J Geod (2012) 86:123–136
DOI 10.1007/s00190-011-0501-z

ORIGINAL ARTICLE

Fast integer least-squares estimation for GNSS high-dimensional ambiguity resolution using lattice theory

S. Jazaeri · A. R. Amiri-Simkooei · M. A. Sharifi

Received: 14 December 2010 / Accepted: 15 July 2011 / Published online: 30 July 2011

Abstract GNSS ambiguity resolution is the key issue in the high-precision relative geodetic positioning and navigation applications. It is a problem of integer programming plus integer quality evaluation. Different integer search estimation methods have been proposed for the integer solution of ambiguity resolution. Slow rate of convergence is the main obstacle to the existing methods where tens of ambiguities are involved. Herein, integer search estimation for the GNSS ambiguity resolution based on the lattice theory is proposed. It is mathematically shown that the closest lattice point problem is the same as the integer least-squares (ILS) estimation problem and that the lattice reduction speeds up searching process. We have implemented three integer search strategies: Agrell, Eriksson, Vardy, Zeger (AEVZ), modification of Schnorr–Euchner enumeration (M-SE) and modification of Viterbo-Boutros enumeration (M-VB). The methods have been numerically implemented in several simulated examples under different scenarios and over 100 independent runs. The decorrelation process (or unimodular transformations) has been first used to transform the original ILS problem to a new one in all simulations. We have then applied different search algorithms to the transformed ILS problem. The numerical simulations have shown that AEVZ, M-SE, and M-VB are about 320, 120 and 50 times faster than LAMBDA, respectively, for a search space of dimension 40. This number could change to about 350, 160 and 60 for dimension 45. The AEVZ is shown to be faster than MLAMBDA by a factor of 5. Similar conclusions could be made using the application of the proposed algorithms to the real GPS data.

Keywords Integer least-squares estimation · GNSS ambiguity resolution · Lattice theory · Pohst enumeration · Schnorr–Euchner enumeration

Electronic supplementary material The online version of this article (doi:10.1007/s00190-011-0501-z) contains supplementary material, which is available to authorized users.

S. Jazaeri (✉) · M. A. Sharifi
Department of Surveying and Geomatics Engineering,
College of Engineering, University of Tehran, Tehran, Iran
e-mail: jazayeri@ut.ac.ir

M. A. Sharifi
e-mail: sharifi@ut.ac.ir

A. R. Amiri-Simkooei
Department of Surveying Engineering, Faculty of Engineering,
University of Isfahan, 81746-73441 Isfahan, Iran
e-mail: ar_amiri@yahoo.com

1 Introduction

High-precision GNSS positioning is achieved using the carrier phase observables in the relative positioning mode. GNSS relative positioning is used for many high-precision applications such as surveying, mapping, GIS, and precise navigation. A prerequisite to this is the successful determination of the integer double difference carrier phase ambiguity parameters. Mathematically, double difference carrier phase observation equation is a mixed integer nonlinear model. Linearizing the carrier phase observation equation yields the following mixed integer linear model (Teunissen 1995; Xu et al. 1995; Xu 2006):

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{y}$ is a t-dimensional vector of observed minus approximate double difference carrier phase observations, $\boldsymbol{a}$ is an n-dimensional integer vector, $\boldsymbol{b}$ is an m-dimensional real-valued vector, $\boldsymbol{e}$ is the error vector of observations, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the $t \times n$ and $t \times m$ real-valued matrices, respectively.

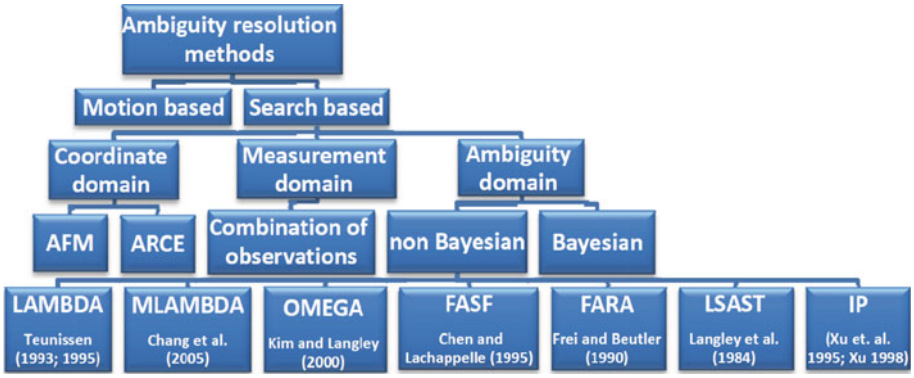

Fig. 1 Some of carrier phase ambiguity resolution methods available in literature

Applying the least-squares criterion to (1), to estimate the unknown parameters, yields

$$\min(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{a}-\boldsymbol{B}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{a}-\boldsymbol{B}\boldsymbol{b}) \tag{2}$$

where $\boldsymbol{P}$ is the weight matrix of observables. This minimization problem, which is a type of mixed integer nonlinear programming (MINLP) problem, is also referred to as a mixed integer least-squares (ILS) problem. The ambiguity parameters are initially determined as part of the ordinary least-squares by neglecting the integerness of ambiguities. What can be obtained are the real-valued parameters $\left(\begin{bmatrix}\hat{\boldsymbol{a}}\\ \hat{\boldsymbol{b}}\end{bmatrix}\right)$ along with their covariance matrix $\left(\begin{bmatrix}Q_{\hat{a}} & Q_{\hat{a}\hat{b}}\\ Q_{\hat{b}\hat{a}} & Q_{\hat{b}}\end{bmatrix}\right)$. The minimization problem in (2) is equivalent to the following problem (Teunissen 1993; Xu et al. 1995):

$$\check{\boldsymbol{a}} = \min(\hat{\boldsymbol{a}}-\boldsymbol{a})^{\mathrm{T}}\boldsymbol{P}(\hat{\boldsymbol{a}}-\boldsymbol{a}) \tag{3}$$

which is the standard ILS or the integer quadratic programming (IQP) problem. It is clear that the solution to the original mixed ILS problem (2) depends on the solution of ILS (3). Denoting the integer solution of (3) by $\check{\boldsymbol{a}}$, one can then obtain the least-squares estimates of the real parameters $\check{\boldsymbol{b}}$ as

$$\check{\boldsymbol{b}} = (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{y}-\boldsymbol{A}\check{\boldsymbol{a}}) \tag{4}$$

Various methods have been developed in the past to deal with carrier phase ambiguity resolution. Figure 1 shows some of the well known strategies considered so far in literature.

On the whole, ambiguity resolution methods can be classified into two main categories, namely motion-based and search-based methods (Buist 2007). Motion-based category takes advantage of the information contained in the changes in visible GNSS satellites or the motion of the platform, i.e. changes in receiver-satellite geometry. This motion-based method takes time to ambiguity solution and requires at least three non-coplanar baselines (Cohen 1996). Therefore, they are not applicable for real time GNSS positioning and the latter category, i.e. search-based methods are always selected because these methods are not necessarily dependent on motion.

Search-based methods are classified into three groups: searching in coordinate domain, measurement domain, and ambiguity domain. Two types of solutions in ambiguity domain are called Bayesian and non-Bayesian. There are several non-Bayesian methods for ambiguity resolution in ambiguity domain. We can at least mention least-squares ambiguity search technique (LSAST) proposed by Langley et al. (1984) and Wei (1986), fast ambiguity resolution approach (FARA) proposed by Frei and Beutler (1990), least-squares ambiguity decorrelation adjustment (LAMBDA) proposed by Teunissen (1993, 1994, 1995), modified LAMBDA method proposed by Chang et al. (2005), fast ambiguity search filter (FASF) proposed by Chen (1994) and Chen and Lachapelle (1995), integer programming (IP) proposed by Xu et al. (1995) and Xu (1998) and optimal method for estimating GPS ambiguities (OMEGA) proposed by Kim and Langley (2000).

In mathematical language, ambiguity resolution is a problem of IP plus integer quality evaluation. We will discuss the first problem in this paper from the point of view of the closest lattice point (CLP) in lattice theory. The study of lattices was originally motivated by the geometry of numbers. A lattice L is a discrete additive subgroup of $\mathbb{R}^d$, generated by the set of integer linear combinations of no more than d vectors $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_k}$, where $k \leq d$. If these vectors are linearly independent, we say that they are a basis of the lattice L. Lattice basis reduction is the computation of lattice bases where the base vectors are not only as orthogonal as possible to each other, but also as short as possible. This is an important problem in geometry of numbers with applications in communications, combinatorial optimization such as IP, computer algebra and cryptography (see, for example Kannan 1987; Steinfeld et al. 2007).

One of the most famous problem related to lattices is the CLP problem: given a lattice basis and a target vector in $\mathbb{R}^d$, find a lattice vector that is closest to the target. The CLP problem has applications in various fields, including number theory, cryptography and communication theory (Agrell et al. 2002). This contribution presents another application of the CLP problem to the problem of integer ambiguity resolution.

In addition to the US system GPS, the Russian GLONASS is also in place and the European Galileo, Chinese COMPASS, Japanese QZSS are currently under construction and will also transmit multiple frequency signals. The increase in satellite availability and transmitted signals will increase the number of ambiguity parameters. Many methods have been developed to increase the search efficiency. However, when dealing with high-dimensional ambiguity parameters, the existing methods are still slow. The goal of this paper is to apply the popular methods taken from the mathematical field of lattice theory to integer search estimation for GNSS ambiguity resolution. We test out these strategies in the rest of the paper and compare them to the LAMBDA and MLAMBDA methods. Numerical results indicate substantial improvement in speed to the problem of ILS.

The remaining of this paper is organized as follows. In Sect. 2, we introduce the closest point in lattice versus integer ambiguity resolution problem. It is shown how the ILS problem can be converted to the problem of the closest point in the lattice. Section 3 presents the implementation of the lattice theory and mathematically shows lattice basis reduction speeds up searching process. Section 4 gives Pohst and Schnorr–Euchner enumerations. It then presents three searching closest point algorithms based on the Pohst and Schnorr–Euchner enumerations. We give numerical simulated and real GPS experiment results in Sect. 5. Conclusions are presented in Sect. 6.

