm, only the size reduction of the secondary diagonal element is generally required. However, considering the lattice basis reduction efficiency and numerical stability of the algorithm, it is necessary to carry out size reduction for partial non-principal secondary diagonal elements under certain conditions.

$$r_{i-1,i} = r_{i-1,i} - \zeta r_{i-1,i-1} \quad (15)$$

where  $\zeta = [r_{i-1,i}/r_{i-1,i-1}]_{\text{int}}$ . The size reduction is applied to the non-principal secondary diagonal elements when they satisfy  $|(r_{i-1,i}/r_{i-1,i-1})_{\text{int}}| \geq 2$ , viz:

$$r_{k,i} = r_{k,i} - [r_{k,i}/r_{k,k}]_{\text{int}} r_{k,k}, \quad k = i-2, i-3, \dots, 1 \quad (16)$$

It should be noted that matrix element size reduction is an integer transformation process, which not only reduces the size of the element itself, but also updates the rest of the column vector accordingly.

Givens rotation has better numerical stability than GSO. Therefore, Givens rotation is used for triangularization after the exchange of the basis vector of the PLLL reduction algorithm. Suppose we exchange the  $k-1$  and  $k$  columns of  $\mathbf{R}$ , i.e.:

$$RP_{k-1,k} = \begin{bmatrix} R_{1,1} & \bar{R}_{1,2} & R_{1,3} \\ & \bar{R}_{2,2} & R_{2,3} \\ & & R_{3,3} \end{bmatrix} \begin{matrix} k-2 \\ 2 \\ n-k \end{matrix} \quad (17)$$

$$\begin{matrix} k-2 & 2 & n-k \end{matrix}$$

then

$$P_{k-1,k} = \begin{bmatrix} I_{k-2} & & \\ & P & \\ & & I_{n-k} \end{bmatrix}, P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tilde{R}_{2,2} = \begin{bmatrix} r_{k-1,k} & r_{k-1,k-1} \\ r_{k,k} & 0 \end{bmatrix}, \bar{R}_{1,2} = [R_{1:k-2,k-1} \ R_{1:k-2,k}].$$

It can be seen that the block matrix  $\tilde{R}_{2,2}$  is not an upper triangular matrix. Therefore, it is triangularized using Givens rotation. Assuming that the Givens rotation matrix is  $\Gamma$ , we have:

$$\bar{R}_{2,2} := \Gamma \tilde{R}_{2,2} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} r_{k-1,k} & r_{k-1,k-1} \\ r_{k,k} & 0 \end{bmatrix} \quad (18)$$

where

$$c = \frac{r_{k-1,k}}{\sqrt{r_{k-1,k}^2 + r_{k,k}^2}}, s = \frac{r_{k,k}}{\sqrt{r_{k-1,k}^2 + r_{k,k}^2}}.$$

Therefore, it can be concluded that

$$\Gamma_{k-1,k} RP_{k-1,k} = \bar{R} = \begin{bmatrix} R_{1,1} & \bar{R}_{1,2} & R_{1,3} \\ \bar{R}_{2,2} & \bar{R}_{2,3} \\ R_{3,3} \end{bmatrix}, \Gamma_{k-1,k} = \begin{bmatrix} I_{k-2} & & \\ & \Gamma & \\ & & I_{n-k} \end{bmatrix}, \bar{R}_{2,3} = \Gamma R_{2,3}.$$

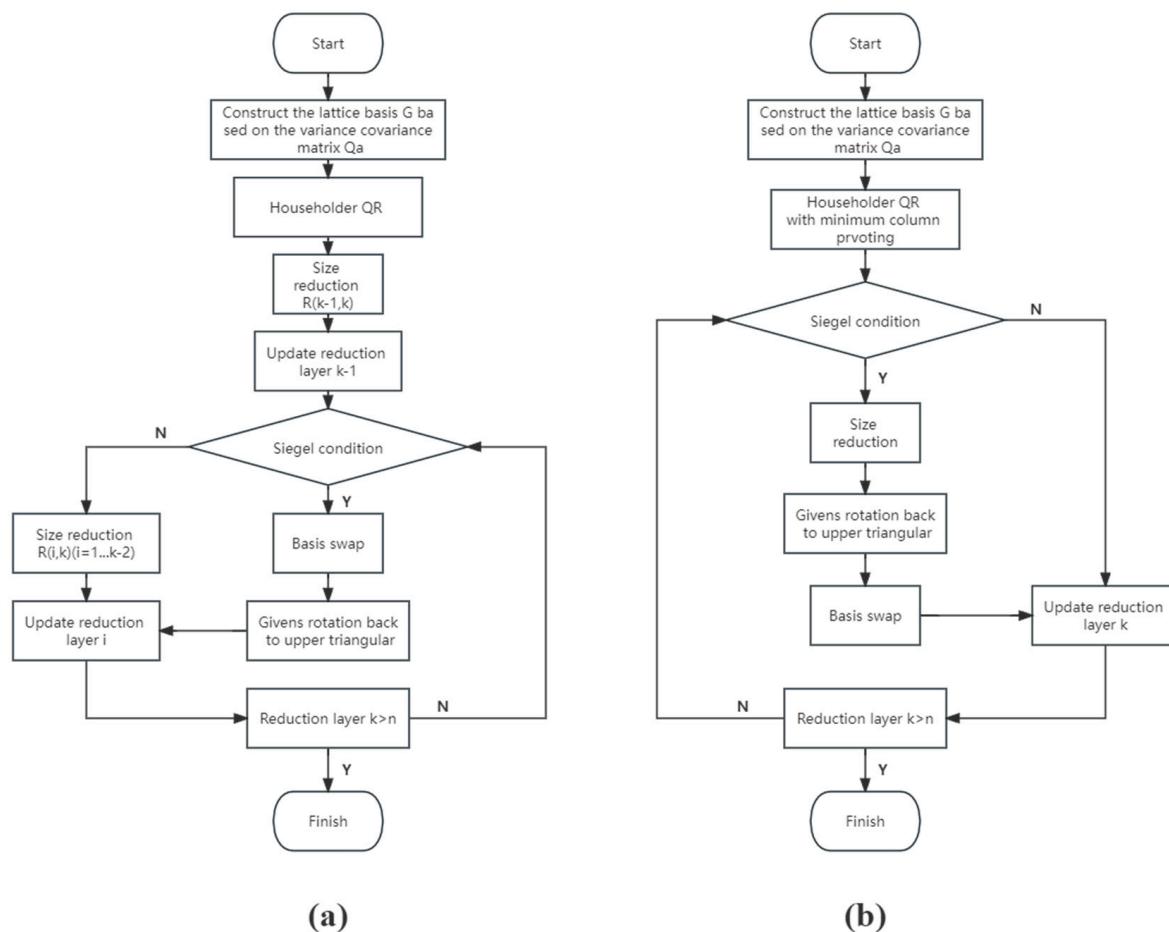
### 2.3.3. Improvement of the LLL Algorithm

From the PLLL reduction algorithm we note that size reduction is only carried out when the basis vector exchange occurs, and the resulting matrix  $\mathbf{R}$  is not fully regulated. Therefore, we add an additional size reduction process at the end of the PLLL reduction algorithm and convert  $\mathbf{R}$  to the LLL reduction matrix. We denote the PLLL algorithm with extra size reduction as PLLLR.

