On 'decorrelation' in solving integer least-squares problems for ambiguity determination

M. A. Borno[1], X.-W. Chang[2] and X. H. Xie*[2]

This paper intends to shed light on the decorrelation or reduction process in solving integer least squares (ILS) problems for ambiguity determination. We show what this process should try to achieve to make the widely used discrete search process fast and explain why neither decreasing correlation coefficients of real least squares (RLS) estimates of the ambiguities nor decreasing the condition number of the covariance matrix of the RLS estimate of the ambiguity vector should be an objective of the reduction process. The new understanding leads to a new reduction algorithm, which avoids some unnecessary size reductions in the Lenstra-Lenstra-Lovász (LLL) reduction and still has good numerical stability. Numerical experiments show that the new reduction algorithm is faster than LAMBDA's reduction algorithm and MLAMBDA's reduction algorithm (to less extent) and is usually more numerically stable than MLAMBDA's reduction algorithm and LAMBDA's reduction algorithm (to less extent).

Keywords: Integer least squares estimation, LAMBDA, MLAMBDA, Decorrelation, Reduction, Condition number, Search

Introduction

In high precision relative GNSS positioning, a key component is to resolve the unknown double differenced ambiguities of the carrier phase data as integers. The most successful approach, which was proposed by Teunissen, see, e.g., [**17**], [**18**], [**19**], [**21**] and [**22**], is to solve an integer least squares (ILS) problem. The corresponding numerical method proposed by Teunissen is referred to as the LAMBDA (Least-squares AMBiguity Decorrelation Adjustment) method. A detailed description of the algorithm and implementation is given by [**6**]. Its educational software (Fortran version and Matlab version) is available from Delft University of Technology. Frequently asked questions and misunderstanding about the LAMBDA method are addressed by [**9**]. The LAMBDA method can be used to solve some high-dimensional problems arising in dense network processing as indicated in [**11**]. Recently a modified and faster method called MLAMBDA was proposed in [**4**], which was then further modified and extended to handle mixed ILS problem by using orthogonal transformations, resulting in the Matlab package MILES [**5**].

An often used approach to solving an ILS problem in the literature, including the papers mentioned above, is the discrete search approach. Most methods based on the discrete search approach have two stages: reduction and search. In the first stage, the original ILS problem is transformed to a new one by a reparametrisation of the original ambiguities. In this stage, LAMBDA and other methods decorrelate the ambiguities in the GNSS context. For this reason, it is called the 'decorrelation' stage in the GNSS literature. The word 'decorrelation' seems to have caused some confusion in some literature, where it was believed that this stage is to make the correlation coefficients small, see, e.g., [**13**, sec. 5]. In [**4**], the first stage is referred to as 'reduction', because the process is essentially a lattice reduction process. Like [**4**], we prefer 'reduction' to 'decorrelation' in this paper, as the former is less confusing. In the second stage, the optimal ILS estimate or a few optimal or suboptimal ILS estimates of the parameter vector over a region in the ambiguity parameter space are found by enumeration. The purpose of the reduction process is to make the search process more efficient.

The typical search process which is now widely used is the Schnorr and Euchner based depth-first tree search, which enumerates (part of) integer points in a hyperellipsoid to find the solution, see, [**1**], [**4**], [**6**], [**15**] and [**21**]. A comparison of some different reduction and search strategies was attempted in [**8**]. What should the reduction process achieve to make the search process faster? One (partial) answer in the literature is that the reduction process should try to decorrelate the covariance matrix of the real least squares (RLS) estimate as much as possible, i.e., make the off-diagonal entries of the covariance matrix as small as possible; see, e.g., [**4**], [**6**], [**16**, p. 495], [**19**] and [**21**] (note that these publications also mentioned other objectives to achieve). Another answer in the literature is that the reduction process should reduce the condition number of the covariance matrix as much as possible; see, e.g., [**13**], [**14**], [**25**] and [**26**]. In this paper, we will argue that although decorrelation or reducing the condition number may be

[1]Department of Computer Science, University of Toronto, Toronto, Ont., Canada
[2]School of Computer Science, McGill University, Montreal, QC, Canada

*Corresponding author, email xiaohu.xie@mail.mcgill.ca

Received 3 May 2012; accepted 15 September 2012
DOI 10.1179/1752270612Y.0000000029

helpful for the discrete search process, they are not the right answers to the question. We shed light on what the reduction process should try to achieve. The new understanding leads to a more efficient reduction algorithm that will be presented in this paper. More theoretical discussion about how reduction can improve search efficience can be found in [**3**].

The paper is organised as follows. First, we briefly review the typical reduction and search strategies used in solving the ILS problem. Then, we show that using lower triangular integer Gauss transformations alone in the reduction process do not affect the search speed of the search strategy. In particular, we explain why decreasing the correlation coefficients of the RLS estimates of the ambiguities should not be an objective of the reduction process. After that, we argue why decreasing the condition number of the covariance matrix of the RLS estimate of the ambiguity vector should not be as an objective of the reduction process. Then, a new more efficient reduction algorithm is presented. And numerical results are given to compare different reduction algorithms. Finally, we give a brief summary.

We now describe the notation to be used in this paper. The sets of all real and integer $m\times n$ matrices are denoted by $\mathbb{R}^{m\times n}$ and $\mathbb{Z}^{m\times n}$, respectively, and the set of real and integer n-vectors are denoted by $\mathbb{R}^n$ and $\mathbb{Z}^n$, respectively. The identity matrix is denoted by $\mathbf{I}$ and its ith column is denoted by $\mathbf{e}_i$. We use Matlab-like notation to denote a submatrix. Specifically, if $\mathbf{A}=(a_{ij})\in\mathbb{R}^{m\times n}$, then $\mathbf{A}_{i,:}$ denotes the ith row, $\mathbf{A}_{:,j}$ the jth column, and $\mathbf{A}_{i_1:i_2,j_1:j_2}$ the submatrix formed by rows i_1 to i_2 and columns j_1 to j_2. For $\mathbf{z}\in\mathbb{R}^n$, we use $[\mathbf{z}]$ to denote its nearest integer vector, i.e., each entry of $\mathbf{z}$ is rounded to its nearest integer (if there is a tie, the one with smaller magnitude is chosen).

Reduction and search

Suppose $\hat{\mathbf{x}}\in\mathbb{R}^n$ is the RLS estimate of the integer parameter vector (i.e., the double differenced integer ambiguity vector in GNSS) and $\mathbf{W}_{\hat{\mathbf{x}}}\in\mathbb{R}^{n\times m}$ is its covariance matrix, which is symmetric positive definite. The ILS estimate of the integer parameter vector is the solution of the minimisation problem

$$\min_{\mathbf{x}\in\mathbb{Z}^n}(\mathbf{x}-\hat{\mathbf{x}})^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}^{-1}(\mathbf{x}-\hat{\mathbf{x}}) \tag{1}$$

Although equation (1) is in the form of an integer quadratic optimisation problem, it is easy to rewrite it in the standard ILS form

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

In fact, suppose $\mathbf{W}_{\hat{\mathbf{x}}}^{-1}$ has the Cholesky factorisation $\mathbf{W}_{\hat{\mathbf{x}}}^{-1}=\mathbf{R}^{\mathrm{T}}\mathbf{R}$, where $\mathbf{R}$ is upper triangular, then, with $\mathbf{y}=\mathbf{R}\hat{\mathbf{x}}$ and $\mathbf{A}=\mathbf{R}$, problem (1) can be written as problem (2). Conversely, an ILS problem in the standard form problem (2) with $\mathbf{A}$ being of full column rank can be transformed to problem (1). Let $\mathbf{A}\in\mathbb{R}^{n\times m}$ have the QR factorisation $\mathbf{A}=[\mathbf{Q}_1,\mathbf{Q}_2]\begin{bmatrix}\boldsymbol{R}\\0\end{bmatrix}=\mathbf{Q}_1\mathbf{R}$, where $[\mathbf{Q}_1,\mathbf{Q}_2]\in\mathbb{R}^{m\times m}$ is orthogonal and $\mathbf{R}\in\mathbb{R}^{n\times n}$ is upper triangular. Then

$$\begin{aligned}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2&=\left\|[\mathbf{Q}_1,\mathbf{Q}_2]^{\mathrm{T}}(\mathbf{A}\mathbf{x}-\mathbf{y})\right\|_2^2\\&=\left\|\mathbf{R}\mathbf{x}-\mathbf{Q}_1^{\mathrm{T}}\mathbf{y}\right\|_2^2+\left\|\mathbf{Q}_2^{\mathrm{T}}\mathbf{y}\right\|_2^2\\&=\left\|\mathbf{R}(\mathbf{x}-\hat{\mathbf{x}})\right\|_2^2+\left\|\mathbf{Q}_2^{T}\mathbf{y}\right\|_2^2\end{aligned}$$

Thus, with $\mathbf{W}_{\hat{\mathbf{x}}}=\mathbf{R}^{\mathrm{T}}\mathbf{R}^{-1}$, problem (2) can be transformed to problem (1). But we want to point out that it may not be numerically reliable to transform problems (1) to (2), or vice versa. As the quadratic form of the ILS problem (1) is often used in the GNSS literature, for the sake of comparison convenience, we also use it in this paper. An approach to solving problem (1) can be modified to solve problem (2) without using the transformation mentioned above.