2 Lattice theory versus integer ambiguity resolution

2.1 Introduction to lattices

In geometry of numbers, a lattice in $\mathbb{R}^d$ is a discrete, additive, abelian subgroup of $\mathbb{R}^d$ consisting of points. Discrete signifies that there are no cluster points but all points have a minimum Euclidean distance from each other. Let the vectors $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_k} \in \mathbb{R}^d, k \leq d$ be linear independent. The set

$$\Lambda = \left\{ \boldsymbol{u} \in \mathbb{R}^d \,\middle|\, \boldsymbol{u} = \sum_{i=1}^{k} a_i \boldsymbol{b_i}\ ,\ a_i \in \mathbb{Z} \right\} \tag{5}$$

is called a lattice, where k is the rank of the lattice and d is the dimension of its basis vector. Therefore, every lattice can be represented by a set $\boldsymbol{B} = \{\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_k}\}$ of its basis vectors called the basis of the lattice. For a lattice with $k = d$, we refer to as a full rank lattice. In the matrix form the lattice can be represented as follows:

$$\Lambda(\boldsymbol{B}) = \{\boldsymbol{Ba}: \quad \boldsymbol{a} \in \mathbb{Z}^k\} \tag{6}$$

The addition of vectors is associative and commutative, and further a lattice is an abelian group. Two lattices with the basis matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are called identical if all points of the two lattices are the same. A lattice basis is not unique

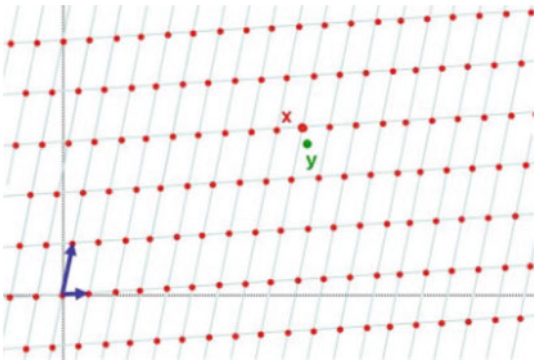

Fig. 2 The closest lattice point x to a given input point y

and every basis $\boldsymbol{B}$ can be transformed into another basis $\boldsymbol{C}$ such that $\Lambda(\boldsymbol{B}) = \Lambda(\boldsymbol{C})$. This is achieved using a unimodular transformation for which we refer to Xu et al. (1995).

By the CLP problem, for a given input point $\boldsymbol{y} \in \mathbb{R}^d$, one searches a vector $\boldsymbol{x}$ such that

$$\|\boldsymbol{y} - \boldsymbol{x}\|^2 \leq \|\boldsymbol{y} - \boldsymbol{c}\|^2 \quad \forall \boldsymbol{c} \in \Lambda \tag{7}$$

Using the lattice defined with the generator matrix $\boldsymbol{B}$ in Eq. (6), there is a vector $\boldsymbol{a} \in \mathbb{Z}^k$ such that for $\boldsymbol{x} \in \Lambda, \boldsymbol{x} = \boldsymbol{Ba}$. Therefore, the CLP problem in a lattice, with a generator matrix $\boldsymbol{B}$, is equivalent to finding a vector $\boldsymbol{a} \in \mathbb{Z}^k$ such that

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{x}\|^2 = \|\boldsymbol{y} - \boldsymbol{Ba}\|^2 &= \min_{\forall \boldsymbol{c} \in \Lambda} \|\boldsymbol{y} - \boldsymbol{c}\|^2 \\ &= \min_{\forall \boldsymbol{b} \in \mathbb{Z}^k} \|\boldsymbol{y} - \boldsymbol{Bb}\|^2 \end{aligned} \tag{8}$$

Figure 2 depicts the CLP (i.e. $\boldsymbol{x}$) to a given input point $\boldsymbol{y}$.

For a d-dimensional lattice in $\mathbb{R}^d$ and for i ranging from 1 to k, the ith successive minimum $\lambda_i(\Lambda)$ is the radius of the smallest closed ball centered at the origin containing i linear independent lattice vectors. The shortest vector problem is to find a vector in $\Lambda - \{\boldsymbol{0}\}$ that has the smallest Euclidean norm.

2.2 Integer least-squares problem as a CLP problem

Consider again the ILS problem (IQP) introduced in Eq. (3). Since the covariance matrix $\boldsymbol{Q_{\hat{a}}}$ of the estimated ambiguity parameters is positive definite, the Cholesky decomposition of the matrix $\boldsymbol{P} = \boldsymbol{Q_{\hat{a}}^{-1}}$ reads

$$\boldsymbol{P} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} \tag{9}$$

which yields

$$\begin{aligned} (\hat{\boldsymbol{a}} - \boldsymbol{a})^{\mathrm{T}}\boldsymbol{P}(\hat{\boldsymbol{a}} - \boldsymbol{a}) &= (\hat{\boldsymbol{a}} - \boldsymbol{a})^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a}) \\ &= (\boldsymbol{R}\hat{\boldsymbol{a}} - \boldsymbol{Ra})^{\mathrm{T}}(\boldsymbol{R}\hat{\boldsymbol{a}} - \boldsymbol{Ra}) \end{aligned} \tag{10}$$

Assuming $\boldsymbol{R}\hat{\boldsymbol{a}} = \boldsymbol{y}$, one obtains

$$(\hat{\boldsymbol{a}} - \boldsymbol{a})^{\mathrm{T}}\boldsymbol{P}(\hat{\boldsymbol{a}} - \boldsymbol{a}) = (\boldsymbol{y} - \boldsymbol{Ra})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{Ra}) = \|\boldsymbol{y} - \boldsymbol{Ra}\|^2 \tag{11}$$

Therefore, ILS problem using Eq. (11) can be rewritten as follows

$$\breve{\boldsymbol{a}} = \min(\hat{\boldsymbol{a}} - \boldsymbol{a})^{\mathrm{T}} \boldsymbol{P} (\hat{\boldsymbol{a}} - \boldsymbol{a}) = \min \|\boldsymbol{y} - \boldsymbol{R}\boldsymbol{a}\|^2 \tag{12}$$

The preceding minimization problem is equivalent to the CLP problem of the lattice $\Lambda(\boldsymbol{R}) = \{\boldsymbol{R}\boldsymbol{a} : \ \boldsymbol{a} \in \mathbb{Z}^n\}$, with the basis matrix $\boldsymbol{R}$ and the given point $\boldsymbol{y}$. Therefore, ILS problem is in fact a CLP problem in lattice $\Lambda(\boldsymbol{R})$.

3 Background on lattice basis reduction

3.1 Lattice basis reduction

Hermite (1850) published the first lattice reduction (LR) algorithm in an arbitrary dimension by trying to generalize Lagrange's two-dimensional algorithm (Lagrange 1773). In his famous letters to Jacobi, Hermite described two reduction notions (along with algorithms) in the language of quadratic forms: the first letter presented an algorithm to show the existence of Hermite's constant (which guarantees the existence of short lattice vectors), while the second letter presented a slightly different algorithm to further prove the existence of lattice bases with bounded orthogonality defect (Nguyen and Stehlé 2009). Hermite's algorithms can be viewed as the ancestors of the Lenstra, Lenstra and Lovasz (LLL) algorithm.

In mathematics, the goal of lattice reduction is to transform a given lattice basis into another lattice basis of which its vectors have the smallest possible length and they are close to orthogonal (for lattice-search problems, this was first noted by Coveyou and Macpherson 1967). There exists no perfect lattice basis reduction algorithm because it depends on many factors like runtime, dimension of the basis, the given problem to solve, and the expected quality of the solution. A perfect algorithm should be able to handle high-dimensional lattice and provide appropriate solutions in acceptable time. Since the runtime plays an important role, a trade-off between the runtime and a good solution for a given lattice problem is necessary. For example, Korkine–Zolotareff (KZ) reduction is very strong, but expensive to compute. On the contrary, LLL reduction is fairly cheap, but an LLL-reduced basis is of much lower quality (Hanrot and Stehle 2007). The orthogonalization, size reduction, and vector swapping constitute the three fundamentals of the lattice basis reduction, which will be followed in the next subsections.

The covariance matrix of the ambiguities geometrically defines a hyper-ellipsoid centered on the float ambiguities, related to the searching space. Within a short observation span, there is a high correlation between ambiguities and the search ellipsoid may be particularly elongated (Teunissen 1996). Therefore, the search process can be very time consuming. In the GNSS literature, reduction stage is called decorrelation. To increase the searching speed for the CLP or integer ambiguities, one needs to decrease the correlation among the original ambiguities. By a reparametrization of the ambiguities which is called decorrelation or reduction, the hyper ellipsoid is transformed to an almost spheroid and consequently the searching process speeds up.

3.2 Orthogonalization

Consider the lattice basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n})$. The orthogonalization process provides the Gram–Schmidt coefficients g_{ij} and the squared 2-norms of the orthogonalized lattice basis vectors $\boldsymbol{b_i^0}$, i.e. $\|\boldsymbol{b_i^0}\|^2$, with the help of QR decomposition using the Gram–Schmidt process in the shape of $\widetilde{\boldsymbol{R}}$:

$$\widetilde{\boldsymbol{R}} = \begin{pmatrix} \|\boldsymbol{b_1^0}\|^2 & g_{2,1} & \cdots & g_{n-1,1} & g_{n,1} \\ 0 & \|\boldsymbol{b_2^0}\|^2 & g_{3,2} & \cdots & g_{n,2} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & \cdots & 0 & \|\boldsymbol{b_{n-1}^0}\|^2 & g_{n,n-1} \\ 0 & 0 & \cdots & 0 & \|\boldsymbol{b_n^0}\|^2 \end{pmatrix} \tag{13}$$

QR decomposition and the Gram–Schmidt process are described in Appendix. Basis orthogonalization is explained in Algorithm 1.