In addition, in the procedure of LLL reduction, it is necessary to detect whether the basis vector satisfies the exchange condition to decide whether it enters the column exchange step or not, and it is obvious that the procedure of LLL reduction can be simplified if the exchange condition of the basis vector is appropriately relaxed in order to reduce the operations, such as the column exchange afterward. Inspired by the literature [36], we replace the Lovasz condition in the LLL reduction with the Siegel condition, and Equation (13) becomes:

$$\left( \delta - \frac{1}{2} \right) r_{i-1,i-1}^2 \leq r_{j,j}^2 \frac{3}{4} \leq \delta \leq 1 \quad (19)$$

We denote the LLL algorithm based on the Householder QR decomposition as HLLL. The HLLL algorithm is where the basis vector exchange condition is replaced by the Siegel condition as HSLLL, and the PLLL algorithm is where the Siegel condition is used as the basis vector exchange condition as PSLLL. The specific flow of the two improved algorithms is shown in Figure 1.



**Figure 1.** (a) Flow chart of HSLLL algorithm; (b) Flow chart of PSLLL algorithm.

### 3. Experiments and Results Analysis

In order to verify the effectiveness of the LLL improvement algorithm proposed in this paper in the application of ambiguity resolution, simulation experiments and measured data are used to compare and analyze HLLL, HSLLL, PLLL, PSLLL, and PLLLR, and to evaluate the performance advantages and disadvantages of each algorithm in terms of the extent of reduction basis orthogonality and the quality of reduction basis size reduction. In ambiguity resolution, the searching process of the ambiguity degree adopts the SE-VB strategy which is widely used at present [10]. The experimental environment is a private PC (Intel Core i7-9700 CPU, 2.80 GHz, 16.0 GB of RAM, 64-bit Windows 10 operating system) and the software is MATLAB R2017 a.

#### 3.1. Indicators for Evaluating the Quality of the Reduced Basis

In measuring the performance of lattice basis reduction, an orthogonal defect (OD) is usually used to reflect the orthogonality of the basis vector, but it has an obvious disadvantage in that only the OD value is obtained, which is not able to intuitively judge the extent of the orthogonality of the reduced basis [37–39]. Therefore, in this paper, the minimum angle  $\theta$  of the reduced basis vector is used instead of the orthogonality defect to measure the extent of the orthogonality of the reduced basis. Its expression is given as:

$$\theta(G) = \min\{\theta_{i,j}, 1 \leq i < j \leq n\} \quad (20)$$

where

$$\theta_{i,j} = \min\{\arccos(\rho_{i,j}), 180^\circ - \arccos(\rho_{i,j})\}$$

$$\rho_{i,j} = \frac{(g_i \cdot g_j)}{\|g_i\| \|g_j\|}$$

By definition, it follows that  $0^\circ \leq \theta(G) \leq 90^\circ$ . If  $\theta(G) = 90^\circ$ , it means that all basis vectors are orthogonal to each other.  $\theta$ , as an alternative indicator of the extent of orthogonality, can be used to roughly determine the orthogonality of the reduced basis intuitively. And the calculation of  $\theta$  and OD is based only on the elements of the variance covariance matrix  $Q_{\hat{a}}$ ; it does not increase the computational complexity.

The purpose of the lattice basis reduction is to make the reduced basis as orthogonal as possible and to make the length of the first basis vector as short as possible after the basis vector exchange. Based on this property, the Hermite factor in lattice theory is introduced as another indicator for evaluating the performance of the reduction [40,41], which is defined as:

$$\kappa = \frac{\|g_1\|}{[\det(Q_{\hat{a}})]^{\frac{1}{2n}}} \quad (21)$$

where  $g_1$  denotes the first basis vector of the lattice basis  $G$ . Obviously,  $\det(Q_{\hat{a}})$  is a fixed value, then the size of the Hermite factor depends on the length of  $g_1$ . The smaller the value of  $\kappa$ , the shorter the length of the first basis vector after the lattice basis reduction, the more adequate the basis vector exchange, and the better the quality of the reduction, and vice versa.

### 3.2. Simulation Experiment

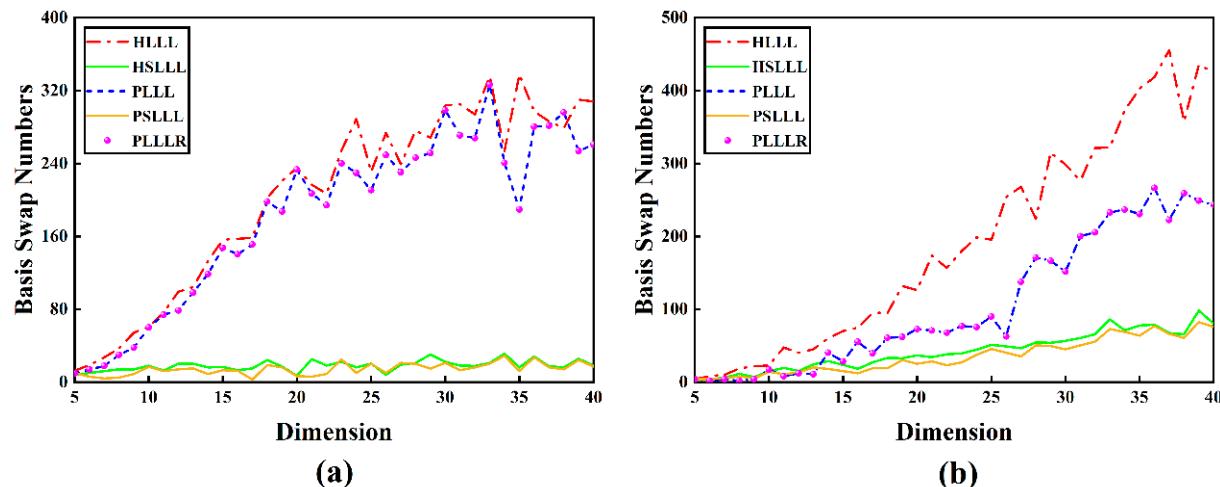
The random simulation method in the literature [8] is used to construct 5–40 dimensional ambiguity float solution  $\hat{a}$  and variance covariance matrix  $Q_{\hat{a}}$ . Each dimension constructs 100 groups of data, which are processed by HLLL, HSLLL, PLLL, PSLLL, and PLRLR algorithms for lattice basis reduction, respectively. These calculate the average number of basis vector swaps, the average reduction time consumption, and the average number of ambiguity candidate points for the 100 groups of data. The specific construction is as follows:

$$\hat{a} = 100 \times \text{randn}(n, 1) \quad Q_{\hat{a}} = LDL^T \quad (22)$$

- Scheme 1:  $L$  is an upper triangular matrix unit and the upper triangular element  $l_{j,i}$  follows the standard normal distribution;  $D = \text{diag}(n^{-1}, (n-1)^{-1}, \dots, 1)$ .
- Scheme 2:  $L$  is a random orthogonal matrix, obtained by the QR decomposition of the random matrix generated by  $\text{randn}(n, n)$ ;  $d_1 = 2^{\frac{n}{4}}, d_n = 2^{-\frac{n}{4}}, d_i \in (d_n, d_1), D = \text{diag}(d_1, \dots, d_i, \dots, d_n)$ .