In the following two subsections, we discuss the reduction or decorrelation process used in the GNSS literature and briefly review the search process.

Reduction

The reduction process uses a unimodular matrix $\mathbf{Z}$ to transform equation (1) into

$$\min_{\mathbf{z}\in\mathbb{Z}^n}(\mathbf{z}-\hat{\mathbf{z}})^{\mathrm{T}}\mathbf{W}_{\mathbf{z}}^{-1}(\mathbf{z}-\hat{\mathbf{z}}), \tag{3}$$

where $\mathbf{z}=\mathbf{Z}^{\mathrm{T}}\mathbf{x}$, $\hat{\mathbf{z}}=\mathbf{Z}^{\mathrm{T}}\hat{\mathbf{x}}$ and $\mathbf{W}_{\hat{\mathbf{z}}}=\mathbf{Z}^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{Z}$. If $\hat{\mathbf{x}}$ is the minimiser of problem (3), then $\breve{\mathbf{x}}=\mathbf{Z}^{-\mathrm{T}}\breve{\mathbf{z}}$ is the minimiser of problem (1). The benefit the reduction process brings is that the discrete search process for solving the new optimisation problem (3) can be much more efficient by choosing an appropriate $\mathbf{Z}$.

Let the $\mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{z}}}$ be

$$\mathbf{W}_{\hat{\mathbf{z}}}=\mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L} \tag{4}$$

where $\mathbf{L}=(l_{ij})$ is unit lower triangular and $\mathbf{D}=\mathrm{diag}\,(d_1,\ldots,d_n)$ with $d_i>0$. These factors have a statistical interpretation. Let $\bar{z}_j$ denote the RLS estimate of z_j when $z_{j+1},\ldots,z_n$ are fixed. It is easy to show (cf. [21, p. 337]) that $d_i=\sigma_{\bar{z}_i}^2$, where $\sigma_{\bar{z}_i}^2$ is the variance of $\bar{z}_i$, and $l_{ij}=\sigma_{\bar{z}_i\hat{z}_j}\sigma_{\bar{z}_i}^{-2}$ for $i>j$, where $\sigma_{\bar{z}_i\hat{z}_j}$ denotes the covariance between $\bar{z}_i$ and $\hat{z}_j$.

In the literature, see, e.g., [**4**], [**6**], [**16**, p. 498], and [**21**], it is often mentioned that the following two objectives should be pursued in the reduction process, because it is believed that they are crucial for the efficiency of the search process:

(i) $\mathbf{W}_{\hat{\mathbf{z}}}$ is as diagonal as possible. The motivation is that if $\mathbf{W}_{\hat{\mathbf{z}}}$ is a diagonal matrix, i.e., the entries of $\hat{\mathbf{z}}$ are uncorrelated to each other, then simply setting $z_i=\lfloor\hat{z}_i\rceil$, for $i=1,2,\ldots,n$, would solve equation (3). Thus, it is assumed that the search process would be fast if $\mathbf{W}_{\hat{\mathbf{z}}}$ is nearly diagonal. That a covariance matrix is close to diagonal means that there is little correlation between its random variables. In other words, an objective of the reduction is to decorrelate the RLS estimates of the ambiguities as much as possible. For example in [**6**, sec. 3·7], it stated that 'For the actual integer minimisation we strive for largely decorrelated ambiguities'.

(ii) The diagonal entries of $\mathbf{D}$ are distributed in decreasing order if possible, i.e., one strives for

$$d_1\gg d_2\gg\cdots\gg d_n \tag{5}$$

Note that $d_1d_2\cdots d_n=\det(\mathbf{W}_{\hat{\mathbf{z}}})=\det(\mathbf{W}_{\hat{\mathbf{x}}})$, which is invariant with respect to the unimodular matrix

$\mathbf{Z}$. The ordering of d_j is also known as the signature of the spectrum of conditional variances. This objective will flatten the spectrum. The importance of this objective was explained in detail in [**17**], [**19**], [**21**].

LAMBDA's reduction process and MLAMBDA's reduction process as well start with the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{x}}}$ and updates the factors to give the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{z}}}$ by using integer Gauss transformations (IGTs) and permutations, both of which are unimodular matrices. The IGTs are used to make the off-diagonal entries of $\mathbf{L}$ as small as possible, while permutations are used to strive for order (5). Specifically, after reduction, the L-factor of the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{z}}}$ satisfies the conditions of the well-known Lenstra-Lenstra-Lovász (LLL) reduction [**10**]

$$|l_{ij}| \le 1/2, \quad i = j+1, \ldots, n, j = 1, \ldots, n-1, \tag{6}$$

$$d_j + l_{j+1,j}^2 d_{j+1} \ge \alpha d_{j+1}, \quad j = 1, \ldots, n-1, \tag{7}$$

with $\alpha = 1$. In the general LLL reduction, the parameter α satisfies $1/4 < \alpha \le 1$. The inequality (6) is referred to the size reduced condition and the inequality (7) is often called Lovász's condition.

We will show that, contrary to the common belief, decorrelations done to pursue the first objective may not make the search process more efficient unless they help to achieve the second objective. The second objective is very crucial for the search speed, but it is not mentioned in some GNSS literature.

In some papers such as [**13**], [**14**] and [**25**], instead of (i) and (ii), decreasing the condition number of the covariance matrix is regarded as an objective of the reduction process. Although this may be helpful for achieving the second objective, we will argue later that it should not be an objective of the reduction.

In the following, we introduce the integer Gauss transformations and permutations, which will be used later in describing algorithms.

Integer Gauss transformations

An integer Gauss transformation $\mathbf{Z}_{ij}$ has the following form

$$\mathbf{Z}_{ij} = \mathbf{I} - \mu \mathbf{e}_i \mathbf{e}_j^T, \ \mu \text{ is an integer} \tag{8}$$

Applying $\mathbf{Z}_{ij}$ ($i > j$) to $\mathbf{L}$ from the right gives

$$\bar{\mathbf{L}} \equiv \mathbf{L}\mathbf{Z}_{ij} = \mathbf{L} - \mu \mathbf{L}\mathbf{e}_i \mathbf{e}_j^T$$

Thus $\bar{\mathbf{L}}$ is the same as $\mathbf{L}$, except that

$$\bar{l}_{kj} = l_{kj} - \mu l_{ki}, \quad k = i, \ldots, n$$

To make $\bar{l}_{ij}$ as small as possible, we choose $\mu = \lfloor l_{ij} \rceil$, which ensures that

$$|\bar{l}_{ij}| \le 1/2, \quad i > j \tag{9}$$

We use the following algorithm (see [**4**]) to apply the IGT $\mathbf{Z}_{ij}$ to transform the ILS problem.

Algorithm 1. (Integer Gauss Transformations). Given a unit lower triangular $\mathbf{L} \in \mathbb{R}^{n \times n}$, index pair (i, j), $\hat{\mathbf{x}} \in \mathbb{R}^n$ and $\mathbf{Z} \in \mathbb{Z}^{n \times n}$. This algorithm first applies the integer Gauss transformation $\mathbf{Z}_{ij}$ to $\mathbf{L}$ such that $|(\mathbf{L}\mathbf{Z})_{i,j}| \le 1/2$, then computes $\mathbf{Z}_{ij}^T\hat{\mathbf{x}}$ and $\mathbf{Z}\mathbf{Z}_{ij}$, which overwrite $\hat{\mathbf{x}}$ and $\mathbf{Z}$, respectively.

function: $[\mathbf{L}, \hat{\mathbf{x}}, \mathbf{Z}] = \text{Gauss}(\mathbf{L}, i, j, \hat{\mathbf{x}}, \mathbf{Z})$
$\mu = \lfloor l_{ij} \rceil$
if $\mu \neq 0$
  $\mathbf{L}_{i:n,j} = \mathbf{L}_{i:n,j} - \mu \mathbf{L}_{i:n,i}$
  $\mathbf{Z}_{1:n,j} = \mathbf{Z}_{1:n,j} - \mu \mathbf{Z}_{1:n,i}$
  $\hat{x}_j = \hat{x}_j - \mu \hat{x}_i$
end

Permutations

In order to strive for order (5), symmetric permutations of the covariance matrix $\mathbf{W}_{\hat{\mathbf{x}}}$ are needed in the reduction process. After a permutation, the factors $\mathbf{L}$ and $\mathbf{D}$ of the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation have to be updated.