Algorithm 1 Basis Orthogonalization

Input: Lattice basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n}) \in \mathbb{R}^n$
Output: $\widetilde{\boldsymbol{R}}$
1: Computing QR decomposition by means of Gram–Schmidt process
2: **for** $i = 1 : n$
3: **for** $j = i + 1 : n$
4: $\tilde{r}_{i,j} = \frac{r_{i,j}}{r_{i,i}}$
5: **end for**
6: $\tilde{r}_{i,i} = {r^2}_{i,i}$
7: **end for**

3.3 Size reduction

A lattice basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n})$ is said to be size-reduced if its Gram–Schmidt orthogonalization coefficients g_{ij}'s all satisfy $|g_{i,j}| \leq \frac{1}{2}$ for all $1 \leq j < i \leq k$. Size reduction was introduced by Lagrange (1773).

Any basis can be converted into a size reduced basis by Algorithm 2.

Algorithm 2 Basis Size Reduction

Input : Lattice basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n}) \in \mathbb{R}^n$ and the Gram–Schmidt coefficients
Output : Size reduced basis $\boldsymbol{B}$
1: **for** $i = 2 : n$
2: **for** $j = i - 1 : n$
3: **If** $|g_{i,j}| > \frac{1}{2}$
4: $\boldsymbol{b_i} = \boldsymbol{b_i} - \text{round}(g_{i,j})\boldsymbol{b_j}$
5: $g_{i,j} = g_{i,j} - \text{round}(g_{i,j})$
6: **for** $l = 1 : j - 1 \quad g_{i,l} = g_{i,l} - \text{round}(g_{i,j})g_{j,l}$
7: **End if**
8: **end for**
9: **end for**

3.4 Vector swapping

To gain a better reduction, vector swapping is used which is a unimodulary transformation operation. The order of $\boldsymbol{b_i}$ and then the dedicated orthogonalized versions $\boldsymbol{b_i^*}$ will be changed by swapping. Algorithm 2 demonstrates that the basis vector b_k can be size reduced using the Gram–Schmidt coefficients by $k-1$ vectors, the vector $\boldsymbol{b_{k-1}}$ by $k-2$ vectors, and finally $\boldsymbol{b_2}$ by only one vector, i.e. $\boldsymbol{b_1}$. In this manner, the vector $\boldsymbol{b_1}$ itself is not size reduced. Consequently, the relative short vectors should be taken to the front and the long vectors to the end of the basis. Then, a long vector has a higher chance to get size reduced and the shortest vector cannot be size reduced at all.

It is useful to swap two vectors $\boldsymbol{b_i}$ and $\boldsymbol{b_{i+1}}$ when the dedicated orthogonalized version $\boldsymbol{b_i^*}$ gets shorter. Swap condition for consecutive vectors $\boldsymbol{b_i}$ and $\boldsymbol{b_{i+1}}$ is as follows (Lenstra et al. 1982):

$$\delta\|\boldsymbol{b_i^*}\|_2^2 \le \|\boldsymbol{b_{i+1}^*}\|_2^2 + g_{i+1,i}^2\|\boldsymbol{b_i^*}\|_2^2 \quad \frac{1}{4} < \delta \le 1 \qquad i = 1,\ldots,n-1 \tag{14}$$

LLL reduction proposed by Lenstra et al. (1982) who focused on $\delta = \frac{3}{4}$. Two types of reduction that are more frequently used in practice are Korkine–Zolotareff (KZ) reduction proposed by Korkine and Zolotareff (1873) and LLL reduction. A basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n})$ is LLL-reduced if it is size-reduced and if its Gram–Schmidt orthogonalization vectors satisfy the $(n-1)$ conditions (14). This implies that the lengths of the $\boldsymbol{b_i^*}$'s cannot decrease too fast: intuitively, the vectors are not far from being orthogonal.

The LLL reduction is often used in situations where the KZ reduction would be too time consuming and terminates in polynomial time according to the lattice dimension. One reason for their reputation is that their algorithms are recursive. The n-dimensional reduction problem can be recursively reduced to a $n-1$-dimensional reduction problem which is not feasible with Minkowski (1905) reduction. Finding good reduced bases has proved invaluable in many fields of computer science and mathematics (see for example, Cohen 1995; Grotschel et al. 1993), particularly in cryptology (see for instance, Nguyen and Stern 2001; Micciancio and Goldwasser 2002). For the issue on numerical stability for lattice basis reduction algorithms the reader can consult for instance, Nguyen and Stehlé (2004, 2009) and Pujol and Stehle (2008). Recently improved lattice basis reduction algorithms are studied in Bartkewitz (2009).

Definition 2 A Lattice is called δ-LLL-reduced when

1. it is size reduced
2. $\delta\|\boldsymbol{b^*}_{i-1}\|_2^2 \le \|\boldsymbol{b^*}_i\|_2^2 + g_{i,i-1}^2\|\boldsymbol{b^*}_{i-1}\|_2^2$ $i = 2,\ldots,n\frac{1}{4} < \delta \le 1$

The LLL algorithm uses Gram–Schmidt process for the orthogonalization and the size reduction algorithm (Algorithm 2). Algorithm 3 shows the LLL algorithm.

Algorithm 3 LLL-Algorithm for basis reduction

Input: Lattice basis $\boldsymbol{B} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n})$ and reduction parameter δ with $\frac{1}{4} < \delta < 1$
Output: δ-LLL-reduced basis $\boldsymbol{B}$
1: Using Gram–Schmidt process to compute coefficients $g_{i,j}$
2: $i = 2$
3: **while** $i \le n$
4: Using Algorithm 2 to size reduce the vector $\boldsymbol{b_i}$
5: **If** $\delta\|\boldsymbol{b^*}_{i-1}\|_2^2 > \|\boldsymbol{b^*}_i\|_2^2 + g_{i,i-1}^2\|\boldsymbol{b^*}_{i-1}\|_2^2$ **then**
6: Swap basis vectors $\boldsymbol{b_i}$ and $\boldsymbol{b_{i-1}}$
7: Update Gram–Schmidt coefficients $g_{l,j}$ for $l > j$
8: $i = \max(i-1, 2)$
9: **else**
10: $i = i + 1$
11: **end if**
12: **end while**

The decorrelation process is utilized to alter the integer ambiguity resolution problem of Eq. (3) to a new one

$$\check{\boldsymbol{a}}_{\boldsymbol{1}} = \min(\hat{\boldsymbol{a}}_{\boldsymbol{1}} - \boldsymbol{a_1})^{\mathrm{T}}\boldsymbol{P_1}(\hat{\boldsymbol{a}}_{\boldsymbol{1}} - \boldsymbol{a_1}) \tag{15}$$

by the so called $\boldsymbol{G}$ transformation such that

$$\boldsymbol{a_1} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{a} \quad \boldsymbol{P_1} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{G} \tag{16}$$

where $\boldsymbol{G}$ is unimodular, i.e. $|\det(\boldsymbol{G})| = 1$. When the optimal integer estimate of $\boldsymbol{a_1}$ in model (16) is found, the integer ambiguity parameters $\check{\boldsymbol{a}}$ are calculated via $\check{\boldsymbol{a}} = (\boldsymbol{G}^{\mathrm{T}})^{-1}\check{\boldsymbol{a}}_{\boldsymbol{1}}$. Since $\boldsymbol{G}$ is an integer matrix, $\boldsymbol{G}^{-1}$ and as a result $\check{\boldsymbol{a}}$ will remain integer.

Several decorrelation techniques have been proposed in GNSS literature. We can point out the Gaussian decorrelation technique applied in LAMBDA, proposed by Teunissen (1995), inverse integer Cholesky decomposition proposed by Xu (2001), LLL proposed by Hassibi and Boyed (1998) and Grafarend (2000), modified reduction algorithm proposed by Chang et al. (2005), united ambiguity decorrelation proposed by Liu et al. (1999) and (inverse) paired Cholesky integer transformation proposed by Zhou (2010).

3.5 Why lattice reduction speeds up searching process?

We now mathematically show that the lattice reduction method reduces the numbers of candidates for searching and consequently the searching process speeds up. Consider again the ILS problem

$$\|\hat{\boldsymbol{a}} - \boldsymbol{a}\|_{\boldsymbol{P}}^2 = (\hat{\boldsymbol{a}} - \boldsymbol{a})^{\mathrm{T}}\boldsymbol{P}(\hat{\boldsymbol{a}} - \boldsymbol{a}) = (\boldsymbol{R}\hat{\boldsymbol{a}} - \boldsymbol{R}\boldsymbol{a})^{\mathrm{T}}(\boldsymbol{R}\hat{\boldsymbol{a}} - \boldsymbol{R}\boldsymbol{a}) = \|\boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a})\|^2 \tag{17}$$

where $\boldsymbol{R}$ is the generator matrix of lattice $\Lambda(\boldsymbol{R}) = \{\boldsymbol{R}\boldsymbol{a}: \boldsymbol{a} \in \mathbb{Z}^n\}$. Also, consider that the upper bound of $\|\hat{\boldsymbol{a}} - \boldsymbol{a}\|_{\boldsymbol{P}}^2$ is r^2 i.e. $\|\hat{\boldsymbol{a}} - \boldsymbol{a}\|_{\boldsymbol{P}}^2 \le r^2$. Applying a reduction algorithm such as LLL algorithm to the row vectors of $\boldsymbol{R}^{-1}$ instead of $\boldsymbol{R}$ (the reason will become clear soon), we obtain $\boldsymbol{R_r^{-1}} = \boldsymbol{U}\boldsymbol{R}^{-1}$, or equivalently

$$\boldsymbol{R} = \boldsymbol{R}_r \boldsymbol{U} \tag{18}$$

Then

$$\|\boldsymbol{R}(\hat{\boldsymbol{a}}-\boldsymbol{a})\|^2 = \|\boldsymbol{R}_r \boldsymbol{U}(\hat{\boldsymbol{a}}-\boldsymbol{a})\|^2 = \|\boldsymbol{R}_r(\hat{\boldsymbol{s}}-\boldsymbol{s})\|^2 < r^2 \tag{19}$$

where $\mathbf{s} = \boldsymbol{U}\boldsymbol{a}$ and $\hat{\mathbf{s}} = \boldsymbol{U}\hat{\boldsymbol{a}}$. Since $(\hat{\boldsymbol{s}} - \boldsymbol{s}) = \boldsymbol{R}_r^{-1}\boldsymbol{R}_r(\hat{\mathbf{s}} - \mathbf{s})$ we have

$$(\hat{s}_i - s_i) = \boldsymbol{r}_i^{-1}\boldsymbol{R}_r(\hat{\mathbf{s}} - \mathbf{s}) \tag{20}$$