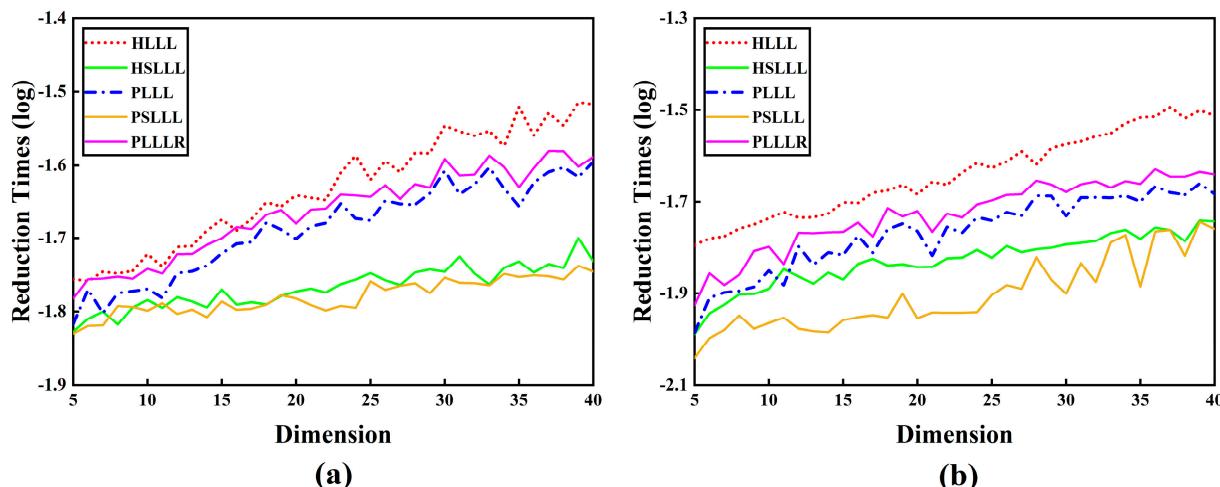
Figure 2 shows the trend of the number of basis vector swaps for the five algorithms in different schemes and dimensions. As seen in Figure 2, the number of basis vector swaps for the five algorithms is positively correlated with the number of dimensions; overall, PSLLL has the fewest number of swaps, and PLLL and PLRLR have the same number of swaps.

By analyzing the results in Figure 2, it can be seen that PLRLR is equivalent to PLLL in terms of the number of basis vector swaps because PLRLR only adds an additional size reduction process, which has no effect on the ordering of the basis vectors, a phenomenon that is in line with the theory. HSLLL and PSLLL simplify the LLL reduction process by relaxing the swap condition of the basis vector, which reduces the number of basis vector swaps.



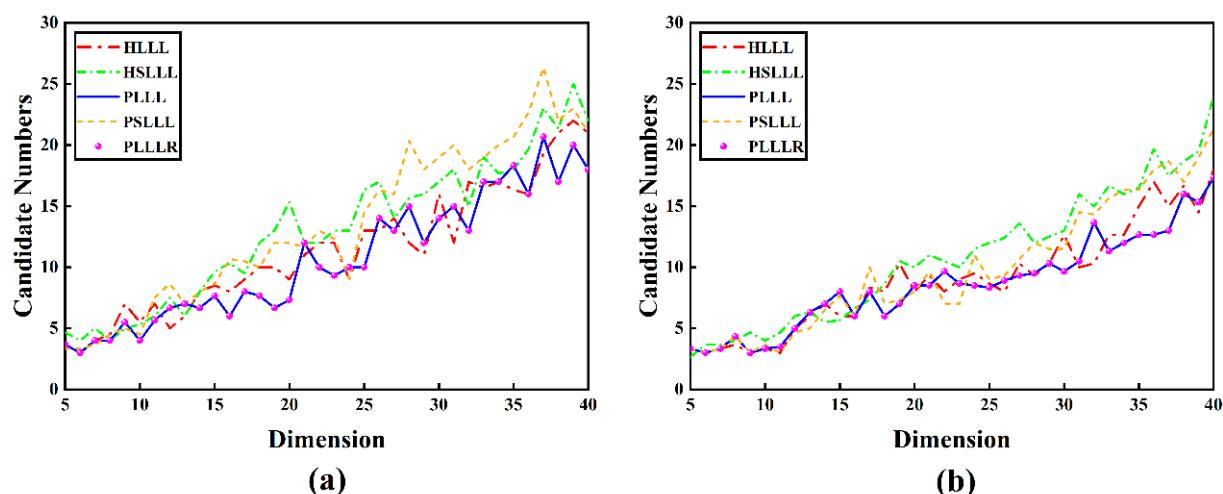
**Figure 2.** (a) Number of basis vector swaps in different dimensions of Scheme 1; (b) Number of basis vector swaps in different dimensions of Scheme 2.

Figure 3 shows the reduction time consumption for the five algorithms with different schemes and dimensions. It can be intuitively seen from Figure 3 that as the number of dimensions increases, the overall trend of the reduction time consumption for ambiguity is upward, and PSLLL has the smallest reduction times. From Figure 3a, it can be observed that the PLLLR reduction time consumption is lower than the HLLL, except for the 16th dimension. The PSLLL reduction time consumption is lower than the HSLLL (except for the 6th and 11th dimensions). A similar conclusion can be drawn in Figure 3b. The possible reasons for the above special cases are that the reduction time consumption is smaller in the case of lower dimensions and due to the running error of MATLAB.



**Figure 3.** (a) Reduction time consumption in different dimensions of Scheme 1; (b) Reduction time consumption in different dimensions of Scheme 2.

Figure 4 represents the number of search candidate points for the five algorithms with different schemes and dimensions. As can be seen from Figure 4, the change in the number of search candidate points and dimensions have the same trend overall, that is, the number of candidate points increase with the growth of dimensions. PLLR and PLL have the same number of search candidate points, whereas the number of candidate points for the ambiguity of HSLLL and PSLLL is more than HLLL and PLL in most dimensions compared to the other methods, which indicates that it may be more time consuming in the ambiguity search process.



**Figure 4.** (a) The number of search candidate points in different dimensions of Scheme 1; (b) The number of search candidate points in different dimensions of Scheme 2.

In analyzing the results of Figure 4, since the simple size reduction does not change the candidate integer vector for the ambiguity search, the number of candidate points for the search of PLLL is equivalent to the search results of the PLLLR algorithm, which is consistent with the theory. Since HSLLL and PSLLL adopt different basis vector exchange conditions from the regular LLL algorithms, the column exchange operation is reduced on the basis vector exchange, thus speeding up the lattice basis reduction procedure. Therefore, the final basis vector lengths obtained are different from those of HLLL and PLLL (the basis vector is obtained by exchanging them in a certain order), which results in a different number of search candidate vectors for HSLLL and PSLLL in different dimensions compared to HLLL and PLLL.