If we partition the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{x}}}$ as follows

$$\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}^T\mathbf{D}\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11}^T & \mathbf{L}_{21}^T & \mathbf{L}_{31}^T \\ & \mathbf{L}_{22}^T & \mathbf{L}_{32}^T \\ & & \mathbf{L}_{33}^T \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & & \\ & \mathbf{D}_2 & \\ & & \mathbf{D}_3 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11} & & \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \begin{matrix} k-1 \\ 2 \\ n-k-1 \end{matrix}$$
$$\begin{matrix} k-1 & 2 & n-k-1 \end{matrix}$$

Let

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{P}_{k,k+1} = \begin{bmatrix} \mathbf{I}_{k-1} & & \\ & \mathbf{P} & \\ & & \mathbf{I}_{n-k-1} \end{bmatrix} \tag{10}$$

It can be shown that $\mathbf{P}_{k,k+1}^T \mathbf{W}_{\hat{\mathbf{x}}} \mathbf{P}_{k,k+1}$ has the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation (cf. [**6**])

$$\mathbf{P}_{k,k+1}^T = \mathbf{W}_{\hat{\mathbf{x}}} \mathbf{P}_{k,k+1} = \begin{bmatrix} \mathbf{L}_{11}^T & \bar{\mathbf{L}}_{21}^T & \mathbf{L}_{31}^T \\ & \bar{\mathbf{L}}_{22}^T & \bar{\mathbf{L}}_{32}^T \\ & & \mathbf{L}_{33}^T \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 & & \\ & \bar{\mathbf{D}}_2 & \\ & & \mathbf{D}_3 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11} & & \\ \bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22} & \\ \mathbf{L}_{31} & \bar{\mathbf{L}}_{32} & \mathbf{L}_{33} \end{bmatrix} \tag{11}$$

where

$$\bar{\mathbf{D}}_2 = \begin{bmatrix} \bar{d}_k & \\ & \bar{d}_{k+1} \end{bmatrix} \tag{12}$$
$$\bar{d}_{k+1} = d_k + l_{k+1,k}^2 d_{k+1}, \quad \bar{d}_k = \frac{d_k}{\bar{d}_{k+1}} d_{k+1}$$

$$\bar{\mathbf{L}}_{22} \equiv \begin{bmatrix} 1 & \\ \bar{l}_{k+1,k} & 1 \end{bmatrix}, \quad \bar{l}_{k+1,k} = \frac{d_{k+1} l_{k+1,k}}{\bar{d}_{k+1}} \tag{13}$$

$$\mathbf{L}_{21} = \begin{bmatrix} -l_{k+1,k} & 1 \\ \frac{d_k}{\bar{d}_{k+1}} & \bar{l}_{k+1,k} \end{bmatrix} \mathbf{L}_{21} = \begin{bmatrix} -l_{k+1,k} & 1 \\ \frac{d_k}{\bar{d}_{k+1}} & \bar{l}_{k+1,k} \end{bmatrix} \mathbf{L}_{k:k+1,1:k-1} \tag{14}$$

$$\bar{\mathbf{L}}_{32} = \mathbf{L}_{32}\mathbf{P} = [\mathbf{L}_{k+2:n,k+1}, \mathbf{L}_{k+2:n,1:k}] \tag{15}$$

We refer to such an operation as a permutation of pair $(k, k+1)$. We describe the process as an algorithm (see [**4**]).

Algorithm 2. (Permutations). Given the $\mathbf{L}$ and $\mathbf{D}$ factors of the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{x}}} \in \mathbb{R}^{n \times n}$, index k, scalar δ which is equal to $\bar{d}_{k+1}$ in equation (12), $\hat{\mathbf{x}} \in \mathbb{R}^n$, and $\mathbf{Z} \in \mathbb{Z}^{m \times n}$. This algorithm computes the updated $\mathbf{L}$ and $\mathbf{D}$

factors in equation (11) after rows and columns k and $k+1$ of $\mathbf{W}_{\hat{\mathbf{x}}}$ are interchanged. It also interchanges entries k and $k+1$ of $\hat{\mathbf{x}}$ and columns $_k$ and $k+1$ of $\mathbf{Z}$.

function: $[\mathbf{L},\mathbf{D},\hat{\mathbf{x}},\mathbf{Z}] = \text{Permute}(\mathbf{L},\mathbf{D},k,\delta,\hat{\mathbf{x}},\mathbf{Z})$
$\eta = d_k/\delta$ //see (12)
$\lambda = d_{k+1,k+1}l_{k+1,k}/\delta$ //see (13)
$d_k = \eta d_{k+1,k+1}$ //see (12)
$d_{k+1,k+1} = \delta$
$\mathbf{L}_{k:k+1,1:k-1} = \begin{bmatrix} -l_{k+1,k} & 1 \\ \eta & \lambda \end{bmatrix} \mathbf{L}_{k:k+1,1:k-1}$ //see (14)
$l_{k+1,k} = \lambda$
Swap columns $\mathbf{L}_{k+2:n,k}$ and $\mathbf{L}_{k+2:n,k+1}$ //see (15)
Swap columns $\mathbf{Z}_{1:n,k}$ and $\mathbf{Z}_{1:n,k+1}$
Swap entries $\hat{x}_k$ and $\hat{x}_{k+1}$

Search

In this subsection, we briefly review the often used discrete search process (see [**6**] and [**15**]) in solving an ILS problem. Substituting the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation (4) into (3), we obtain

$$\min_{\mathbf{z}\in\mathbb{Z}^n} (\mathbf{z}-\hat{\mathbf{z}})^T\mathbf{L}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-T}(\mathbf{z}-\hat{\mathbf{z}}) \tag{16}$$

Define $\bar{\mathbf{z}}$ as

$$\bar{\mathbf{z}} = \mathbf{z} - \mathbf{L}^{-T}(\mathbf{z}-\hat{\mathbf{z}}) \tag{17}$$

Thus we have

$$\mathbf{L}^T(\mathbf{z}-\bar{\mathbf{z}}) = (\mathbf{z}-\hat{\mathbf{z}})$$

which can be expanded to

$$\bar{z}_j = \hat{z}_j + \sum_{i=j+1}^{n} l_{ij}(z_i - \bar{z}_i), \quad j = n, n-1, \ldots, 1 \tag{18}$$

where when $j=n$, the summation term vanishes. Observe that $\bar{z}_j$ depends on $z_{j+1},\ldots,z_n$. Actually $\bar{z}_j$ is the RLS estimate of z_j when $z_{j+1},\ldots,z_n$ are fixed. With equation (17), we can rewrite the optimisation problem (16) as

$$\min_{\mathbf{z}\in\mathbb{Z}^n} (\mathbf{z}-\bar{\mathbf{z}})^T\mathbf{D}^{-1}(\mathbf{z}-\bar{\mathbf{z}}) \tag{19}$$

or equivalently

$$\min_{\mathbf{z}\in\mathbb{Z}^n} \sum_{j=1}^{n} (z_j - \bar{z}_j)^2/d_j \tag{20}$$

Assume that the solution of equation (20) satisfies the bound

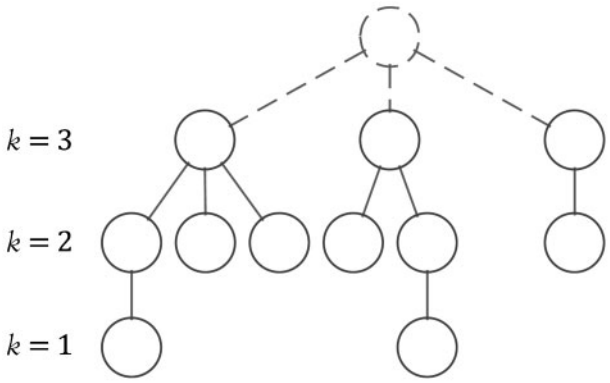
$$\sum_{j=1}^{n} \frac{(z_j-\bar{z}_j)^2}{d_j} < \chi^2 \tag{21}$$

for some χ. Note that equation (21) is a hyper-ellipsoid, which is referred to as an ambiguity search space. If $\mathbf{z}$ satisfies equation (21), then it must also satisfy inequalities

$$\text{level } k: \quad \frac{(z_k-\bar{z}_k)^2}{d_k} < \chi^2 - \frac{\sum_{j=k+1}^{n} (z_j-\bar{z}_j)^2}{d_j} \tag{22}$$

for $k=n,\ n-1,\ldots,1$. From inequality (22), the range of z_k is $[l_k, u_k]$, where

$$l_k = \left\lceil \bar{z}_k - d_k^{1/2}(\chi^2 - \sum_{j=k+1}^{n} (z_j-\bar{z}_j)^2/d_j)^{1/2} \right\rceil \tag{23}$$



1 Search tree

$$u_k = \left\lfloor \bar{z}_k + d_k^{1/2}(\chi^2 - \sum_{j=k+1}^{n} (z_j-\bar{z}_j)^2/d_j)^{1/2} \right\rfloor \tag{24}$$

The search process starts at level n and moves down to level 1. Suppose that $z_n,\ldots,z_{k+1}$ have been fixed. At level k, if $l_k > u_k$, then there is no integer satisfying the inequality (22) and the search process moves back to level $k+1$; otherwise it chooses $z_k = \lceil \bar{z}_k \rceil$, which is in $[l_k, u_k]$, and moves to level $k-1$. If at level $k-1$ it cannot find any integer in $[l_{k-1}, u_{k-1}]$ it moves back to level k and tries to find the next nearest integer to $\bar{z}_k$ in $[l_k, u_k]$. In general the enumeration order at level k is as follows

$$z_k = \begin{cases} \lceil \bar{z}_k \rceil, \lfloor \bar{z}_k \rceil - 1, \lceil \bar{z}_k \rceil + 1, \ldots, & \text{if } \bar{z}_k \le \lfloor \bar{z}_k \rceil \\ \lfloor \bar{z}_k \rceil, \lceil \bar{z}_k \rceil + 1, \lfloor \bar{z}_k \rceil - 1, \ldots, & \text{otherwise} \end{cases} \tag{25}$$

When an integer point, say $\mathbf{z}^*$, is found, update χ^2 by setting

$$\chi^2 = (\mathbf{z}^* - \hat{\mathbf{z}})^T\mathbf{W}_{\hat{\mathbf{z}}}^{-1}(\mathbf{z}^* - \hat{\mathbf{z}})$$

and the search process tries to update $\mathbf{z}^*$ to find an integer point within the new hyper-ellipsoid.