Similarly

$$(\hat{a}_i - a_i) = \bar{\boldsymbol{r}}_i^{-1}\boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a}) \tag{21}$$

where s_i is the ith element of $\mathbf{s}$, a_i is the ith element of $\boldsymbol{a}$, $\boldsymbol{r}_i^{-1}$ denotes the ith row vector of $\boldsymbol{R}_r^{-1}$ and $\bar{\boldsymbol{r}}_i^{-1}$ denotes the ith row vector of $\boldsymbol{R}^{-1}$. Considering $\boldsymbol{r}_r = \boldsymbol{R}_r(\hat{\boldsymbol{s}} - \boldsymbol{s})$ and $\bar{\boldsymbol{r}}_r = \boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a})$, where $\boldsymbol{r}_r$ and $\bar{\boldsymbol{r}}_r$ are column vectors, and $\langle . \rangle$ denoting the inner product, one has

$$(\hat{s}_i - s_i)^2 = \left(\boldsymbol{r}_i^{-1}\boldsymbol{R}_r(\hat{\mathbf{s}} - \mathbf{s})\right)^2 = \left(\langle \boldsymbol{r}_i^{-1}.\boldsymbol{r}_r \rangle\right)^2 \tag{22}$$

$$(\hat{a}_i - a_i)^2 = \left(\bar{\boldsymbol{r}}_i^{-1}\boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a})\right)^2 = \left(\langle \bar{\boldsymbol{r}}_i^{-1}.\bar{\boldsymbol{r}}_r \rangle\right)^2 \tag{23}$$

Applying the Cauchy–Shwarz inequality to the equations yields

$$\begin{aligned}(\hat{s}_i - s_i)^2 &= \left(\left\langle \boldsymbol{r}_i^{-1}.\boldsymbol{r}_r \right\rangle\right)^2 \le \left\|\boldsymbol{r}_i^{-1}\right\|^2 \|\boldsymbol{r}_r\|^2 \\ &= \left\|\boldsymbol{r}_i^{-1}\right\|^2 \left\|\boldsymbol{R}_r(\hat{\boldsymbol{s}} - \boldsymbol{s})\right\|^2 \end{aligned} \tag{24}$$

$$\begin{aligned}(\hat{a}_i - a_i)^2 &= \left(\left\langle \bar{\boldsymbol{r}}_i^{-1}.\bar{\boldsymbol{r}}_r \right\rangle\right)^2 \le \left\|\bar{\boldsymbol{r}}_i^{-1}\right\|^2 \|\bar{\boldsymbol{r}}_r\|^2 \\ &= \left\|\bar{\boldsymbol{r}}_i^{-1}\right\|^2 \left\|\boldsymbol{R}(\hat{\boldsymbol{a}} - \boldsymbol{a})\right\|^2 \end{aligned} \tag{25}$$

Equations (24) and (25) with Eq. (19) will convert to

$$(\hat{s}_i - s_i)^2 \le \left\|\boldsymbol{r}_i^{-1}\right\|^2 r^2 \tag{26}$$

$$(\hat{a}_i - a_i)^2 \le \left\|\bar{\boldsymbol{r}}_i^{-1}\right\|^2 r^2 \tag{27}$$

Because $\boldsymbol{U}$ is a unimodular matrix we have $\det(\boldsymbol{U}) = \mathbf{1}$ and hence $\det(\boldsymbol{R}_r^{-1}) = \det(\boldsymbol{R}^{-1})$. According to Hadamard's inequality we have

$$|\det(\boldsymbol{R}^{-1})| \le \prod_{i=1}^{n} \left\|\bar{\boldsymbol{r}}_i^{-1}\right\| \tag{28}$$

For orthogonal vectors $\boldsymbol{r}_i^{-1}$, equality is achieved and we have (Mow 2003)

$$|\det(\boldsymbol{R}_r^{-1})| = \prod_{i=1}^{n} \left\|\boldsymbol{r}_i^{-1}\right\| \tag{29}$$

Equations (28) and (29) yield

$$\prod_{i=1}^{n} \|\boldsymbol{r}_i^{-1}\| \le \prod_{i=1}^{n} \left\|\bar{\boldsymbol{r}}_i^{-1}\right\| \tag{30}$$

Using Eqs. (26) and (27) we have

$$\prod_{i=1}^{n} |(\hat{s}_i - s_i)| \le \prod_{i=1}^{n} \left\|\boldsymbol{r}_i^{-1} r\right\| \tag{31}$$

$$\prod_{i=1}^{n} |(\hat{a}_i - a_i)| \le \prod_{i=1}^{n} \left\|\bar{\boldsymbol{r}}_i^{-1} r\right\| \tag{32}$$

Using Eq. (30), the product $\left\|\boldsymbol{r}_1^{-1}\right\| \left\|\boldsymbol{r}_2^{-1}\right\| \ldots \left\|\boldsymbol{r}_n^{-1}\right\|$ is smaller than $\left\|\bar{\boldsymbol{r}}_1^{-1}\right\| \left\|\bar{\boldsymbol{r}}_2^{-1}\right\| \ldots \left\|\bar{\boldsymbol{r}}_n^{-1}\right\|$. Equations (31) and (32) indicate that the numbers of candidates as a whole for searching in the reduced lattice will decrease and consequently the searching process speeds up.

4 Closest point search algorithms

In this section, we start with a conceptual description of various lattice search algorithms. In this framework, we introduce the Pohst strategy, the Schnorr–Euchner refinement of the Pohst strategy and three CLP search algorithms, i.e. Agrell, Eriksson, Vardy, Zeger (AEVZ), modification of Viterbo-Boutros (M-VB) and modification of Schnorr–Euchner (M-SE) that are basically applications of the studies by Fincke and Pohst (1985)) and Schnorr and Euchner (1994).

Pohst (1981) proposed an efficient algorithm for enumerating all lattice points within a sphere with a certain radius. The search strategy used in the LAMBDA method, proposed by Teunissen (1993), is a variant of Pohst enumeration strategy. Pohst enumeration approach has been extensively used in CLP search problems because of its efficiency.

Pohst closest point search algorithm is briefly outlined as follows.

Consider the CLP problem in lattice $\Lambda(\boldsymbol{R})$ defined in model (12). Let R_0 be the squared radius of an n-dimensional sphere centered at $\boldsymbol{y}$. Equation (12) gives

$$\|\boldsymbol{y} - \boldsymbol{R}\mathbf{a}\|^2 \le R_0 \tag{33}$$

Due to the upper triangular form of $\boldsymbol{R}$, the inequality implies a set of conditions as

$$\sum_{j=i}^{n} \left(y_j - \sum_{l=j}^{m} R_{j,l} a_l \right)^2 \le R_0 \quad i = 1, \ldots, n \tag{34}$$

Considering the above conditions in the order from $i = n$ down to 1, the set of admissible values of each variable a_i is achieved using the values given for variables $a_{i+1}, \ldots, a_n$. More explicitly if the values of a_j for $i + 1 \le j \le n$ are fixed, the component $a_i, i = n - 1, n - 2, \ldots, 1$ can take values in the range of integers $[L_i, U_i]$ where

$$L_i = \text{round}\left[\frac{1}{R_{i,i}}\left(y_i - \sum_{j=i+1}^{n} R_{i,j}a_j - \sqrt{R_0 - \sum_{j=i+1}^{n}\left(y_j - \sum_{l=j}^{n} R_{j,l}a_l\right)^2}\right)\right] \tag{35}$$

$$U_i = \text{round}\left[\frac{1}{R_{i,i}}\left(y_i - \sum_{j=i+1}^{n} R_{i,j}a_j + \sqrt{R_0 - \sum_{j=i+1}^{n}\left(y_j - \sum_{l=j}^{n} R_{j,l}a_l\right)^2}\right)\right] \tag{36}$$

where round(.) denotes rounding to the closest integer. If

$$\sum_{j=i+1}^{n}\left(y_j - \sum_{l=j}^{n} R_{j,l}a_l\right)^2 > R_0 \tag{37}$$

there is no value of a_i satisfying the inequalities (35) and (36). Therefore, the points corresponding to the fixed values $a_{i+1}, \ldots, a_n$ do not belong to the sphere with radius $\sqrt{R_0}$ centered at $\boldsymbol{y}$. Pohst algorithm consists of spanning at each level i the admissible interval $[L_i, U_i]$, starting from level $i = n$ and climbing up to level $i = n-1, n-2, \ldots, 1$. At each level, the interval $[L_i, U_i]$ is determined by the values of the variables at lower levels corresponding to higher indices. If the interval $[L_1, U_1]$ is non-empty, all $a \in [L_1, U_1]$ and fixed values of the variables at lower levels yield lattice points in defined sphere with radius R_0. The squared Euclidean distances between such points and $\boldsymbol{y}$ are

$$d^2(\boldsymbol{y}, \boldsymbol{Ra}) = \sum_{j=1}^{n}\left(y_j - \sum_{l=j}^{n} R_{j,l}a_l\right)^2 \tag{38}$$

The Pohst algorithm provides the point $\check{\boldsymbol{a}}$ for which Euclidean distances defined in Eq. (38) is minimum. If, after spanning the interval $[L_n, U_n]$, corresponding to a_n, no point in the sphere is found (empty sphere), the search fails. In this case, the search squared radius R_0 must be increased and the search is resumed with new squared radius. In Pohst method every variable a_i takes values in the order $L_i, L_i + 1, \ldots, U_i$ at each level. Schnorr–Euchner strategy is the variant of Pohst strategy and the intervals at every level are spanned in a zig-zag order, starting from the midpoint of the interval. The midpoint of every interval at level i is as follows

$$M_i = \text{round}\left(\frac{1}{R_{i,i}}\left(y_i - \sum_{j=i+1}^{n} R_{i,j}a_j\right)\right) \tag{39}$$