The minimum angle  $\theta$  and Hermite factor  $\kappa$  between the basis vectors after the reduction of Schemes 1 and 2 using the HLLL, HSLLL, PLLL, PSLLL, and PLLLR algorithms are listed in Tables 1 and 2, respectively. As can be seen from the average basis vectors of the five algorithms in Table 1 with minimum angles  $\theta$ , all five algorithms in Scheme 1 have good reduction effects in general. Considering the extent of the orthogonality of the basis vectors, the HSLLL reduction performance is optimal, followed by PLLLR, PLLL, and PSLLL, and HLLL is the worst. Similar conclusions to Table 1 can be drawn from Table 2, but with slight differences in terms of the reduction performance advantages and disadvantages, with PLLLR being the best, followed by PSLLL, HSLLL, and PLLL, and HLLL being the worst. The reason for this difference is that HSLLL and PSLLL are less stable compared to PLLLR. The minimum value of the PSLLL algorithm in Table 1 is  $41.3540^\circ$ , which fluctuates a lot, which means that the orthogonal performance will be poor, whereas both HSLLL and PLLL's minimum values are greater than  $45^\circ$  and the reduction performance is more stable. Similarly, the same is true for HSLLL in Table 2, which will not be explained here. Combining Tables 1 and 2, PLLLR is superior in terms of the stability and extent of orthogonality combined.

**Table 1.** Basis vector minimum angle (deg) and Hermite factor for the five algorithms of Scheme 1.

Methods	HLLL		HSLLL		PLLL		PSLLL		PLLLR	
	$\theta$	$\kappa$								
Max	65.0848	0.7704	72.4634	0.6544	68.6206	0.7537	65.6931	0.6544	68.6206	0.7537
Min	41.1187	0.6279	46.2028	0.6277	43.5084	0.6277	41.3540	0.6277	45.6056	0.6277
Mean	51.2901	0.6622	58.7290	0.6355	54.7261	0.6599	52.5564	0.6361	57.1343	0.6598

**Table 2.** Basis vector minimum angle (deg) and Hermite factor for the five algorithms of Scheme 2.

Methods	HLLL		HSLLL		PLLL		PSLLL		PLLLR	
	$\theta$	$\kappa$								
Max	67.2323	1.0223	67.5014	0.9893	67.3050	0.9893	67.3050	0.9893	67.3050	0.9893
Min	39.3641	0.7985	44.3996	0.4801	45.4427	0.6445	47.4158	0.4801	47.6255	0.6445
Mean	50.0438	0.8652	55.4014	0.7804	54.7020	0.8237	57.0028	0.7705	57.7065	0.8237

From the Hermite factor  $\kappa$  of the five algorithms in Tables 1 and 2, it can be observed that the Hermite factors of PLLLR and PLLL are basically the same, and the relative error is 0.0072%, which is negligible. This indicates that it is difficult to evaluate the performance advantages and disadvantages of the two algorithms from the Hermite factor indicator. There is little difference in the reduction performance between HSLLL and PSLLL, and both outperform the other three algorithms. HSLLL slightly outperforms PSLLL in Scheme 1, while the opposite is true for Scheme 2, which may be related to the type and randomness of the reduced basis. HLLL has the worst reduction performance.

### 3.3. Measured Experiment 1

To further validate the effectiveness of the algorithm and the reduction effect, using the GPS dual-frequency observation data from the US CORS station LWES with DSTR on 15 March 2023 (DOY-074) for 2778 epochs, the baseline length is 7.79 km and the sampling interval is 30 s. The ambiguity dilution of precision (ADOP) is usually used to evaluate the accuracy of the ambiguity resolution [42]. Figure 5 shows the variation trend of the ambiguity dimension and ADOP in DOY-074. It can be observed from Figure 5 that the ambiguity dimension of DOY-074 ranges from 12 to 22 dimensions, and the ADOP value is all less than 0.1. The dimensions of the first 200 epochs are about 20, and the ADOP value is all less than 0.06. Therefore, in this paper, we select the data of the first 200 epochs to verify the effectiveness and reduction performance of the improved algorithm.

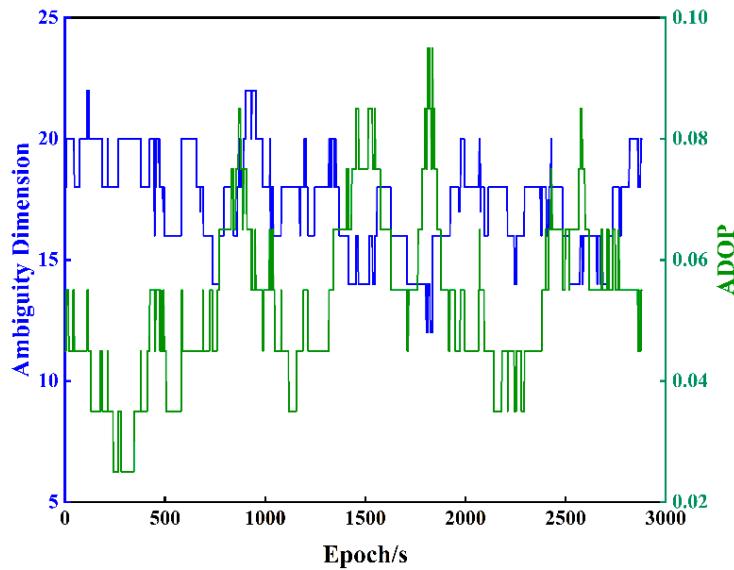
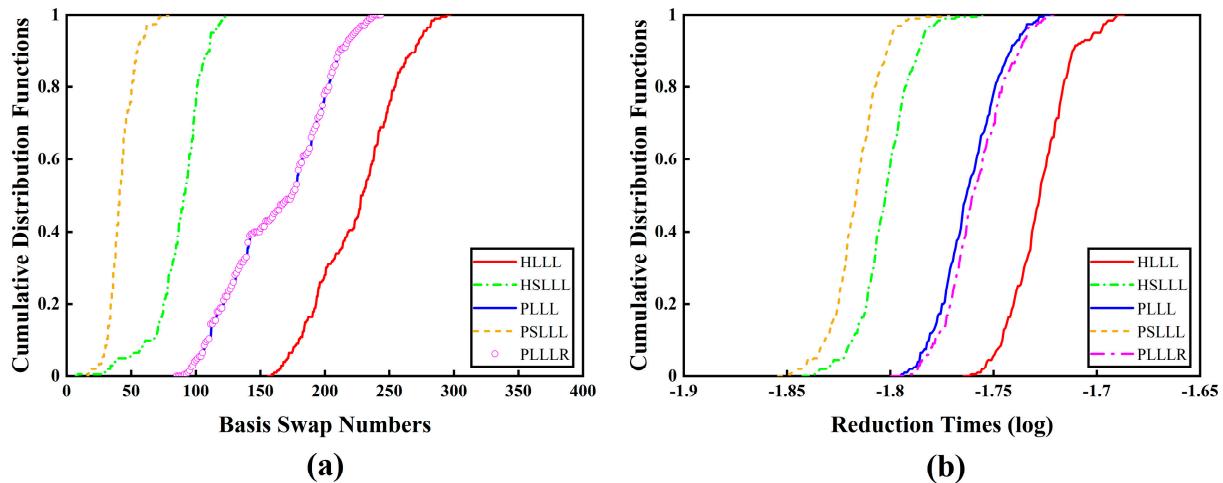
**Figure 5.** Ambiguity dimensions and ADOP values for DOY-074.