The initial χ^2 can be set to infinity. With initial $\chi^2 = \infty$, the first integer point found in the search process is referred to as the Babai integer point (see [**2**] and [**4**]) or the bootstrapping estimate (see [**23**]). The search process is actually a depth-first search in a tree, see Fig. 1, where $n=3$. Each node in the tree, except for the root node, represents an actual step in the search process – assigning a value to x_k. In Fig. 1, each leaf at level 1 corresponds to an integer point found in the search process and leaves at other levels correspond to invalid integer points.

For the extension of the search process to find more than one optimal solutions to problem (3), see [**4**] and [**6**].

Impact of IGTs on the search process

As seen in the previous section, according to the GNSS literature, one of the two objectives of the reduction process is to decorrelate the ambiguities as much as possible. Decorrelating the ambiguities as much as possible means making the covariance matrix as diagonal as possible. To achieve this, a natural way, as given in the literature, is to make the absolute values of the off-diagonal entries of the L-factor of the covariance matrix as small as possible by using IGTs. In the following, we rigorously show that solely reducing the off-diagonal

entries of $\mathbf{L}$ will have no impact on the search process given in the previous section.

Theorem 1 *Given the ILS problem* (1) *and the reduced ILS problem* (3), *if the unimodular transformation matrix* $\mathbf{Z}$ *is a product of lower triangular IGTs, then the search process for solving problem* (1) *is as efficient as the search process for solving problem* (3).

Proof. We will show that the structure of the search tree is not changed after the transformation.

Let the $\mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L}$ factorisations of $\mathbf{W}_{\hat{\mathbf{x}}}$ and $\mathbf{W}_{\hat{\mathbf{z}}}$ be

$$\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L}, \quad \mathbf{W}_{\hat{\mathbf{z}}} = \bar{\mathbf{L}}^{\mathrm{T}}\bar{\mathbf{D}}\bar{\mathbf{L}}$$

As shown in previous section, the ILS problems (1) and (3) can respectively be written as (cf. equation (20))

$$\min_{x\in\mathbb{Z}^{\mathrm{n}}} \sum_{\mathrm{j}=1}^{\mathrm{n}} (x_{\mathrm{j}} - \bar{x}_{\mathrm{j}})^2/d_{\mathrm{j}} \tag{26}$$

where

$$\bar{x}_{\mathrm{j}} = \hat{\bar{x}}_{\mathrm{j}} + \sum_{\mathrm{i}=\mathrm{j}+1}^{\mathrm{n}} l_{\mathrm{ij}}(x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}) \tag{27}$$

and

$$\min_{z\in\mathbb{Z}^{\mathrm{n}}} \sum_{\mathrm{j}=1}^{\mathrm{n}} (z_{\mathrm{j}} - \bar{z}_{\mathrm{j}})^2/\bar{d}_{\mathrm{j}} \tag{28}$$

where

$$\bar{z}_{\mathrm{j}} = \hat{\bar{z}}_{\mathrm{j}} + \sum_{\mathrm{i}=\mathrm{j}+1}^{\mathrm{n}} \bar{l}_{\mathrm{ij}}(z_{\mathrm{i}} - \bar{z}_{\mathrm{i}}) \tag{29}$$

We first consider the case where $\mathbf{Z}$ is a single lower triangular IGT $\mathbf{Z}_{\mathrm{st}} = \mathbf{I} - \mu\mathbf{e}_{\mathrm{s}}\mathbf{e}_{\mathrm{t}}^{\mathrm{T}}$ with $s>t$, which is applied to $\mathbf{L}$ from the right to reduce l_{st}. Then we have

$$\bar{\mathbf{D}} = \mathbf{D}, \quad \bar{\mathbf{L}} = \mathbf{L}\mathbf{Z}_{\mathrm{st}} = \mathbf{L} - \mu\mathbf{L}\mathbf{e}_{\mathrm{s}}\mathbf{e}_{\mathrm{t}}^{\mathrm{T}}$$

where

$$\bar{l}_{\mathrm{it}} = l_{\mathrm{it}}, \ \ i = t, t+1, \ldots, s-1 \tag{30}$$

$$\bar{l}_{\mathrm{it}} = l_{\mathrm{it}} - \mu l_{\mathrm{is}}, \quad i = s, s+1, \ldots, n \tag{31}$$

$$\bar{l}_{\mathrm{ij}} = l_{\mathrm{ij}}, \ \ i = j, j+1, \ldots, n, \quad j \neq t \tag{32}$$

With $\hat{\bar{\mathbf{z}}} = \mathbf{Z}_{\mathrm{st}}^{\mathrm{T}}\hat{\mathbf{x}}$, we have

$$\hat{\bar{z}}_{\mathrm{i}} = \begin{cases} \hat{\bar{x}}_{\mathrm{i}}, & \text{if } i \neq t \\ \hat{\bar{x}}_{\mathrm{i}} - \mu\hat{\bar{x}}_{\mathrm{s}}, & \text{if } i = t \end{cases} \tag{33}$$

Suppose that in the search process $x_{\mathrm{n}}, x_{\mathrm{n}-1}, \ldots, x_{\mathrm{k}+1}$ and $z_{\mathrm{n}}, z_{\mathrm{n}-1}, \ldots, z_{\mathrm{k}+1}$ have been fixed. We consider the search process at level k. At level k, the inequalities need to be checked are respectively

$$(x_{\mathrm{k}} - \bar{x}_{\mathrm{k}})^2/d_{\mathrm{k}} < \chi^2 - \sum_{\mathrm{j}=\mathrm{k}+1}^{\mathrm{n}} (x_{\mathrm{j}} - \bar{x}_{\mathrm{j}})^2/d_{\mathrm{j}} \tag{34}$$

$$(z_{\mathrm{k}} - \bar{z}_{\mathrm{k}})^2/d_{\mathrm{k}} < \chi^2 - \sum_{\mathrm{j}=\mathrm{k}+1}^{\mathrm{n}} (z_{\mathrm{j}} - \bar{z}_{\mathrm{j}})^2/d_{\mathrm{j}} \tag{35}$$

There are three cases:

Case 1: $k>t$. Note that $\mathbf{L}_{\mathrm{k:n,k:n}} = \bar{\mathbf{L}}_{\mathrm{k:n,k:n}}$. From equations (27), (29) and (33), it is easy to conclude that we have $\bar{x}_{\mathrm{i}} = \bar{z}_{\mathrm{i}}$ and $x_{\mathrm{i}} = z_{\mathrm{i}}$ for $i=n, n-1, \ldots, k+1$. Thus, at level k, $\bar{x}_{\mathrm{i}} = \bar{z}_{\mathrm{i}}$ and the search process takes an identical value for x_{k} and z_{k}. For the chosen value, the two inequalities (34) and (35) are identical. So both hold or fail at the same time.

Case 2: $k=t$. According to Case 1, we have $x_{\mathrm{i}} = z_{\mathrm{i}}$ and $\bar{x}_{\mathrm{i}} = \bar{z}_{\mathrm{i}}$ for $i=n, n-1, \ldots, t+1$. Thus, by equations (29), (30), (31), (33) and (27)

$$\begin{aligned}
\bar{z}_{\mathrm{t}} &= \hat{\bar{z}}_{\mathrm{t}} + \sum_{\mathrm{i}=\mathrm{t}+1}^{\mathrm{n}} \bar{l}_{\mathrm{it}}(z_{\mathrm{i}} - \bar{z}_{\mathrm{i}}) \\
&= \hat{\bar{x}}_{\mathrm{t}} - \mu\hat{\bar{x}}_{\mathrm{s}} + \sum_{\mathrm{i}=\mathrm{t}+1}^{\mathrm{s}-1} l_{\mathrm{it}}(x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}) + \sum_{\mathrm{i}=\mathrm{s}}^{\mathrm{n}} (l_{\mathrm{it}} - \mu l_{\mathrm{is}})(x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}) \\
&= \hat{\bar{x}}_{\mathrm{t}} + \sum_{\mathrm{i}=\mathrm{t}+1}^{\mathrm{n}} l_{\mathrm{it}}(x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}) - \mu[\hat{\bar{x}}_{\mathrm{s}} + \sum_{\mathrm{i}=\mathrm{s}+1}^{n} l_{\mathrm{is}}(x_{\mathrm{i}} - x_{\mathrm{i}})] - \mu(x_{\mathrm{s}} - \bar{x}_{\mathrm{s}}) \\
&= \bar{x}_{\mathrm{t}} - \mu\bar{x}_{\mathrm{s}} - \mu(x_{\mathrm{s}} - \bar{x}_{\mathrm{s}}) \\
&= \bar{x}_{\mathrm{t}} - \mu x_{\mathrm{s}},
\end{aligned}$$

where μx_{s} is an integer. Since z_{t} and x_{t} respectively take values according to the same order (cf. (25)), the values of z_{t} and x_{t} chosen by the search process must satisfy $z_{\mathrm{t}} = x_{\mathrm{t}} - \mu x_{\mathrm{s}}$ Thus, $z_{\mathrm{t}} - \bar{z}_{\mathrm{t}} = x_{\mathrm{t}} - \bar{x}_{\mathrm{t}}$, and again the two inequalities (34) and (35) hold or fail at the same time.