In Schnorr–Euchner enumeration, every variable at level i takes values in the order $M_i, M_i+1, M_i-1, M_i+2, M_i-2, \ldots$, if

$$y_i - \sum_{j=i+1}^{n} R_{i,j}a_j - R_{i,i}M_i \geq 0 \tag{40}$$

or, in the order $M_i, M_i-1, M_i+1, M_i-2, M_i+2, \ldots$, if

$$y_i - \sum_{j=i+1}^{n} R_{i,j}a_j - R_{i,i}M_i < 0 \tag{41}$$

This enumeration strategy was firstly introduced by Schnorr and Euchner (1994). Teunissen (1995) explained that instead of scanning the interval per ambiguity from left to right for integers (Pohst strategy), one can search in an alternating way around the conditional estimate in a zig-zag order. That strategy is, however, different from Schnorr–Euchner strategy and selecting integer in the interval (per ambiguity) is not based on the conditions (40) and (41). In Schnorr–Euchner strategy, R_0 can be set to infinity ($R_0 = \infty$) and there is no need to compute the search radius at first. In this way the search never fails and the first point found corresponds to the Babai point (Babai 1986; Agrell et al. 2002)

$$a_i^{\text{Babai}} = \text{round}\left(\frac{1}{R_{i,i}}\left(y_i - \sum_{j=i+1}^{n} R_{i,j}a_j^{\text{Babai}}\right)\right) \tag{42}$$

An efficient closest point search algorithm, based on the Schnorr–Euchner variant of the Pohst technique, is implemented by Agrell et al. (2002). We call this CLP search algorithm AEVZ, as an abbreviation for the names of the authors. This strategy is shown to be considerably faster than other known methods, by means of a theoretical comparison with the Kannan algorithm (Kannan 1983) and an experimental comparison with the Pohst algorithm and its variants, such as the Viterbo-Boutros decoder (Viterbo and Boutros 1999). The algorithm can be modified to solve a number of related search problems for lattices, which includes finding the shortest vector, determining the kissing number, computing the Voronoi-relevant vectors, and finding a Korkine–Zolotareff reduced basis (Agrell et al. 2002).

AEVZ is the variant of Pohst strategy and the intervals at every level are spanned also based on Schnorr–Euchner enumeration. The squared search radius is set to infinity. Based on Schnorr–Euchner strategy, it is easy to see that the first point found with $R_0 = \infty$ corresponds to the Babai point. Therefore, the first lattice point generated will be the Babai point when the search process is starting from level $i = n$ and reaching to level 1. After the Babai point is found, the R_0 is set to the distance $d^2(\boldsymbol{y}, \boldsymbol{Ra}^{\boldsymbol{Babai}})$. In this manner the search never fails. During the search process, the search sphere shrinks each time when a new integer point is found. This is crucial to the efficiency of the search process. This method starts to

search a new candidate with returning to level 2 and take the next integer at this level based on Schnorr–Euchner enumeration (Eqs. 40 and 41). If the new calculated squared search radius is smaller than previous radius, the search process moves to level 1, otherwise it proceeds with level 3 to take the next integer value at this level. The search process will be continued until a new candidate at level 1 will be found. Finally, when the search process fails to find a new integer value at level n, i.e. its squared distance is larger than best squared distance found so far, the search process stops and the latest integer point found is the optimal solution we look for.

Modification of the Schnorr–Euchner enumeration (M-SE) is similar to AEVZ CLP search algorithm. The squared radius search is, however, not considered infinity but an input parameter (Damen et al. 2003). A too small squared radius may result in an empty sphere, whereas a too large one may result in too many points to be enumerated. A usual candidate for R_0 is the covering radius, defined to be the radius of the spheres centered at the lattice points that cover the entire space in the most efficient way. The covering radius can be computed by exhaustive search and the running time for exhaustive search becomes forbiddingly large. This problem is called NP-hard (Guruswami et al. 2005). To calculate the initial squared radius R_0, the reader is referred to Teunissen et al. (1997), Hassibi and Boyed (1998), Zhou and Giannakis (2005), and Hassibi and Vikalo (2005). The AEVZ and M-SE methods are implemented in MATLAB and their codes are provided in the supplementary electronic file.

The VB implementation (Viterbo and Boutros 1999) is a variant of Pohst strategy and contrary to the AEVZ CLP search algorithm, the squared search radius is set to R_0. But R_0 is changed adaptively along the search. In this method search strategy continue until it reaches level 1 and obtain the first integer point. Then R_0 is updated and the search is restarted in the new sphere with smaller radius. The new process starts at level 1 to search through all other valid integers from the smallest to the largest. Then move up to level 2 to update the integer value to be the next nearest integer based on Pohst enumeration. If it belongs to the authorized interval at level 2, it moves down to level 1 to update the integer value at this level; otherwise it moves up to level 3 to update the integer value at this level and so on. Finally, when it fails to find a new integer value at level n that satisfies the inequality $(y_n - R_{n,n}a_n)^2 < R_0$, the search process stops. The latest integer point found is the optimal solution we look for. During the search process, the search sphere shrinks each time when a new integer point is found.

A drawback of this technique is that the VB algorithm may re-span values of a_i for some levels i, $1 \leq i \leq n$, that have already been spanned in the previous sphere. In modified-VB (M-VB) algorithm, once a lattice point is found, all the upper bounds of the intervals are updated without restarting. This is the main advantage of this strategy over VB method. In other words, some values of a_i that have already been examined, will not be reconsidered after reducing the sphere radius. For further information, the reader is referred to Damen et al. (2003). The M-VB method is implemented in MATLAB and the code is provided in the supplementary electronic file.

We point out that the AEVZ, M-VB and M-SE CLP search algorithms are the fastest algorithms currently available for finding CLP, which can accordingly be used for ILS estimation.

5 Numerical results and discussion

To compare different search strategies for GNSS high-dimensional ambiguity resolution, in this section, we use many different simulated data to compare the performance of three integer search methods presented in the previous section. This includes simulations such as those implemented in Chang et al. (2005) applied to compare the LAMBDA and MLAMBDA, and simulations using algorithms presented by Xu (2001). Further, we have also included results on a real GPS data set in which real ambiguity vector along with its covariance matrix were used. We test out the searching speed of the methods and compare them to the LAMBDA and MLAMBDA. All presented results in this section are performed in MATLAB 7.6.0 on a PC, 2.8 GHz with 2.96 GB memory running Windows XP professional.

We highlight that the goal is not to test the performance of the LAMBDA and MLAMBDA. We hypothesize that the original LAMBDA implemented in MATLAB by Delft University of Technology is not likely optimized to allow for performance comparison in terms of computational efficiency because this available version is intended for educational purposes. We only compare searching speed of the methods presented among each other and compare them to the current version of LAMBDA available in the website of Delft University of Technology and the MLAMBDA provided by Xiao-Wen Chang.

Simulations are performed for different cases. The real vector $\hat{\boldsymbol{a}}$ was constructed as

$$\hat{\boldsymbol{a}} = 100 \times randn(n, 1) \tag{43}$$

where $randn(n, 1)$ is a MATLAB built-in function to produce a vector of n random entries having standard normal distribution.

Similar to the simulations in Chang et al. (2005), to construct covariance matrix of real ambiguity parameters, we consider seven cases. The first four cases are based on $\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{D}\boldsymbol{L}$, where $\boldsymbol{L}$ is a unit lower triangular matrix with each l_{ij} (for $i > j$) being a random number generated by $randn$, and $\boldsymbol{D}$ is generated in four different cases:

- Case 1: $\boldsymbol{D} = \text{diag}(d_i)$, $d_i = rand$, where $rand$ is a MATLAB built-in function to generate uniformly distributed random numbers in (0, 1).
- Case 2: $\boldsymbol{D} = \text{diag}(n^{-1}, (n-1)^{-1}, \ldots, 1)$
- Case 3: $\boldsymbol{D} = \text{diag}(1, (2)^{-1}, \ldots, n^{-1})$
- Case 4: $\boldsymbol{D} = \text{diag}(200, 200, 200, 0.1, 0.1, \ldots, 0.1)$

The other three cases are as follows:

- Case 5: $\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{T}}$, $\boldsymbol{U}$ is a random orthogonal matrix obtained by the QR factorization of a random matrix generated by $randn(n, n)$, $\boldsymbol{D} = \text{diag}(d_i t)$, $d_i = rand$.
- Case 6: $\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{T}}$, $\boldsymbol{U}$ is generated in the same way as in case 5, $d_1 = 2^{-\frac{n}{4}}$, $d_n = 2^{\frac{n}{4}}$, other diagonal elements of $\boldsymbol{D}$ is randomly distributed between d_1, d_n, n is the dimension of $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$. Thus the condition number of $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$ is $2^{\frac{n}{2}}$.
- Case 7: $\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}, \boldsymbol{A} = randn(n, n)$.

Case 4 is motivated because the covariance matrix $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$ in GPS usually has a large gap between the third conditioned standard deviation and the forth one (Teunissen 1998a, Sect. 8.3.3).

To fairly compare all search processes and to speed up searching in all simulations, after applying the decorrelation process, we used the search algorithms to the transformed ILS problem. All presented results are performed over 100 independent runs. Numerical results just show the average search time of the three presented algorithms (AEVZ, M-VB and M-SE), LAMBDA and MLAMDA. The computation time for the decorrelation was not included. For the LAMBDA, MLAMBDA and the three lattice search algorithms, the Gaussian decorrelation method is used before the search process.

Because we used the same decorrelation method for all search algorithms, the total ILS estimation time will increase by the same amount if we include the computation time for the decorrelation. This indicates that only the computation time of different search algorithms is of interest in the present contribution. Obviously, if we do not apply the decorrelation process and use the search process on the original least-squares problem, the search time for the presented algorithms, LAMBDA and MLAMBDA will all increase proportional to the computation search time presented for different methods when applying the decorrelation process. In each run the best optimal ILS is estimated and we search only for the (first) best solution with LAMBDA and MLAMBDA. The average integer searching time (excluding the reduction time) for all simulated data and for dimensions of 40 and 45 are given below.