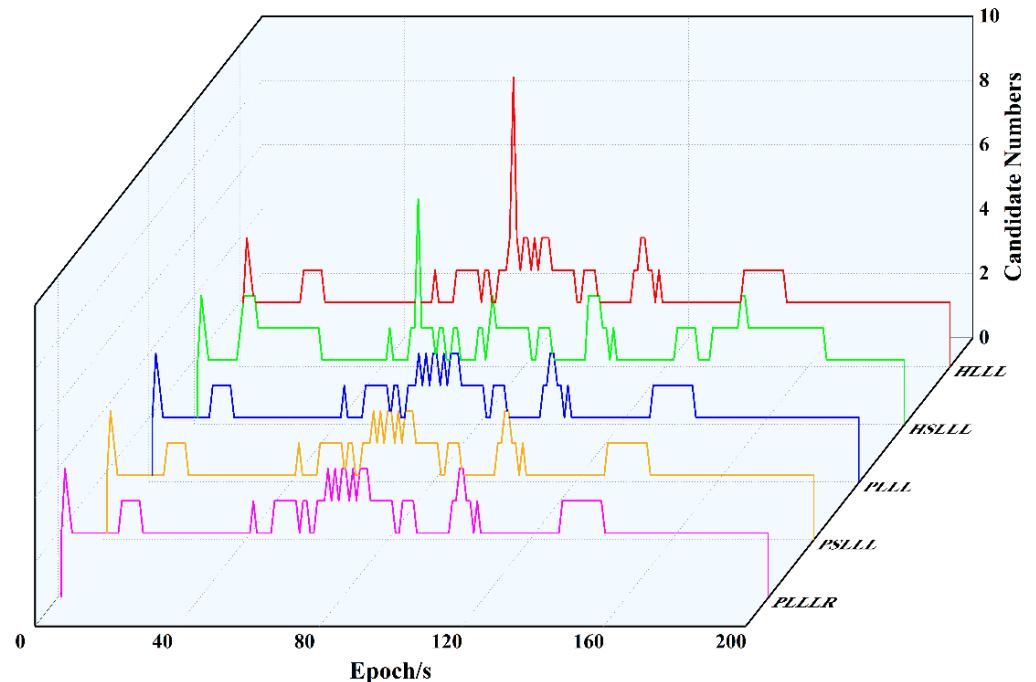
Figure 6 shows the cumulative distribution functions of the number of basis vector swaps and reduction time consumption for the first 200 epochs of the five algorithms. It can be seen from Figure 6a that PLL and PLLR have the same number of basis vector swaps, and PSLLL has the smallest number of basis vector swaps, followed by HSLLL. This is consistent with the conclusion of the simulation experiments in Section 3.2. From Figure 6b, it can be observed that the reduction time consumption of PSLLL and HSLLL

is significantly less than the other three algorithms, and PLLLR consumes slightly more reduction time than PLL due to the extra added size reduction, which is in line with the theory in Section 2.3. The five algorithms in descending order of reduction efficiency are PSLLL, HSLLL, PLL, PLLLR, and HLLL.



**Figure 6.** (a) Plot of the cumulative distribution functions of the number of basis vector swaps for the five algorithms in the first 200 epochs; (b) Plot of the cumulative distribution functions of the reduction time consumption for the five algorithms in the first 200 epochs.

Figure 7 shows the variation of the number of ambiguity candidate points for the five reduction algorithms in the first 200 epochs, from which it can be seen that the number of ambiguity candidate points is exactly the same for PLL and PLLLR, which is consistent with the conclusion of the simulation experiments, and will not be explained here. HLLL, HSLLL, and PSLLL have different numbers of candidate points for ambiguity, and the differences between HLLL and HSLLL can be clearly seen in the figure, while the overall trend of PSLLL is in line with PLL and PLLLR.



**Figure 7.** Number of ambiguity candidate points for the five algorithms in the first 200 epochs.

Table 3 shows the statistical results for the five algorithmic basis vectors' minimum angles (deg) and Hermite factors in the first 200 epochs. As seen in Table 3, the average basis vector minimum angle of all five algorithms is greater than  $45^\circ$ , and the PLLLR has the best reduction performance. From the Hermite factors  $\kappa$  of the five algorithms, it can be observed that the order of the reduction performance is consistent with Scheme 2 in the simulation experiments, PLLLR and PLL have the same Hermite factor, and PSLLL outperforms several other methods.

**Table 3.** Basis vector minimum angle (deg) and Hermite factor for five algorithms of Measured Experiment 1.

Methods	HLLL		HSLLL		PLL		PSLLL		PLLLR	
	$\theta$	$\kappa$								
Max	52.0347	2.1952	55.8564	2.0917	53.4141	2.1952	52.1806	2.0917	56.5125	2.1952
Min	41.0043	1.1528	44.0579	1.1527	42.1093	1.1528	41.4327	1.1527	45.8418	1.1528
Mean	45.5869	1.7399	50.3149	1.7338	47.6742	1.7383	46.8228	1.7331	50.7791	1.7383

Table 4 represents the solution time consumption (reduction time consumption, search time consumption, and total time consumption) of the five algorithms for the first 200 epochs, from which it can be observed that the five algorithms have the highest overall efficiency in the order of PSLLL, HSLLL, PLLLR, PLL, and HLLL. The PSLLL has the highest overall efficiency. The PLLLR algorithm has the highest search efficiency by further size reduction and has the best stability, which is favorable for improving the search efficiency of ambiguity.

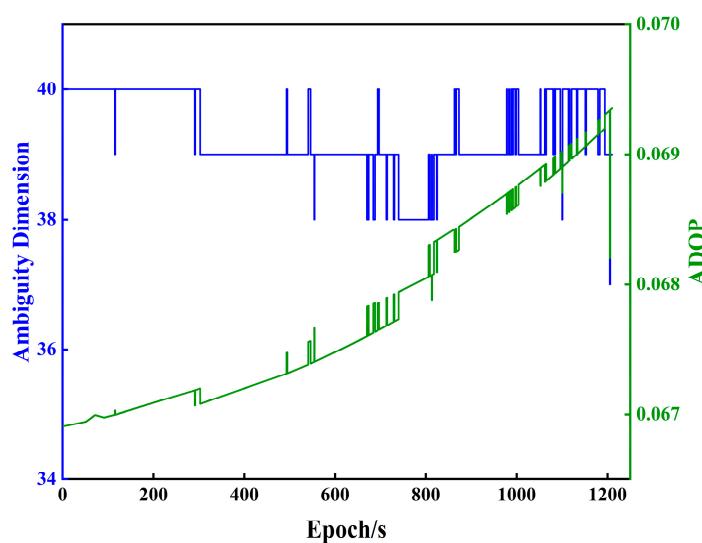
**Table 4.** Statistical results of five algorithms' resolution times for Measured Experiment 1 (ms).