Case 3: $k<t$. According to Cases 1 and 2, $z_{\mathrm{i}} - \bar{z}_{\mathrm{i}} = x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}$ for $i=n, n-1, \ldots, t$. Then for $k=t-1$, by equations (29), (32) and (27)

$$\bar{z}_{\mathrm{k}} = \hat{\bar{z}}_{\mathrm{k}} + \sum_{\mathrm{i}=\mathrm{k}+1}^{\mathrm{n}} \bar{l}_{\mathrm{ik}}(z_{\mathrm{i}} - \bar{z}_{\mathrm{i}}) = \hat{\bar{x}}_{\mathrm{k}} + \sum_{\mathrm{i}=\mathrm{k}+1}^{\mathrm{n}} l_{\mathrm{ik}}(x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}) = \bar{x}_{\mathrm{k}}$$

Thus, the search process takes an identical value for z_{k} and x_{k} when $k=t-1$. By induction we can similarly show this is true for a general $k<t$. Thus, again inequalities (34) and (35) hold or fail at the same time.

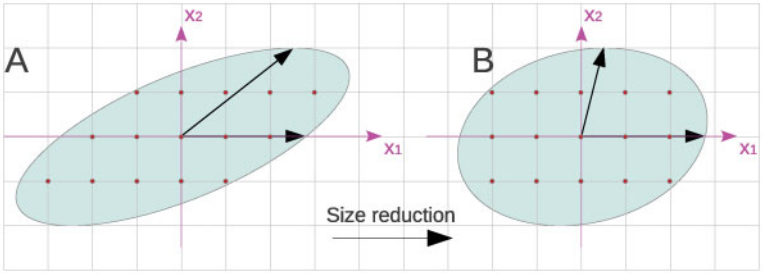
The above has shown that the two search trees for problems (1) and (3) have identical structures.

Now we consider the general case where $\mathbf{Z}$ is a product of lower triangular IGTs, i.e., $\mathbf{Z} = \mathbf{Z}_1 \ldots \mathbf{Z}_{\mathrm{m}}$. As is shown, applying Z_1 to problem (1) will transform the ILS problem, but not modify the structure of the search tree. Applying $\mathbf{Z}_2$ to this transformed ILS problem will also not modify the structure of the search tree, and so on. Thus, if $\mathbf{Z}$ is a product of lower triangular IGTs, the search trees for problems (1) and (3) have the same structure.

It is easy to observe that the computational costs for fixing x_{k} and z_{k} at each level k in search are the same. Therefore we can conclude that the two search processes have the same computational efficiency.

Here we make a remark. Obviously, all diagonal entries of $\mathbf{L}$ remain invariant under integer Gauss transformations. Thus the complete spectrum of ambiguity conditional variances, d_{i}'s in order (5), remains untouched as stated in [**19**]. But whether the search speed solely depends on the spectrum was not proved.

From the proof of Theorem 1, we can easily observe that the search ordering given in equation (25) may not be necessary, i.e., Theorem 1 may still hold if the ordering is in a different way. For example, Theorem 1 still holds if the enumerated integers at level k are ordered from left to right in the interval $[l_{\mathrm{k}}, u_{\mathrm{k}}]$ – this enumeration strategy is used in the LAMBDA package. Theorem 1 also holds when the shrinking technique is



2 Size reduction

employed. This can be seen from the fact that the residues of the corresponding integer points found in the two search trees are always equal, i.e.

$$\sum_{j=1}^{n}\left(x_j-\bar{x}_j\right)^2/d_j=\sum_{j=1}^{n}(z_j-\bar{z}_j)^2/d_j$$

We would like to point out that [**12**] gave a geometric argument that the Babai integer point (i.e. the bootstrapping estimate) encountered in solving the standard form of the ILS problem (2), referred to as the successive interference cancellation decoder in communications, is not affected by the size reduction of the off-diagonal entries (except the super-diagonal entries) of **R** of the QR factorisation of **A**. Our result given in Theorem 1 is more general, because the Babai integer point is the first integer point found in the search process.

In [**8**, sec. 3·5], to prove that the lattice reduction speeds up searching process, the authors tried to prove that the more orthogonality the columns of **R**, the less the number of candidates for searching. But the proof is not correct because the number of candidates is not closely dependent on the column orthogonality of **R**. In fact, a series of IGTs can be found to reduce all the off-diagonal entries of **R**. Applying them to **R** will make **R** more orthogonal. But according to Theorem 1, this can neither reduce the number of candidates nor speed up the search process.

Theorem 1 showed that modifying the unit lower triangular matrix **L** by a unimodular matrix will not change the computational efficiency of the search process. Here we first give a geometric interpretation for the general two-dimensional case, then give a specific two-dimensional integer search example to illustrate it.

Suppose the ambiguity search space is the ellipse given in Fig. 2*a*, with the integer candidates marked. The search process will try to fix x_2 and then fix x_1 for each value of x_2. After the size reduction, we get Fig. 2*b*. As explained in [**21**], the result of this transformation is to push the vertical tangents of the ambiguity search space. The horizontal tangents are unchanged. Note that for each integer value of x_2, the number of choices for x_1 is not changed after the size reduction. This indicates the size reduction does not change the search speed, although the elongation of the ellipse becomes smaller.

Example 1 Let the covariance matrix be

$$\mathbf{W}_{\hat{\mathbf{x}}}=\begin{bmatrix}11026 & 1050\\ 1050 & 100\end{bmatrix}$$

The factors **L** and **D** of its $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation are

$$\mathbf{L}=\begin{bmatrix}1 & 0\\ 10{\cdot}5 & 1\end{bmatrix},\quad \mathbf{D}=\begin{bmatrix}1 & 0\\ 0 & 100\end{bmatrix}$$

Let the RLS estimate be $\hat{\mathbf{x}}=[5{\cdot}38,\ 18{\cdot}34]^T$. We can reduce l_{21} by the IGT $\mathbf{Z}=\begin{bmatrix}1 & 0\\ -10 & 1\end{bmatrix}$. Then the modified covariance matrix and its **L** factor become

$$\mathbf{W}_{\hat{\mathbf{z}}}=\mathbf{Z}^T\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{Z}=\begin{bmatrix}26 & 50\\ 50 & 100\end{bmatrix},\quad \bar{\mathbf{L}}=\mathbf{L}\mathbf{Z}=\begin{bmatrix}1 & 0\\ 0{\cdot}5 & 1\end{bmatrix}$$

In [**21**], the correlation coefficient ρ and the elongation of the search space e are used to quantify the correlation between the ambiguities. The correlation coefficient ρ between random variables s_1 and s_2 is defined as $\rho=\sigma_{s_1\,s_2}/(\sigma_{s_1}\,\sigma_{s_2})$; see, e.g., [**16**, p. 322]. The elongation of the search space e is given by square root of the 2-norm condition number of the covariance matrix. For the original ambiguity vector, we have $\rho=0{\cdot}9995$ and $e=1{\cdot}1126\times10^3$. For the transformed ambiguity vector, we have $\rho=0{\cdot}9806$ and $e=12{\cdot}520$. These measurements indicate that the transformed RLS estimates of ambiguities are more decorrelated. The points $[x_1,\ x_2]^T$ and $[z_1,\ z_2]^T$ encountered during search are shown in Table 1, which indicates that no valid integer is found. In both cases, the first point encountered is valid, while the others points are invalid. The ILS solution is $\mathbf{x}=[2,18]^T$. As expected, we observe that the lower triangular IGT did not reduce the number of points (including invalid points) encountered in the search process.

We have already shown that solely decorrelating the RLS estimates of the ambiguities by applying lower triangular IGTs to the L-factor of the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of the covariance matrix will not help the search process. However, we will show that some IGTs are still useful for improving the efficiency of the search process. The search speed depends on the **D** factor. The more flattened the spectrum of ambiguity conditional variances, the more efficient the search (see, e.g. [**19**]). We strive for order (5) in the reduction process. If an IGT is helpful in striving for order (5), then this IGT is useful for improving the search speed. When we permute pair $(k,k+1)$ in the reduction process, **D** is modified according to equation (12). In order to make $\bar{d}_{k+1}$ as small as possible, from equation (12), we observe that $|l_{k+1,k}|$ should be made as small as possible. An example would be helpful to show this.