For weak models in which the success rate is low, the decorrelation and search time will generally be longer. To have strong models, in all simulations, the success rate is considered to be 99.999%. To reach exactly this high success rate, the simulated covariance matrices in some cases will slightly be scaled as $Q_{\hat{\boldsymbol{a}}} = \sigma Q_{\hat{\boldsymbol{a}}}$. A simple measure to infer the float ambiguity precision is the ambiguity dilution of precision (ADOP) defined in Teunissen (1997). We applied the success rate defined in Teunissen (1998b), which is a function of ADOP. Average searching time in seconds for all simulation cases are presented in Table 1.

We also constructed the real vector $\hat{\boldsymbol{a}}$ using the MATLAB built-in function in the statistics toolbox ($mvnrnd.m$), which allows to generate the ambiguities for a given covariance matrix. Tables 2 and 3 outline the results of all simulation cases 1–7 for dimensions $n = 40$ and $n = 45$, respectively.

When compared with the LAMBDA and MLAMBDA methods, the methods presented in this contribution are faster. Table 1 shows, on average, that AEVZ is about 127, 470, 399, 261, 106, 269, 256, M-VB is about 24, 121, 35, 58, 22, 77, 57 and M-SE is about 56, 315, 38, 103, 58, 177, 187 times as fast as LAMBDA for cases 1–7, respectively. In Table 2, AEVZ is about 105, 443, 197, 396, 153, 224, 446, M-VB is about 20, 84, 7, 83, 47, 39, 99 and M-SE is about 44, 374, 7, 147, 101, 132, 323 times faster than LAMBDA,

Table 1 Average searching time over 100 independent runs (seconds) of different algorithms for dimension $n = 40$

Case	Method				
	LAMBDA	MLAMDA	AEVZ	M-VB	M-SE
1	1.8067	0.0974	0.0142	0.0737	0.0322
2	1,266.0964	16.5427	2.6934	10.4765	4.0202
3	0.5983	0.0089	0.0015	0.0169	0.0158
4	2.6668	0.0668	0.0102	0.0455	0.0257
5	4.7601	0.2790	0.0448	0.2091	0.0817
6	40.1202	1.0440	0.1489	0.5204	0.2264
7	216.5075	5.5765	0.8446	3.8094	1.1548

Real ambiguities are simulated using Eq. (43)

Table 2 Average searching time over 100 independent runs (seconds) of different algorithms for dimension $n = 40$

Case	Method				
	LAMBDA	MLAMDA	AEVZ	M-VB	M-SE
1	1.5421	0.1077	0.0147	0.0769	0.0351
2	952.4090	11.4410	2.1501	11.2924	2.5474
3	0.1021	0.0022	0.000519	0.0136	0.0133
4	3.3670	0.0458	0.0085	0.0403	0.0228
5	7.9429	0.2739	0.0520	0.1700	0.0784
6	32.4202	0.9899	0.1444	0.8309	0.2449
7	368.1308	5.1078	0.8255	3.7123	1.1392

Real ambiguities are simulated using the MATLAB built-in function $mvnrnd.m$

Table 3 Average searching time over 100 independent runs (seconds) of different algorithms for dimension $n = 45$

Case	Method				
	LAMBDA	MLAMDA	AEVZ	M-VB	M-SE
1	4.7110	0.3246	0.0508	0.2482	0.0892
2	6,279.4891	83.428	13.2846	81.3527	27.2622
3	0.4278	0.0011	0.0006389	0.0187	0.0186
4	9.3553	0.1196	0.0231	0.0864	0.0474
5	21.6018	1.0358	0.1859	0.5323	0.2590
6	269.1465	4.5191	0.8125	3.3407	0.9809
7	2,117.5926	30.5525	6.1057	24.7601	7.6046

Real ambiguities are simulated using the MATLAB built-in function *mvnrnd.m*

respectively. In Table 3, AEVZ is about 92, 472, 669, 404, 116, 331, 347, M-VB is about 19, 77, 23, 108, 40, 80, 85 and M-SE is about 53, 230, 23, 197, 83, 274, 278 times as fast as LAMBDA, respectively. AEVZ is about 7, 6, 6, 7, 6, 7, and 7 times in Table 1, 7, 5, 4, 5, 5, 7, and 6 times in Table 2 and 6, 6, 2, 5, 6, 6, and 5 times in Table 3 as fast as MLAMBDA for cases 1–7, respectively.

When AEVZ is compared to the M-VB and M-SE, it appears to be the fastest method. The results of the average searching time (in seconds) are provided for the AEVZ method in Fig. 3, for $n = 10, 11, \ldots, 50$. Also Fig. 4 shows the average number of candidates searched in each case.

Another simulation algorithm was proposed by Xu (2001). Let $\boldsymbol{Q}_{\hat{a}}$ be decomposed as follows:

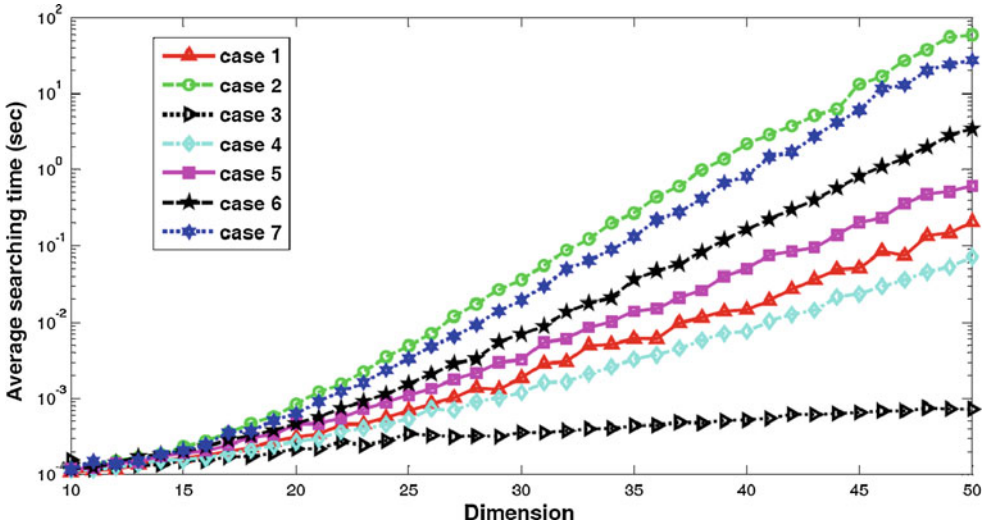

Fig. 3 Average searching time (seconds) in logarithmic scale for the AEVZ method over 100 independent runs for cases 1–7

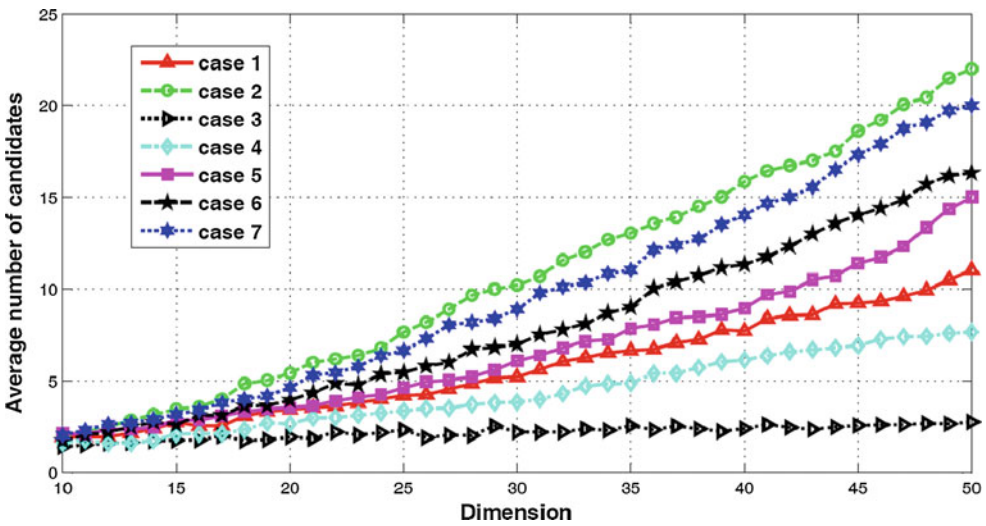

Fig. 4 Average number of candidates for the AEVZ method searched over 100 independent runs for cases 1–7

$$\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\boldsymbol{T}} \tag{44}$$

where $\boldsymbol{U}$ is the normalized orthogonal eigenvector square $n \times n$ matrix and $\boldsymbol{\Lambda}$ is the diagonal matrix with positive eigenvalue elements of $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$. The normalized orthogonal eigenvector matrix $\boldsymbol{U}$ can be uniquely represented as follows:

$$\boldsymbol{U} = \boldsymbol{U}_{n\times(n-1)} \ldots \boldsymbol{U}_{3\times 2}\boldsymbol{U}_{n\times 1} \ldots \boldsymbol{U}_{3\times 1}\boldsymbol{U}_{2\times 1} \tag{45}$$

where

$$\boldsymbol{U}_{ij} = \begin{bmatrix} \boldsymbol{I}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \cos\theta_{ij} & \boldsymbol{0} & \sin\theta_{ij} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\sin\theta_{ij} & \boldsymbol{0} & \cos\theta_{ij} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix} \tag{46}$$

with $\boldsymbol{I}_1, \boldsymbol{I}_2$ and $\boldsymbol{I}_3$ identity matrices of suitable orders, $-\frac{\pi}{2} \leq \theta_{ij} \leq \frac{\pi}{2}$ and $\boldsymbol{0}$ is either a zero matrix or a zero (row or column) vector. For more theoretical details the reader is referred to Xu (1999, 2002) and Xu and Grafarend (1996). $\boldsymbol{\Lambda}$ was constructed similar to those of previous section as

- Case 1: $\boldsymbol{\Lambda} = \text{diag}(\lambda_i), \lambda_i = rand$
- Case 2: $\boldsymbol{\Lambda} = \text{diag}(n^{-1}, (n-1)^{-1}, \ldots, 1)$
- Case 3: Case 3: $\boldsymbol{\Lambda} = \text{diag}(1, (2)^{-1}, \ldots, n^{-1})$
- Case 4: $\boldsymbol{\Lambda} = \text{diag}(200, 200, 200, 0.1, 0.1, \ldots, 0.1)$
- Case 5: $\lambda_1 = 2^{-\frac{n}{4}}, \lambda_n = 2^{\frac{n}{4}}$, other diagonal elements of $\boldsymbol{\Lambda}$ is randomly distributed between λ_1, λ_n, n is the dimension of $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$. Thus the condition number of $\boldsymbol{Q}_{\hat{\boldsymbol{a}}}$ is $2^{\frac{n}{2}}$.