Time	Methods	HLLL	HSLLL	PLL	PSLLL	PLLLR
Reduction	mean	18.7527	15.7789	17.2777	15.3097	17.3245
	max	20.4329	18.8352	18.6567	18.9781	18.4936
Search	mean	2.5271	2.2992	1.6384	2.0732	0.9467
	max	4.9388	4.3730	2.7543	4.7754	2.3275
Total	mean	21.2798	18.0781	18.9161	17.3829	18.2712
	max	24.8507	22.8804	20.8045	22.3506	20.5054

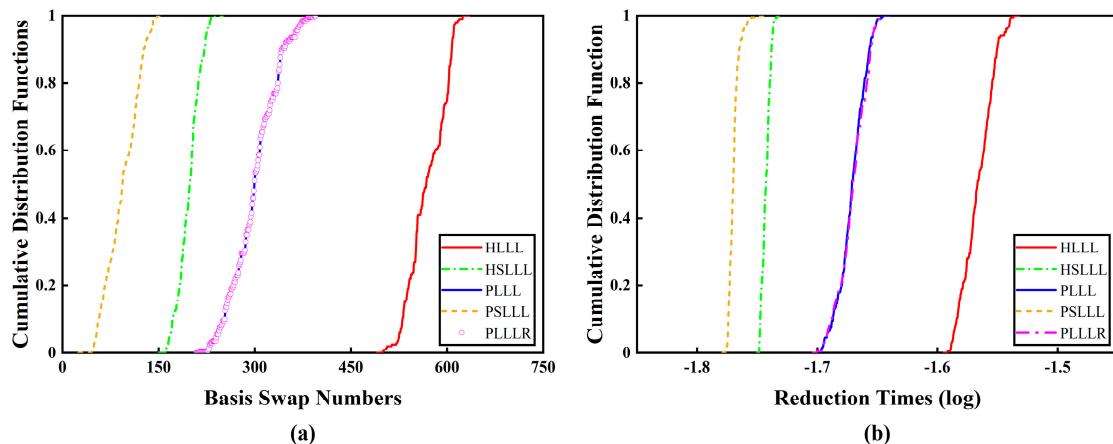
### 3.4. Measured Experiment 2

In order to further verify the reduction performance of the algorithm in the case of multiple GNSS systems and higher dimensionality, the simulated railroad track measured GPS/BDS data of 1210 epochs from Southwest Jiaotong University on 16 August 2023 (DOY-228) are selected, with a baseline length of 9.80 m and a sampling interval of 1 s. Figure 8 shows the trend plot of the ambiguity dimension and ADOP for 1210 epochs. From the figure, it can be seen that the number of ambiguity dimension is greater than 36 and the value of ADOP is less than 0.07. Therefore, the accuracy of the float solution of the ambiguity is better.

Figure 9 shows the cumulative distribution functions of the number of basis vector swaps and the reduction time consumption for the five algorithms, from which it can be observed that PLL and PLLLR have the same number of basis vector swaps. As the number of ambiguity dimensions is close to 40, the variation of the reduction time consumption of PSLLL and HSLLL is small, and the reduction time consumption of PSLLL and HSLLL is significantly smaller than the other three algorithms. The reduction time consumption of PLLLR adding extra size reduction does not increase significantly compared with PLL, which is due to the extra size reduction with low complexity, and the time consumption is basically negligible. There is no difference in the trend of the number of ambiguity candidate points of the five algorithms, which will not be shown here.



**Figure 8.** Trend of ambiguity dimension and ADOP for 1210 epochs of DOY-228.



**Figure 9.** (a) Plot of the cumulative distribution functions of the number of basis vector swaps for the five algorithms; (b) Plot of the cumulative distribution functions of the reduction time consumption for the five algorithms.

Table 5 shows the basis vector minimum angle (deg) and Hermite factor of the five algorithms. It can be seen that the average basis vector minimum angle  $\theta$  of the five algorithms is greater than  $45^\circ$ , and all of them have good reduction effects. The minimum value of  $\theta$  of the PLLLR algorithm is greater than  $45^\circ$ , which indicates that PLLLR has the best robustness in avoiding the reduced basis of poor orthogonality. The Hermite factor  $\kappa$  of PLLLR and PLLL is basically the same, and the relative error is negligible. The superiority of the PLLLR and PLLL algorithms cannot be judged from the Hermite factor  $\kappa$  alone.

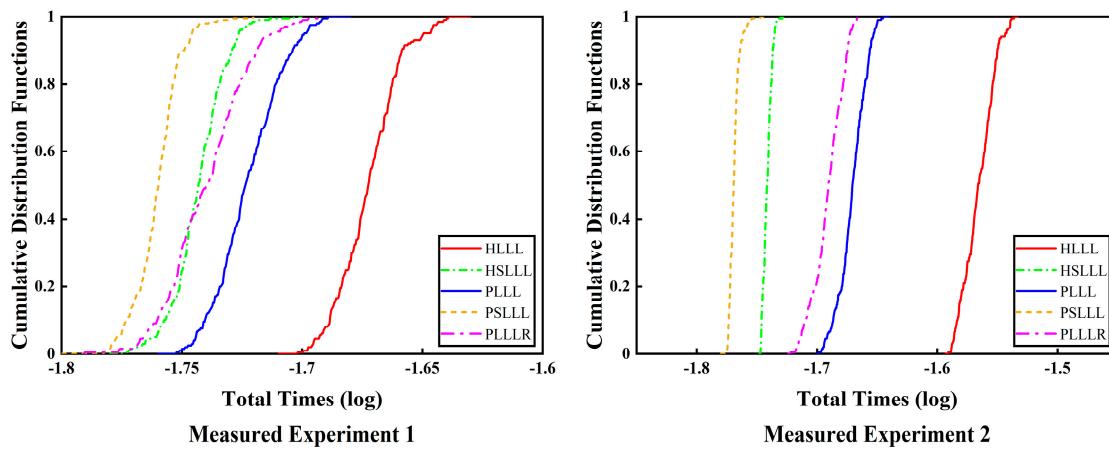
**Table 5.** Basis vector minimum angle (deg) and Hermite factor for five algorithms of Measured Experiment 2.

Methods	HLLL		HSLLL		PLLL		PSLLL		PLLLR	
	$\theta$	$\kappa$								
Max	53.1254	2.1676	55.1371	2.0435	56.4574	2.1676	55.6064	2.0434	58.1738	2.1675
Min	42.2783	1.1366	43.4854	1.1364	44.0098	1.1365	43.7539	1.1364	46.1105	1.1365
Mean	46.0438	1.7013	48.5130	1.6954	48.1327	1.6977	49.6103	1.6912	51.0883	1.6977

Table 6 shows the solution time consumption of the five algorithms (reduction time consumption, search time consumption, and total time consumption), from which the conclusions are consistent with those of Table 4 in Measured Experiment 1, and will not be repeated here. Figure 10 illustrates the cumulative distribution functions of the total time consumed for the two measured experiments, from which it can be seen that the HSLLL, PSLLL, and PLLLR algorithms outperform the HLLL and PLL. The difference is that it is not possible to ascertain the performance of the HSLLL and PLLLR algorithms from Measured Experiment 1, whereas Measured Experiment 2 clearly shows that the efficiency of the HSLLL outperforms that of the PLLLR. The possible reasons for this difference are related to the number of ambiguity dimensions and MATLAB running errors.

**Table 6.** Statistical results of five algorithms' resolution times for Measured Experiment 2 (ms).

Time	Methods	HLLL	HSLLL	PLL	PSLLL	PLLLR
Reduction	mean	27.1813	18.0880	21.3378	17.8412	21.3639
	max	29.0811	28.6279	22.6296	27.8739	22.5280
Search	mean	3.8941	3.3249	2.8544	2.9592	1.7886
	max	6.1841	5.2118	3.2176	5.1017	2.8402
Total	mean	31.0754	21.4129	24.1922	20.8004	23.1525
	max	34.2219	30.5801	25.6345	29.6266	24.5172



**Figure 10.** Plot of the cumulative distribution functions of the total time consuming of the five algorithms for the two measured experiments.