Table 1 Search results for Example 1

x_1	x_2	z_1	z_2
2	18	−178	18
…	19	…	19
…	17	…	17
…	20	…	20
…	…	…	…

Example 2 Let the **L**-factor and **D**-factor of the $\mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L}$ factorisation of a 2×2 covariance matrix $\mathbf{W}_{\hat{\mathbf{x}}}$ be

$$\mathbf{L}=\begin{bmatrix}1 & 0\\ 0{\cdot}8 & 1\end{bmatrix},\quad \mathbf{D}=\begin{bmatrix}1 & 0\\ 0 & 100\end{bmatrix}$$

We have $d_1=1$ and $d_2=100$. Let the RLS estimate be $\hat{\mathbf{x}}=[13{\cdot}5,1{\cdot}2]^{\mathrm{T}}$. The number of integer points $[x_1,\ x_2]^{\mathrm{T}}$ encountered in the search process without reduction is shown in Table 2. If we permute the two ambiguities without first applying an IGT, i.e. $\mathbf{Z}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, using equation (12), we have $\bar{d}_2=65$ and $\bar{d}_1=1{\cdot}54$. The search process will be more efficient after this transformation because $\bar{d}_2<d_2$ allows more pruning to occur (see order (5)). The integer pairs $[z_1,z_2]^{\mathrm{T}}$ encountered during the search are given in Table 2. The ILS solution is $\breve{\boldsymbol{x}}=\mathbf{Z}^{-\mathrm{T}}[2,14]^{\mathrm{T}}=[14,2]^{\mathrm{T}}$ However, we can make $\bar{d}_2$ even smaller by applying a lower triangular IGT before the permutation, which means that $\mathbf{Z}=\begin{bmatrix}0 & 1\\ 1 & -1\end{bmatrix}$. In this case, we have $\bar{d}_2=5$ and $\bar{d}_1=20$. The integer pairs $[\tilde{z}_1,\tilde{z}_2]$ encountered in search are given in the last two columns of Table 2. The ILS solution is $\breve{\boldsymbol{x}}=\mathbf{Z}^{-\mathrm{T}}[2,12]^{\mathrm{T}}=[14,2]^{\mathrm{T}}$. This example illustrates how a lower triangular IGT, followed by a permutation, can prune more nodes from the search tree.

We have seen that it is useful to reduce $l_{\mathrm{k}+1,\mathrm{k}}$ by an IGT if the resulted $\bar{d}_{\mathrm{k}+1}$ satisfies $\bar{d}_{\mathrm{k}+1}<d_{\mathrm{k}+1}$, i.e., a permutation of the pair $(k,k+1)$ will be performed, because it can help to strive for order (5). However, reducing l_{ik} for $i>k+1$ will have no effect on **D** since equation (12) only involves $l_{\mathrm{k}+1,\mathrm{k}}$. Hence, even if $|l_{\mathrm{ik}}|$ is very large, reducing it will not improve the search process at all. This means that reducing all the off-diagonal entries of **L** is unnecessary. We only need to do a partial reduction.

Large off-diagonal entries in **L** indicate that the ambiguities are not decorrelated as much as possible, which contradicts the claim that it should be one of the objectives of the reduction process in the literature. In the following, we provide a three-dimensional example to illustrate the issue.

Example 3 Let the covariance matrix and the RLS estimate of the ambiguity vector **x** be

$$\mathbf{W}_{\hat{\mathbf{x}}}=\begin{bmatrix}2{\cdot}8355 & -0{\cdot}0271 & -0{\cdot}8071\\ -0{\cdot}0271 & 0{\cdot}7586 & 2{\cdot}0600\\ -0{\cdot}8071 & 2{\cdot}0600 & 5{\cdot}7842\end{bmatrix},\ \hat{\mathbf{x}}=\begin{bmatrix}26{\cdot}6917\\ 64{\cdot}1662\\ 42{\cdot}5485\end{bmatrix}$$

Then the correlation coefficients are

$$\rho_{12}=-0{\cdot}0185,\quad \rho_{13}=-0{\cdot}1993,\quad \rho_{23}=0{\cdot}9834 \qquad (36)$$

The integer points $[x_1,x_2,x_3]^{\mathrm{T}}$ encountered during the search are displayed in the first block column of Table 3 and the solution is. With the LAMBDA reduction or MLAMBDA reduction, the unimodular transformation matrix is

Table 2 Search results for Example 2

x_1	x_2	z_1	z_2	$\tilde{z}_1$	$\tilde{z}_2$
13	1	2	14	2	12
14	2	…	13	…	…
…	0	…	…		
…	…				

$$\mathbf{Z}=\begin{bmatrix}4 & -2 & 1\\ -43 & 19 & -11\\ 16 & -7 & 4\end{bmatrix}$$

The covariance matrix becomes

$$\mathbf{W}_{\hat{\mathbf{z}}}=\mathbf{Z}^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{Z}=\begin{bmatrix}0{\cdot}2282 & 0{\cdot}0452 & -0{\cdot}0009\\ 0{\cdot}0452 & 0{\cdot}1232 & 0{\cdot}0006\\ -0{\cdot}0009 & -0{\cdot}0006 & 0{\cdot}0327\end{bmatrix}$$

From this transformed covariance matrix, we obtain the correlation coefficients

$$\rho_{12}=-0{\cdot}2696,\quad \rho_{13}=-0{\cdot}0104,\quad \rho_{23}=0{\cdot}0095$$

which indicates that on average the transformed RLS estimates of ambiguities are less correlated than the original ones (see equation (36)) in terms of correlation coefficients. The integer points $[z_1,z_2,z_3]^{\mathrm{T}}$ encountered during the search are displayed in the second block column of Table 3 and the solution is

$$\breve{\mathbf{x}}=\mathbf{Z}^{-\mathrm{T}}[-1972,868,-509]^{\mathrm{T}}=[27,64,42]^{\mathrm{T}}$$

Now we do not apply IGTs to reduce l_{31} in the reduction process. With this partial reduction strategy, the unimodular transformation matrix is

$$\mathbf{Z}=\begin{bmatrix}0 & 0 & 1\\ 1 & -3 & -11\\ 0 & 1 & 4\end{bmatrix}$$

The covariance matrix and the RLS estimate become

$$\mathbf{W}_{\hat{\mathbf{z}}}=\mathbf{Z}^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{Z}=\begin{bmatrix}0{\cdot}7587 & -0{\cdot}2158 & -0{\cdot}1317\\ -0{\cdot}2158 & 0{\cdot}2516 & 0{\cdot}0648\\ -0{\cdot}1317 & 0{\cdot}0648 & 0{\cdot}0327\end{bmatrix}$$

From this transformed covariance matrix, we obtain the correlation coefficients

$$\rho_{12}=-0{\cdot}4940,\quad \rho_{13}=-0{\cdot}8362,\quad \rho_{23}=0{\cdot}7144$$

which indicates that on average the transformed RLS estimates of ambiguities are more correlated than the original ones (see equation (36)) in terms of correlation coefficients. The integer points $[\tilde{z}_1,\tilde{z}_2,\tilde{z}_3]^{\mathrm{T}}$ encountered during the search are displayed in the third block column of Table 3 and the solution is

$$\breve{\mathbf{x}}=\mathbf{Z}^{-\mathrm{T}}[64,-150,-509]^{\mathrm{T}}=[27,64,42]^{\mathrm{T}}$$

From Table 3, we see that the partial reduction is as effective as the full reduction given in LAMBDA and MLAMBDA and both make the search process more efficient, although they increase and decrease the correlation coefficients in magnitude on average, respectively. This indicates that reducing the correlation coefficients of

Table 3 Search results for Example 3

x_1	x_2	x_3	z_1	z_2	z_3	$\tilde{z}_1$	$\tilde{z}_2$	$\tilde{z}_3$
23	64	43	−1972	868	−509	64	−150	−509
27	64	42	…	…	…	…	…	…
…	…	44						
…	…	41						
…	…	…						

RLS estimates of ambiguities should not be an objective of the reduction process.

The significance of the result we obtained in this section is threefold:

- It indicates that contrary to many people's belief, the computational cost of the search is independent of the off-diagonal entries of **L** and of the correlation between the ambiguities.
- It provides a different explanation on the role of lower triangular IGTs in the reduction process.
- It leads to a more efficient reduction algorithm, see later.

On the condition number of the covariance matrix

In some GNSS literature (see, e.g. [**13**], [**14**], [**25**] and [**26**]) it is believed that the objective of reduction process is to reduce the condition number of the covariance matrix $\mathbf{W}_{\hat{\mathbf{x}}}$. The 2-norm condition number of $\mathbf{W}_{\hat{\mathbf{x}}}$ satisfies

$$\kappa_2(\mathbf{W}_{\hat{\mathbf{x}}}) \equiv \left\|\mathbf{W}_{\hat{\mathbf{x}}}\right\|_2 \left\|\mathbf{W}_{\hat{\mathbf{x}}}^{-1}\right\|_2 = \frac{\lambda_{\max}\left(\mathbf{W}_{\hat{\mathbf{x}}}\right)}{\lambda_{\min}\left(\mathbf{W}_{\hat{\mathbf{x}}}\right)}$$

where $\lambda_{\max}(\mathbf{W}_{\hat{\mathbf{x}}})$ and $\lambda_{\min}(\mathbf{W}_{\hat{\mathbf{x}}})$ are the largest and smallest eigenvalues of $\mathbf{W}_{\hat{\mathbf{x}}}$, respectively. Suppose that the search region is a hyper-ellipsoid

$$(\mathbf{x}-\hat{\mathbf{x}})^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}^{-1}(\mathbf{x}-\hat{\mathbf{x}}) < \chi^2$$

Then, $\kappa_2(\mathbf{W}_{\hat{\mathbf{x}}})$ is equal to the square of the ratio of the major and minor axes of the search ellipsoid (see, e.g. [16, sec. 6·4]). In other words, the condition number measures the elongation of the search ellipsoid. It is true that often the current decorrelation strategies in the literature can reduce the condition number or the elongation of the search ellipsoid and make the search process faster. But should reducing the condition number or the elongation be an objective of a reduction process? In this section, we will argue that it should not.