The average searching time for dimension 40 of the proposed methods, LAMBDA, and MLAMBDA are shown in Tables 4 and 5. In both tables the covariance matrix $\boldsymbol{Q}_{a\cdot}$ is simulated for cases 1–5. In Table 4, the real vector $\hat{\boldsymbol{a}}$ is simulated as $\hat{\boldsymbol{a}} = 100 \times randn(n, 1)$, while, in Table 5 $\hat{\boldsymbol{a}}$ is simulated using the MATLAB built-in function $mvnrnd.m$.

Table 4 shows that, on average, AEVZ is about 97, 53, 38, 29, 2,104, M-VB is about 31, 8, 6, 9, 280 and M-SE is about 55, 11, 8, 14,369 times as fast as LAMBDA for cases 1–5, respectively. AEVZ is about five times faster than the MLAMBDA. For cases 1–5 (Table 5), AEVZ is about 65, 41, 33, 29, 1,467, M-VB is about 19, 21, 2, 7, 151 and M-SE is about 56, 23, 2, 10, 224 times faster than LAMBDA, respectively. And AEVZ is about five times faster than the MLAMBDA.

Table 4 Average searching time (seconds) over 100 independent runs of different algorithms for dimension $n = 40$

Case	Method				
	LAMBDA	MLAMBDA	AEVZ	M-VB	M-SE
1	1.8622	0.1275	0.0192	0.0606	0.0338
2	0.1395	0.01485	0.0026	0.0162	0.0129
3	0.1299	0.0156	0.0034	0.0215	0.0166
4	0.3689	0.0476	0.0123	0.0418	0.0259
5	4.8389	0.0124	0.0023	0.0173	0.0131

Real ambiguities and its covariance matrix are simulated using Eqs. (43) and (44), respectively

Table 5 Average searching time (seconds) over 100 independent runs of different algorithms for dimension $n = 40$

Case	Method				
	LAMBDA	MLAMBDA	AEVZ	M-VB	M-SE
1	1.2326	0.0927	0.0188	0.0630	0.0221
2	0.0237	0.0014	0.0005823	0.0011	0.0010
3	0.0176	0.0013	0.0005348	0.0097	0.0093
4	0.1446	0.0236	0.0049	0.0205	0.0150
5	2.9337	0.0199	0.0020	0.0194	0.0131

Real ambiguities are simulated using the MATLAB built-in function $mvnrnd.m$ and its covariance matrix using Eq. (44)

Table 6 Average searching time (seconds) over 100 independent runs of different algorithms ($n = 40$)

z	Method				
	LAMBDA	MLAMBDA	AEVZ	M-VB	M-SE
4	455.3979	2.1859	0.3481	1.9175	0.2894
5	2,178.0591	5.8084	1.2373	4.1437	1.169
6	6,410.9015	16.8465	2.1080	13.6155	3.6330
7	9,400.0812[a]	29.5532	3.8678	27.1670	5.8936
8	10,679.6293[a]	50.8189	5.5730	32.1611	9.3548

Real ambiguities are simulated using the MATLAB built-in function $mvnrnd.m$ and its covariance matrix is constructed by $\boldsymbol{Q}_{\hat{\boldsymbol{a}}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\boldsymbol{T}}$, where Λ is based on Eq. (47)

[a] For 25 runs

In the next simulation cases we assume that ranges for the eigenvalues and the condition numbers of the covariance matrix are very large. $\boldsymbol{\Lambda}$ was constructed as

$$\boldsymbol{\Lambda} = \text{diag}(\lambda_i), \quad \lambda_i = 10^{rand*z} \quad z = 4, 5, 6, 7, 8 \tag{47}$$

to satisfy large condition numbers. Average searching time for real vector $\hat{\boldsymbol{a}}$ simulated using the MATLAB built-in function $mvnrnd.m$ are presented in Table 6.

Because this simulation may not represent the practical situation, these results have not been included for calculating the average speed of the three presented methods compared with LAMBDA and MLAMBDA.

We further evaluate the performance of different search strategies using a real GPS data set collected using dual frequency GPS Trimble R7 receivers on 14 April 2009, and a 15-s interval. The total number of epochs is 838. The sky plot for this experiment is shown in Fig. 5. In each epoch the float ambiguity vector and its covariance matrix are obtained. The average searching time (over entire observation span) of different search strategies for fixing the ambiguities is

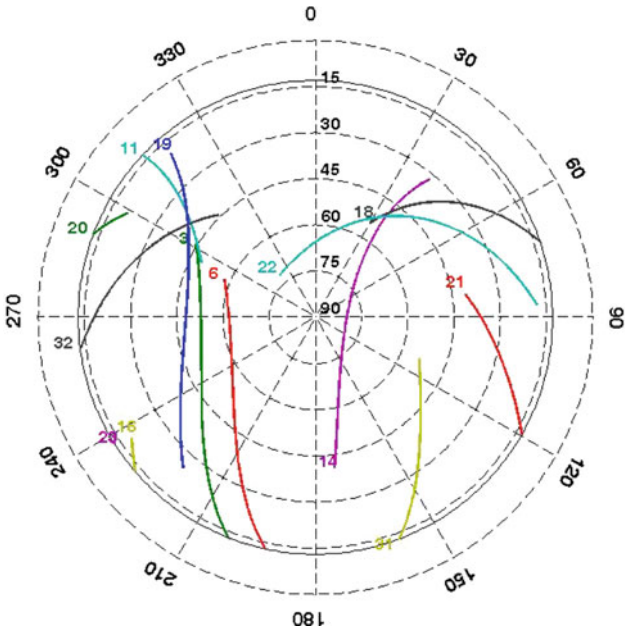

Fig. 5 The sky plot of all observed satellites in view above 13 degree elevation angle for entire time span of 5:25:15 till 8:54:45 on 14 April 2009

Table 7 Average searching time (seconds) of different algorithms

Method				
LAMBDA	MLAMBDA	AEVZ	M-VB	M-SE
12.0195	0.2357	0.0338	0.1844	0.0534

presented in Table 7. In this case we do not apply the decorrelation process and use the search process on the original least-squares problem. Similar results to those presented for the simulated cases are obtained. Among them, it is clear that the AEVZ is the fastest method to the solution of the ILS estimation problem.

The proposed methods speed up searching CLP in lattices and give the optimal integer solution. The results in general show that these methods are faster than the LAMBDA and MLAMBDA methods.

6 Summary and conclusions

There exist several methods for ILS estimation. In cases where tens of integer ambiguities are involved, the existing methods are still slow. In this contribution, we investigated the ILS estimation problem, which was shown to be the same as the CLP problem in lattice theory. The mathematical formulations of three efficient CLP search algorithms, i.e. AEVZ, M-SE and M-VB were presented.

AEVZ and M-SE algorithms are inspired by Schnorr–Euchner enumeration strategy and M-VB is inspired by Pohst enumeration strategy. M-VB is more efficient than the VB searching algorithm. M-SE is similar to AEVZ search algorithm but the squared radius search is not infinity but an input parameter. In AEVZ search algorithm, search radius is set to infinity and therefore the search never fails. These algorithms can all be utilized to solve any IQP problem, including the least-squares integer estimation of ambiguity parameters.

We discussed the mathematical background and presented their implementations. We then tested the performance of the algorithms using different simulated and real GPS data. AEVZ, M-SE and M-VB methods when compared to the available version of LAMBDA and MLAMBDA were proved to be faster on all simulated and real data. The numerical examples show that, on average, AEVZ is about 320 times, M-VB is about 50 times and M-SE is about 120 times faster than LAMBDA for dimension 40. These numbers change to about 350, 60 and 160 times for dimension 45, which shows its efficiency at higher dimensions. The AEVZ was shown to be about 5 times faster than MLAMBDA.

Research into the performance of the presented algorithms for a constrained ILS estimation is ongoing and will be the subject of future publications.

Acknowledgments The authors gratefully acknowledge Xiao-Wen Chang for providing us with the MILES package. Thanks also go to Peiliang Xu for introducing a few references and to Peter Teunissen for his valuable comments on an earlier version of this paper. The authors would also like to acknowledge the valuable comments of the editor, Pascal Willis, the editor-in-chief, Roland Klees, and three anonymous reviewers, which significantly improved the presentation and quality of this paper.