In order to illustrate the performance difference between HLLL, HSLLL, PLL, PSLLL, and PLLLR more clearly, we compare the speed, stability, and computational complexity of the five algorithms, and the results are shown in Table 7.

**Table 7.** Comparison of the five algorithms.

Method	HLLL	HSLLL	PLL	PSLLL	PLLLR
Reduction speed	Slow	Faster	Fast	Fastest	Faster
Search speed	Slow	Fast	Faster	Faster	Fastest
Stability	Good	Good in most cases	Better	Good in most cases	Best
Complexity	$o\left(\frac{4}{3}n^3 + n^5 + n^4 \log \frac{\alpha}{\beta}\right)^*$	Same as HLLL	Same as HLLL	Same as HLLL	$o\left(\frac{7}{3}n^3 + n^5 + n^4 \log \frac{\alpha}{\beta}\right)^*$

\*  $\alpha = \max\|g_i\|$  and  $\beta = \min\|G^{-T}a\|$ .

#### 4. Discussion

The classical LLL algorithm is based on the QR decomposition of the basis matrix by GSO. The computational complexity of GSO is  $2n^3$ , while the Householder QR decomposi-

tion does not require the formation of orthogonality factor  $Q$  during the reduction process, and the computational complexity is  $4/3n^3$ . In addition, the GSO method has poor numerical properties because there is usually a severe loss of orthogonality in the computation of the orthogonality factor  $Q$ . Therefore, in this paper, the Householder QR decomposition is utilized instead of the conventional GSO decomposition, which has lower computational complexity and better numerical stability. We propose corresponding improved algorithms based on the HLLL and PLLL algorithms, and verify the validity of the methods and the performance of the reduction through simulation and measured experiments. Figure 2, Figure 6a, and Figure 9a represent the number of basis vector swaps for the simulation and measured experiments, from which it can be seen that PSLLL has the smallest number of basis vector swaps, and PLLLR has the same number of basis vector swaps as PLLL. From the reduction time consumption of different algorithms in Figure 3, Figure 6b, and Figure 9b, it can be seen that the reduction time consumption of PSLLL and HSLLL is less than the other algorithms, whereas PLLLR makes the reduction time slightly higher than PLLL due to the extra size reduction. Combined with the cumulative distribution functions of the total time consumption (the sum of the reduction time and the search time) of the two measured experiments in Figure 10, it can be seen that although the PLLLR reduction time consumption is slightly higher than that of PLLL, the further size reduction of the R-matrix after exchanging the basis vectors greatly shortens the search time, which improves the overall efficiency of the ambiguity resolution. Figures 4 and 7 analyze the five algorithms from the comparison of the number of ambiguity candidate points. The number of ambiguity candidate points is exactly the same for PLLLR and PLLL, while the other algorithms are slightly different. Tables 4 and 6 show the solution time consumption of the five algorithms, where PSLLL has the fastest reduction efficiency and PLLLR has the best stability. Table 7 summarizes the performance differences of the five algorithms.

There is a close correlation between lattice basis orthogonality and basis vector length, and the ambiguity lattice basis reduction is precise to better solve the CVP on the lattice. In measuring the performance of the lattice basis reduction, the extent of orthogonality between the reduced basis vector is not intuitively determined due to the orthogonal defect indicator. We introduce the minimum angles  $\theta$  among the reduced basis vectors as an alternative to overcome the drawbacks of the orthogonal defect. It can be found through Equation (20) that a good lattice basis reduction algorithm should ensure that  $\theta$  is greater than  $45^\circ$ . As can be seen from the minimum angles among the basis vectors in Tables 1–3 and Table 5, the minimum values of  $\theta$  for the PLLLR algorithm are all greater than  $45^\circ$ , which suggests that it is the most robust in terms of avoiding a reduced basis with poor orthogonality. The Hermite factors can well reflect the property of the first basis vector of the lattice basis reduction, that is, whether the length of the first orthogonal basis vector is short enough. From the Hermite factors statistics in Tables 1–3 and Table 5, it can be observed that HSLLL and PSLLL outperform the other three algorithms in terms of reduction performance. PLLLR and PLLL have basically the same Hermite factors with negligible relative errors. It is difficult to evaluate their performance advantages or disadvantages from the Hermite factors.

## 5. Conclusions

In this paper, for the characteristics of the high dimensionality and high accuracy of ambiguity resolution, based on analyzing the LLL reduction algorithm, we introduce the minimum column pivoting Householder QR decomposition, partial size reduction, and the relaxation of the basis vector exchange condition to improve the regular LLL algorithm. In order to visualize the extent of the orthogonality of the basis vectors, the minimum angle of the basis vectors is used to replace the conventional degree of orthogonality defects, and the quality of the reduced basis size reduction is evaluated by the Hermite factor. Based on the simulation and measured data to verify the effectiveness of the improved algorithm and the reduction effect, the experimental results show that the improved algorithm effectively reduces the size reduction and the number of basis vectors exchanged

in the process of lattice basis reduction, and can obtain a better reduction effect, which significantly improves the reduction performance of the LLL algorithm. HSLLL and PSLLL have a better reduction effect, but are poor in the stability performance of lattice base reduction. The PLLR algorithm loses a small amount of reduction time, but improves the search efficiency of the ambiguity, which effectively improves the overall efficiency of the ambiguity resolution.

**Author Contributions:** Conceptualization, X.L. and Y.X.; methodology, X.L. and Y.X.; software, X.L.; validation, X.L., Y.X., W.C., S.X. and R.Z.; formal analysis, X.L. and Y.X.; investigation, Y.X.; resources, X.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L. and Y.X.; visualization, X.L.; supervision, Y.X.; project administration, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Sichuan Science and Technology Program (Funder Name: Yongliang Xiong. Grant No. 2022YFG0169), and the National Natural Science Foundation of China (Funder Name: Yongliang Xiong. Grant No. 41674028 and 41274044).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset supporting this research can be found at the NOAA's National Geodetic Survey (NGS).