Reducing the off-diagonal entries of the L-factor can usually reduce the condition number of **L** and thus the condition number of the covariance matrix $\mathbf{W}_{\hat{\mathbf{x}}}$. But we have shown that the off-diagonal entries of **L** (except subdiagonal entries) do not affect the search speed and do not need to be reduced in theory. Thus the condition number of the covariance matrix is not a good measure of effectiveness of the reduction process. The following example shows that although the condition number of the covariance matrix can significantly be reduced, the search speed is not changed.

Example 4 Let

$$\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L} = \begin{bmatrix} 1 & 1000{\cdot}5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 0{\cdot}05 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1000{\cdot}5 & 1 \end{bmatrix}$$

Then $\kappa_2(\mathbf{W}_{\hat{\mathbf{x}}}) = 1{\cdot}2527 \times 10^{10}$. After we apply a lower triangular IGT to reduce the (2,1) entry of **L**, the new covariance matrix becomes

$$\mathbf{W}_{\hat{\mathbf{z}}} = \mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L} = \begin{bmatrix} 1 & 0{\cdot}5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 0{\cdot}05 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0{\cdot}5 & 1 \end{bmatrix}$$

Then, $\kappa_2(\mathbf{W}_{\hat{\mathbf{z}}}) = 80{\cdot}5071$, much smaller than the original one. But the search speed will not be changed by the reduction according to Theorem 1.

The condition number of the covariance matrix does not change under permutations. But permutations can change the search speed significantly. This shows again that the condition number is not a good indication of the search speed. We use an example to illustrate this.

Example 5 Let $\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}} = \mathrm{diag}(1,4,16)$, where $\mathbf{L}=\mathbf{I}$ and $\mathbf{D}=\mathbf{W}_{\hat{\mathbf{x}}}$, and let $\hat{\mathbf{x}} = [0{\cdot}4, 0{\cdot}8, 1{\cdot}6]^{\mathrm{T}}$. The search results are given in the first block column of Table 4 and the ILS solution is $\hat{\mathbf{x}} = [0,1,2]^{\mathrm{T}}$. Suppose we permute $\mathbf{W}_{\hat{\mathbf{x}}}$ such that $\mathbf{W}_{\hat{\mathbf{z}}} = \mathbf{P}^{\mathrm{T}}\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{P} = \mathrm{diag}(16,4,1)$. Then $\mathbf{W}_{\hat{\mathbf{z}}} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$, where $\mathbf{L}=\mathbf{I}$ and $\mathbf{D}=\mathbf{W}_{\hat{\mathbf{z}}}$. The search results for the permuted one are given in the second block column of Table 4 and the solution $\hat{\mathbf{x}} = \boldsymbol{P}[2,1,0]^{\mathrm{T}} = [0,1,2]$. Although $\kappa_2(\mathbf{W}_{\hat{\mathbf{x}}})\kappa_2(\mathbf{W}_{\hat{\mathbf{z}}})$, the search speed for the permuted problem is much more efficient than that for the original one. Note that the order of diagonal entries of the **D**-factor of $\mathbf{W}_{\hat{\mathbf{z}}}$ is exactly what a reduction process should pursue, see objective (ii) in the section on 'Reduction and search'. In the section on 'Numerical experiments', we will use more general examples to show that proper permutations of the covariance matrix without doing any decorrelation can improve the search efficiency.

A new reduction algorithm

In this section we would like to propose a new reduction algorithm which is more efficient and numerically stable than the reduction algorithms used in LAMBDA and MLAMBDA. We showed before that except the subdiagonal entries of **L**, the other lower triangular entries do not need to be reduced in theory. Thus we could design an algorithm which does not reduce the entries of **L** below the subdiagonal entries. However, it is possible that some of those entries may become too big after a sequence of size reduction for the subdiagonal entries. This may cause a numerical stability problem. For the sake of numerical stability, therefore, after we reduce $l_{k+1,k}$, we also reduce the entries $l_{k+2,k},\ldots,l_{n,k}$ by IGTs. If we do not reduce the former, we do not reduce the latter either. Thus the first property of the LLL reduction given in inequality (6) may not hold for all $i>j$, while the second property holds for all j. Owing to this, our new reduction algorithm will be referred to as PReduction (where P stands for 'partial').

Now we describe the key steps of PReduction. Like the reduction algorithm used in MLAMBDA (called MReduction), for computational efficiency, we first compute the $\mathbf{L}^{\mathrm{T}}\mathbf{D}\mathbf{L}$ factorisation with minimum symmetric pivoting (see Algorithm 3·1 in [**4**]). This permutation sorts the conditional variances to pursue the objective (5). Then the algorithm works with columns of **L** and **D** from the right to the left. At column k, it computes the new value of $l_{k+,k}$ as if an IGT were applied to reduce $l_{k+1,k}$. Using this new value we compute $\bar{d}_{k+1}$ (see equation (12)). If $\bar{d}_{k+1} \geq d_{k+1}$ holds, then permuting pair $(k,k+1)$ will not help strive for order (5),

Table 4 Search results for Example 5

x_1	x_2	x_3	z_1	z_2	z_3
0	1	2	2	1	0
…	0	2	…	…	…
…	1	1			
…	1	3			
…	1	0			
…	…	…			

therefore the algorithm moves to column $k-1$ without actually applying any IGT; otherwise it first applies IGTs to make $|l_{ik}| \leq 1/2$ for $i=k+1,\ldots,n$, then it permutes pair $(k,k+1)$ and moves back to column $k+1$. In the reduction algorithm used in LAMBDA (to be referred to as LAMBDA Reduction) when a permutation occurs at column k, the algorithm goes back to the initial position $k=n-1$.

We now present the complete new reduction algorithm:

Algorithm 3. (PReduction). Given the covariance matrix $\mathbf{W}_{\hat{\mathbf{x}}}$ and real-valued LS estimate $\hat{\mathbf{x}}$, this algorithm computes a unimodular matrix $\mathbf{Z}$ and the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation $\mathbf{W}_{\hat{\mathbf{z}}} = \mathbf{Z}^T\mathbf{W}_{\hat{\mathbf{z}}}\mathbf{Z} = \mathbf{L}^T\mathbf{D}\mathbf{L}$, which is obtained from the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{z}}}$ by updating. This algorithm also computes $\hat{\mathbf{z}} = \mathbf{Z}^T\hat{\mathbf{x}}$, which overwrites $\hat{\mathbf{x}}$.

function: $[\mathbf{Z},\mathbf{L},\mathbf{D},\hat{\mathbf{x}}] = \text{PReduction}(\mathbf{W}_{\hat{\mathbf{x}}},\hat{\mathbf{x}})$

Compute the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation of $\mathbf{W}_{\hat{\mathbf{x}}}$ with symmetric pivoting

$\mathbf{P}^T\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{P} = \mathbf{L}^T\mathbf{D}\mathbf{L}$
  $\hat{\mathbf{x}} = \mathbf{P}^T\hat{\mathbf{x}}$
  $\mathbf{Z} = \mathbf{P}$
  $k = n-1$
  while $k>0$
    $l = l_{k+1,k} - \lfloor l_{k+1,k} \rceil l_{k+1,k,k+1}$
    $\bar{d}_{k+1} = d_k + l^2 d_{k+1}$
    if $\bar{d}_{k+1} = d_{k+1}$
      if $|l_{k+1,k}| > 1/2$
        for $i = k+1{:}n$
          //See Alg. 1
          $[\mathbf{L},\hat{\mathbf{x}},\mathbf{Z}] = \text{Gauss}(\mathbf{L},i,k,\hat{\mathbf{x}},\mathbf{Z})$
        end
      end
      //See Alg. 2
      $[\mathbf{L},\mathbf{D},\hat{\mathbf{x}},\mathbf{Z}] = \text{Permute}(\mathbf{L},\mathbf{D},k,\bar{d}_{k+1},\hat{\mathbf{x}},\mathbf{Z})$
      if $k<n-1$
        $k = k+1$
      end
    else
      $k = k-1$
    end
  end

The structure of PReduction is similar to that of LAMBDA Reduction. But the former can be much more efficient as it uses the symmetric pivoting strategy in computing the $\mathbf{L}^T\mathbf{D}\mathbf{L}$ factorisation and does less size reductions.