Appendix

Proposition 1 (QR decomposition) *Let $\boldsymbol{A} \in \mathbb{R}^{d\times k}$ and be non-singular, then the QR decomposition $\boldsymbol{A} = \boldsymbol{QR}$ is unique, where $\boldsymbol{Q}$ is unitary and $\boldsymbol{R}$ is an upper triangular matrix with positive diagonal elements Golub and Loan (1996).*

$$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k) = \boldsymbol{QR} = (\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k) \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k} \\ 0 & r_{2,2} & \cdots & r_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{k,k} \end{bmatrix} \quad (1)$$

Proposition 2 (Gram–Schmidt orthogonalization) *The orthogonal basis $\boldsymbol{b}_i^{\boldsymbol{0}}$ of the basis vectors $\boldsymbol{b}_i$ determined by the following iterated process is called the Gram–Schmidt process:*

$$\boldsymbol{b}_1^{\boldsymbol{0}} = \boldsymbol{b}_1 \quad (2)$$

$$\boldsymbol{b}_i^{\boldsymbol{0}} = \boldsymbol{b}_i - \sum_{j=1}^{i-1} g_{i,j} \boldsymbol{b}_j^{\boldsymbol{0}} \quad (3)$$

where

$$g_{i,j} = \begin{cases} \frac{\langle b_i, b_j^0 \rangle}{\langle b_j^0, b_j^0 \rangle} & i > j \\ 1 & i = j \\ 0 & else \end{cases} \quad (4)$$

The Gram–Schmidt process gives the QR decomposition of a basis $\boldsymbol{B} = (\boldsymbol{b_1 b_2} \ldots \boldsymbol{b_k})$ such that

$$\boldsymbol{Q} = (\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots, \boldsymbol{q_n}) = \left(\frac{\boldsymbol{b_1^0}}{\|\boldsymbol{b_1^0}\|^2}, \frac{\boldsymbol{b_2^0}}{\|\boldsymbol{b_2^0}\|^2}, \ldots, \frac{\boldsymbol{b_k^0}}{\|\boldsymbol{b_k^0}\|^2} \right) \quad (5)$$

and

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k} \\ 0 & r_{2,2} & \cdots & r_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{k,k} \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{b_1^0}\|^2 & 0 & \cdots & 0 \\ 0 & \|\boldsymbol{b_2^0}\|^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \|\boldsymbol{b_k^0}\|^2 \end{bmatrix} \times \boldsymbol{R}^* \quad (6)$$

where

$$\boldsymbol{R}^* = \begin{bmatrix} 1 & g_{2,1} & \cdots & g_{k-1,1} & g_{k,1} \\ 0 & 1 & g_{3,2} & \cdots & g_{k,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & g_{k,k-1} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \quad (7)$$

The Gram–Schmidt orthogonalization algorithm and the inverse integer Cholesky decorrelation method are implemented in MATLAB and their codes are provided in the supplementary electronic file.

References

Agrell E, Eriksson T, Vardy A, Zeger K (2002) Closest point search in lattices. IEEE Trans Inf Theory 48:2201–2214

Babai L (1986) On Lováasz' lattice reduction and the nearest lattice point problem. Combinatorica 6:1–13

Bartkewitz T (2009) Improved lattice basis reduction algorithms and their efficient implementation on parallel systems. Diploma thesis, Ruhr University, Germany

Buist PJ (2007) The baseline constrained LAMBDA method for single epoch, single frequency attitude determination applications. In: Proceedings of ION GPS, p 12

Chang X, Yang X, Zhou T (2005) MLAMBDA: a modified LAMBDA algorithm for integer least-squares estimation. J Geod 79:552–565

Chen D (1994) Development of a fast ambiguity search filtering (FASF) method for GPS carrier phase ambiguity resolution. UCGE Reports 20071, PhD dissertation

Chen D, Lachapelle G (1995) A comparison of the FASF and least-squares search algorithms for on-the-fly ambiguity resolution, navigation. J Inst Navig 42(2):371–390

Cohen CE (1996) Attitude determination. In: Parkinson BW, Spilker JJ (eds) Global positioning system: theory and applications II. AIAA, Washington, pp 519–538

Cohen H (1995) A course in computational algebraic number theory. Springer, Berlin

Coveyou RR, Macpherson RD (1967) Fourier analysis of uniform random number generations. J ACM 14(1):100–119

Damen MO, El Gamal H, Caire G (2003) On maximum-likelihood detection and the search for the closest lattice point. IEEE Trans Inf Theory 49(10):2389–2402

Fincke U, Pohst M (1985) Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Math Comput 44(170):463–471

Frei E, Beutler G (1990) Rapid static positioning based on the fast ambiguity resolution approach "FARA": theory and first results. Manuscr Geod 15:325–356

Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore

Grafarend EW (2000) Mixed integer-real valued adjustment (IRA) problems. GPS Solut 4:31–45

Grotschel M, Lovász L, Schrijver A (1993) Geometric algorithms and combinatorial optimization. Springer, Berlin

Guruswami V, Micciancio D, Regev O (2005) The complexity of the covering radius problem. Comput Complex 14:90–121

Hanrot G, Stehle D (2007) Improved analysis of Kannan's shortest lattice vector algorithm. In: Advances in cryptology—crypto 2007, proceedings. A. Menezes, vol 4622, pp 170–186

Hassibi A, Boyed S (1998) Integer parameter estimation in linear models with applications to GPS. IEEE Trans Signal Proc 46:2938–2952

Hassibi B, Vikalo H (2005) On the sphere-decoding algorithm I. expected complexity. IEEE Trans Signal Proc 53:2806–2818

Hermite C (1850) Extraits de lettres de M. Hermite à M. Jacobi sur diff/erents objets de la th/eorie des nombres. J Reine Angew Math 40:279–290

Kannan R (1983) Improved algorithms for integer programming and related lattice problems. Paper presented at the conference proceedings of the annual ACM symposium on theory of computing, pp 193–206

Kannan R (1987) Algorithmic geometry of numbers. Annu Rev Comput Sci 16:231–267

Kim D, Langley RB (2000) A search space optimization technique for improving ambiguity resolution and computational efficiency. Earth Planets Space 52(10):807–812

Korkine A, Zolotareff G (1873) Sur les formes quadratiques. Math Annalen 6(3):366–389 (in French)

Lagrange JL (1773) Recherches d'arithmetique. Nouveaux Memoires de l'Academie de Berlin

Langley RB, Beutler G, Delikaraoglou D, Nickerson B, Santerre R, Vanicek P, Well DE (1984) Studies in the application of the GPS to differential positioning. Technical Report No. 108, University of New Brunswick, Canada

Lenstra AK, Lenstra HW, Lovász L (1982) Factoring polynomials with rational coefficients. Math Ann 261:513–534

Liu LT, Hsu HT, Zhu YZ, Ou JK (1999) A new approach to GPS ambiguity decorrelation. J Geod 73:478–490

Micciancio D, Goldwasser S (2002) Complexity of lattice problems: a cryptographic perspective. Kluwer international series in engineering and computer science, vol 671. Kluwer Academic Publishers, Boston

Minkowski H (1905) Diskontinuitätsbereich für arithmetische Äquivalenz. J Rreine Angew Math 129:220–274 (in German)

Mow WH (2003) Universal lattice decoding: principle and recent advances. Wirel Commun Mobile Comput 3(5):553–569

Nguyen PQ, Stehlé D (2004) Low-dimensional lattice basis reduction revisited (extended abstract). In: Proceedings of the 6th interna-

tional algorithmic number theory symposium (ANTS-VI), lecture notes in computer science, vol 3076. Springer, Berlin, pp 338–357
Nguyen PQ, Stehlé D (2009) An LLL algorithm with quadratic complexity. SIAM J Comput 39(3):874–903
Nguyen PQ, Stern J (2001) The two faces of lattices in cryptology. In: Proceedings of CALC '01, lecture notes in computer science, vol 2146. Springer, Berlin, pp 146–180
Pohst M (1981) On the computation of lattice vector of minimal length, successive minima and reduced bases with applications. ACM SIGSAM Bull 15:37–44
Pujol X, Stehle D (2008) Rigorous and efficient short lattice vectors enumeration. In: Advances in cryptology—Asiacrypt 2008 J Pieprzyk, vol 5350, pp 390–405
Schnorr CP, Euchner M (1994) Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math Program 66:181–199
Steinfeld R, Pieprzyk J, Wang H (2007) Lattice-based threshold changeability for standard Shamir secret-sharing schemes. IEEE Trans Inf Theory 53:2542–2559
Teunissen PJG (1993) Least-squares estimation of the integer GPS ambiguities. Invited lecture, section IV theory and methodology, IAG general meeting, Beijing, China. Also in Delft Geodetic Computing Centre LGR series, No. 6, p 16
Teunissen PJG (1994) A new method for fast carrier phase ambiguity estimation. In: Proceedings of the IEEE PLANS'94, Las Vegas, NV, 11–15 April 1994, pp 562–573
Teunissen PJG (1995) The least-squares ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation. J Geod 70:65–82
Teunissen PJG (1996) An analytical study of ambiguity decorrelation using dual frequency code and carrier phase. J Geod 70:515–528
Teunissen PJG (1997) A canonical theory for short GPS baselines, parts I–IV. J Geod 71:320–336, 389–401, 486–501, 513–525
Teunissen PJG (1998a) GPS carrier phase ambiguity fixing concepts. In: Teunissen P, Kleusberg A (eds) GPS for geodesy, 2nd edn. Springer, Berlin, pp 317–388
Teunissen PJG (1998b) Success probability of integer GPS ambiguity rounding and bootstrapping. J Geod 72:606–612
Teunissen PJG, De Jonge PJ, Tiberius CC (1997) The least-squares ambiguity decorrelation adjustment: its performance on short GPS baselines and short observation spans. J Geod 71:589–602
Viterbo E, Boutros J (1999) A universal lattice code decoder for fading channels. IEEE Trans Inf Theory 45(5):1639–1642
Wei Z (1986) Positioning with NAVSTAR, the global positioning system. Report No. 370, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, OH
Xu PL (1998) Mixed integer observation models and integer programming in geodesy. J Geod Soc Jpn 44:169–187
Xu PL (1999) Spectral theory of constrained second-rank symmetric random tensors. Geophys J Int 138(1):1–24
Xu PL (2001) Random simulation and GPS decorrelation. J Geod 75:408–423
Xu PL (2002) Isotropic probabilistic models for directions, planes and referential systems. Proc R Soc Lond Ser A 458(2024):2017–2038
Xu PL (2006) Voronoi cells, probabilistic bounds and hypothesis testing in mixed integer linear models. IEEE Trans Inf Theory 52:3122–3138
Xu PL, Grafarend E (1996) Statistics and geometry of the eigenspectra of three-dimensional second-rank symmetric random tensors. Geophys J Int 127(3):744–756
Xu PL, Cannon E, Lachapelle G (1995) Mixed integer programming for the resolution of GPS carrier phase ambiguities. Paper presented at IUGG95 assembly, Boulder, 2–14 July
Zhao W, Giannakis GB (2005) Sphere decoding algorithms with improved radius search. IEEE Trans Commun 53(7):1104–1109
Zhou Y (2010) A new practical approach to GNSS high-dimensional ambiguity decorrelation. GPS Solut. doi:10.1007/s10291-010-0192-6