**Acknowledgments:** The authors thank the NOAA's National Geodetic Survey (NGS) for their products and datasets, as well as the researchers who provided the open-source software.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Teunissen, P.J.G. Least-squares estimation of the integer GPS ambiguities. In Proceedings of the Invited Lecture, Section IV Theory and Methodology, IAG General Meeting, Beijing, China, 8–13 August 1993; p. 16.
2. Teunissen, P.J.G. The least-square ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation. *J. Geod.* **1995**, *70*, 65–82. [[CrossRef](#)]
3. Liu, L.T.; Hsu, H.T.; Zhu, Y.Z.; Ou, J.K. A new approach to GPS ambiguity decorrelation. *J. Geod.* **1999**, *73*, 478–490. [[CrossRef](#)]
4. Xu, P. Random simulation and GPS decorrelation. *J. Geod.* **2001**, *75*, 408–423. [[CrossRef](#)]
5. Xu, P. Cholesky-based reduction for the weighted integer least squares problem. *J. Geod.* **2012**, *86*, 35–52. [[CrossRef](#)]
6. Zhou, Y. A new practical approach to GNSS high-dimensional ambiguity decorrelation. *GPS Solut.* **2011**, *15*, 325–331. [[CrossRef](#)]
7. Zhou, Y.; He, Z. ariance reduction of GNSS ambiguity in (inverse) paired Cholesky decorrelation transformation. *GPS Solut.* **2013**, *18*, 1–9.
8. Chang, X.W.; Yang, X.; Zhou, T. MLAMBDA: A modified LAMBDA method for integer least-squares estimation. *J. Geod.* **2005**, *79*, 552–565. [[CrossRef](#)]
9. Liu, J.; Yu, X.; Zhang, X. GNSS ambiguity resolution using the lattice theory. *Acta Geod. Cartogr. Sin.* **2012**, *41*, 636–645.
10. Lv, H. *Research on Key Methods of GNSS Integer Ambiguity Estimation*; Information Engineering University: Zhengzhou, China, 2019.
11. Lenstra, A.K.; Lenstra, H.W.; Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann.* **1982**, *261*, 515–534. [[CrossRef](#)]
12. Hassibi, A.; Boyd, S. Integer parameter estimation in linear models with applications to GPS. *IEEE Trans. Signal Process.* **1998**, *46*, 2938–2952. [[CrossRef](#)]
13. Grafarend, E.W. Mixed integer-real valued adjustment (IRA) problems: GPS initial cycle ambiguity resolution by means of the LLL algorithm. *GPS Solut.* **2000**, *4*, 31–44. [[CrossRef](#)]
14. Lou, L. One modified LLL algorithm in GPS decorrelation. *J. Tongji Univ.* **2004**, *32*, 237–241.
15. Liu, Z.; He, X. An improved LLL algorithm for GPS ambiguity solution. *Acta Geod. Cartogr. Sin.* **2007**, *36*, 286–289.
16. Yang, R.; Hua, X.; Li, Z.; Wu, J. An improved LLL algorithm for GPS ambiguity solution. *Geomat. Inf. Sci. Wuhan Univ.* **2010**, *35*, 21–24.
17. Fan, L.; Zhai, G.; Chai, H. Ambiguity decorrelation with Integer block orthogonalization algorithm. *Acta Geod. Cartogr. Sin.* **2014**, *43*, 818–826.
18. Xie, K.; Chai, H.; Fan, L.; Pan, Z. An improved LLL ambiguity decorrelation algorithm. *Geomat. Inf. Sci. Wuhan Univ.* **2014**, *39*, 1363–1368.
19. Jazaeri, S.; Amiri-Simkoeei, A.R.; Sharifi, M.A. Fast integer least-squares estimation for GNSS high-dimensional ambiguity resolution using lattice theory. *J. Geod.* **2012**, *86*, 123–136. [[CrossRef](#)]

20. Ling, C.; Howgrave-Graham, N. Effective LLL reduction for lattice decoding. In Proceedings of the IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007.
21. Xie, X.; Chang, X.W.; Borno, M.A. Partial LLL reduction. In Proceedings of the IEEE GLOBECOM 2011, Houston, TX, USA, 5–9 December 2011.
22. Lu, L.; Liu, W.; Li, J. An effective LLL reduction algorithm. *Geomat. Inf. Sci. Wuhan Univ.* **2016**, *41*, 1118–1124.
23. Lv, H.; Lv, Z.; Zhai, S. Improved LLL ambiguity reduction algorithm. *J. Chin. Inert. Technol.* **2017**, *25*, 611–617.
24. Li, K.; Tian, C.; Jiao, Y.; Yue, Z. Improved HLLL lattice basis reduction algorithm to solve GNSS integer ambiguity. *Int. J. Aerosp. Eng.* **2023**, *2023*, 1–8. [[CrossRef](#)]
25. Lannes, A. On the theoretical link between LLL-reduction and LAMBDA-decorrelation. *J. Geod.* **2013**, *87*, 323–335. [[CrossRef](#)]
26. Borno, M.A.; Chang, X.W.; Xie, X.H. On ‘decorrelation’ in solving integer least-squares problems for ambiguity determination. *Emp. Surv. Rev.* **2014**, *46*, 37–49. [[CrossRef](#)]
27. Jazaeri, S.; Amiri-Simkoeei, A.R.; Sharifi, M.A. On lattice reduction algorithms for solving weighted integer least squares problems: Comparative study. *GPS Solut.* **2014**, *18*, 105–114. [[CrossRef](#)]
28. Lu, L.; Liu, W.; Li, J. Impact of decorrelation on search efficiency of ambiguity resolution. *Acta Geod. Cartogr. Sin.* **2015**, *44*, 481–487.
29. Teunissen, P.J.G. A new method for fast carrier phase ambiguity estimation. In Proceedings of the Position Location & Navigation Symposium, Las Vegas, NV, USA, 11–15 April 1994.
30. Teunissen, P.J.G. Integer least-squares theory for the GNSS compass. *J. Geod.* **2010**, *84*, 433–447. [[CrossRef](#)]
31. Xu, P.; Cannon, E.; Lachapelle, G. Mixed integer programming for the resolution of GPS carrier phase ambiguities. *arXiv* **2010**, arXiv:1010.1052.
32. Teunissen, P.J.G. The LAMBDA method for the GNSS compass. *Artif. Satell.* **2006**, *41*, 89–103. [[CrossRef](#)]
33. Liu, W.; Lu, L.; Shan, H. A new block processing algorithm of LLL for fast high-dimension ambiguity resolution. *Acta Geod. Et Cartogr. Sin.* **2016**, *45*, 147–156.
34. Schnorr, C.P. Fast LLL-type lattice reduction. *Inf. Comput.* **2006**, *204*, 1–25. [[CrossRef](#)]
35. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 4th ed.; Johns Hopkins University Press: Baltimore, MD, USA, 2013.
36. Gestner, B.; Zhang, W.; Ma, X.; Anderson, D.V. Lattice reduction for MIMO detection: From theoretical analysis to hardware realization. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2011**, *58*, 813–826. [[CrossRef](#)]
37. Nguyen, P.Q.; Stehlé, D. LLL on the average. In Proceedings of the Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, 23–28 July 2006.
38. Xu, P. Experimental quality evaluation of lattice basis reduction methods for decorrelating low-dimensional integer least squares problems. *Eurasip J. Adv. Signal Process.* **2013**, *2013*, 1–29. [[CrossRef](#)]
39. Teunissen, P.J.G. Success probability of integer GPS ambiguity rounding and bootstrapping. *J. Geod.* **1998**, *72*, 606–612. [[CrossRef](#)]
40. Nguyen, P.Q.; Vallée, B. *The LLL Algorithm: Survey and Applications*; Springer Publishing Company: Berlin, Germany, 2010.
41. Fontein, F.; Schneider, M.; Wagner, U. PotLLL: A polynomial time version of LLL with deep insertions. *Des. Codes Cryptogr.* **2014**, *73*, 355–368. [[CrossRef](#)]
42. Odijk, D.; Teunissen, P.J.G. ADOP in closed form for a hierarchy of multi-frequency single-baseline GNSS models. *J. Geod.* **2008**, *82*, 473–492. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.