PReduction is more numerically stable than MReduction, because, unlike the latter, the former does size reduction for the remaining entries of the column in $\mathbf{L}$ immediately after it does size reduction for a subdiagonal entry of $\mathbf{L}$, avoiding quick growth of the off-diagonal entries of $\mathbf{L}$ in the reduction process.

The main difference between PReduction and LAMBDA Reduction is that the latter does more size reduction for the entries below the subdiagonal entries of $\mathbf{L}$. Therefore, it is very likely that there is no big difference between the diagonal of $\mathbf{D}$ obtained by the two reduction algorithms. The structure of MReduction is not quite similar to that of LAMBDA Reduction. But both do the LLL reduction, so the difference between the diagonal of $\mathbf{D}$ obtained by the two algorithms is not expected to be large. Thus these three reduction algorithms should usually have more of less the same effect on the efficiency of the search process. This is confirmed in our numerical tests.

Numerical experiments

To compare Algorithm PReduction given in the previous section with the LAMBDA reduction algorithm and the MLAMBDA reduction algorithm (MReduction) in terms of computational efficiency and numerical stability, in this section, we give some numerical test results. We will also give test results to show the three reduction algorithms have almost the same effect on the efficiency of the search process. The routine for the LAMBDA reduction algorithm is from the LAMBDA package available from TU Delft web site. MLAMBDA's search routine was used for the search process.

We use the CPU running time as a measure of computational efficiency and the relative backward error as a measure of numerical stability. For the concepts of backward error and numerical stability, see [7]. Given $\mathbf{W}_{\hat{\mathbf{x}}}$, a reduction algorithm computes the factorisation

$$\mathbf{Z}^T\mathbf{W}_{\hat{\mathbf{x}}}\mathbf{Z} = \mathbf{L}^T\mathbf{D}\mathbf{L}.$$

The relative backward error is

$$\text{RBE} = \frac{\left\|\mathbf{W}_{\hat{\mathbf{x}}} - \mathbf{Z}_c^{-T}\mathbf{L}_c^T\mathbf{D}_c\mathbf{L}_c\mathbf{Z}_c^{-1}\right\|_2}{\left\|\mathbf{W}_{\hat{\mathbf{x}}}\right\|_2}$$

where $\mathbf{Z}_c$, $\mathbf{L}_c$ and $\mathbf{D}_c$ are the computed versions of $\mathbf{Z}$, $\mathbf{L}$ and $\mathbf{D}$, respectively. In our computations when we computed $\mathbf{Z}_c^{-1}$ we used the fact that $\mathbf{Z}_{ij}^{-1} = \mathbf{I} + \mu\mathbf{e}_i\mathbf{e}_j^T$ (cf. (8)) and $\mathbf{P}_{k,k+1}^{-1} = \mathbf{P}_{k,k+1}$ (cf. equation (10)).

All our computations were performed in Matlab 7·11 on a Linux 2·6, 1·86 GHz machine.

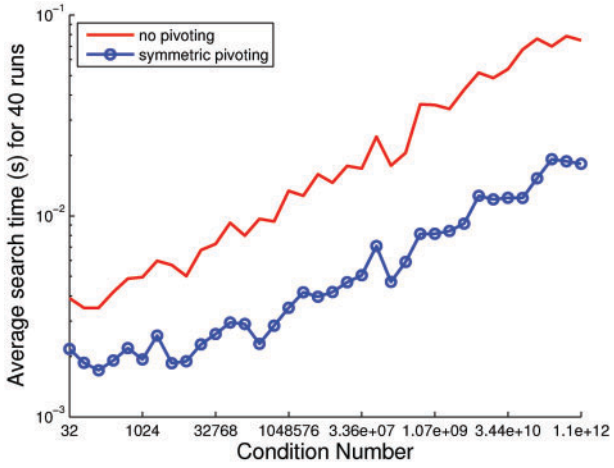
Setup and simulation results

We performed simulations for four different cases (Cases 1, 2 and 3 were used in [4]).

- Case 1: $\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{U}\mathbf{D}\mathbf{U}^T$, $\mathbf{U}$ is a random orthogonal matrix obtained by the QR factorisation of a random matrix generated by randn(n,n), $\mathbf{D}=\text{diag}(d_i)$, where $d_1 = 2^{-k/2}$, $d_n = 2^{k/2}$, other diagonal elements of $\mathbf{D}$ are randomly distributed between d_1 and d_n. We took $n=20$ and $k=5, 6,\ldots, n$. The range of the condition number $\kappa_2(\mathbf{W}_{\hat{\mathbf{x}}})$ is from 2^5 to 2^{20}.
- Case 2: $\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}^T\mathbf{D}\mathbf{L}$, where $\mathbf{L}$ is a unit lower triangular matrix with each l_{ij} (for $i>j$) being a pseudorandom number drawn from the standard normal distribution and generated by the Matlab command randn, $\mathbf{D}=\text{diag}(d_i)$ with each d_i being a pseudorandom number drawn from the standard uniform distribution on the open interval (0,1) and generated by Matlab command rand, and $\hat{\boldsymbol{x}} = 100^*\text{randn}(n,1)$.
- Case 3: $\mathbf{W}_{\hat{\mathbf{x}}} = \mathbf{L}^T\mathbf{D}\mathbf{L}$, where $\mathbf{L}$ is generated in the same way as in Case 1

  $$\mathbf{D} = \text{diag}(200,200,200,0{\cdot}1,0{\cdot}1,\cdots,0{\cdot}1)$$

  and $\hat{\boldsymbol{x}}$ is generated in the same way as in Case 1.
- Case 4: We constructed the linear model $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v}$, where $\mathbf{A} = \text{randn}(n,\ n)$, $\mathbf{x} = [100^*\text{randn}(n,1)]$ and $\mathbf{v} = 0{\cdot}01^{1/2} * \text{randn}(n,1)$. The problem we intend to solve is the ILS problem in the standard form: $\min_{\mathbf{x}\in\mathbb{Z}^n}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$. To solve it, we transformed it into



3 The effect of permutations

the form of (1), see the second paragraph of the section on 'Reduction and search'.

- Case 5: A set of real GPS data. It has 50 instances and the dimension of the integer ambiguity vector in each instance is 18. The data was provided to us by Dr Yang Gao of The University of Calgary.

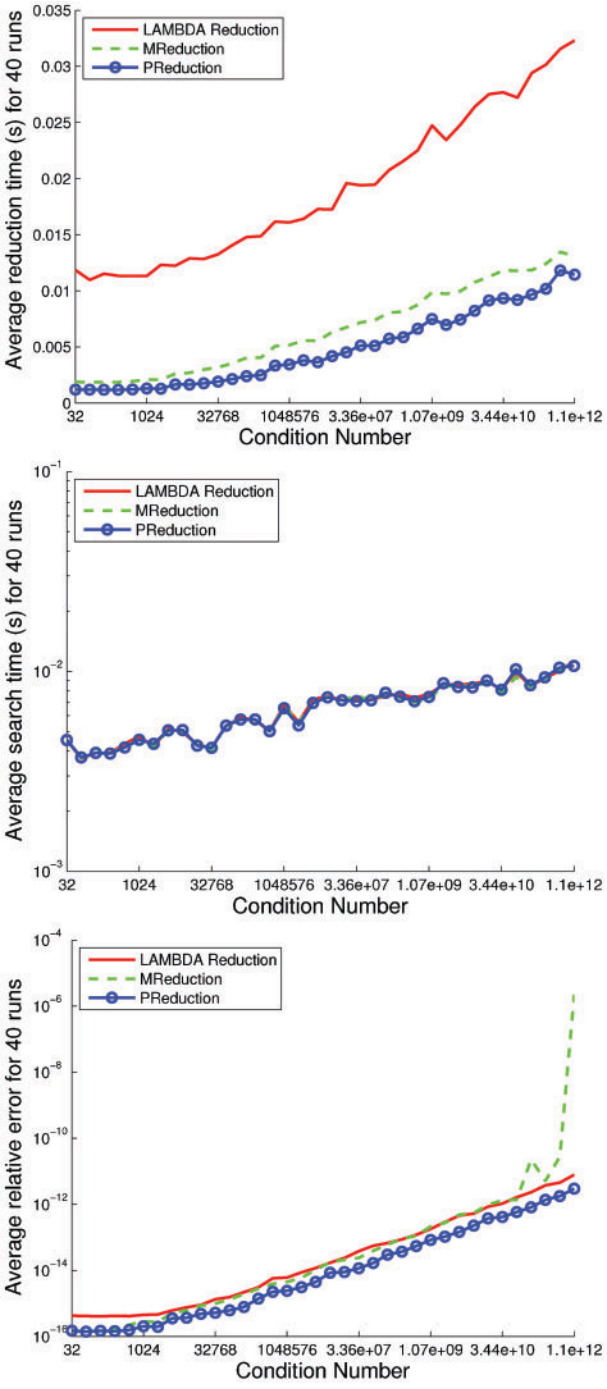
Effect of permutations on the search speed

In Example 5, we showed that a proper permutation of the covariance matrix can improve the search speed. Following a suggestion from a reviewer, here we use Case 1 to show that the minimum symmetric pivoting strategy incorporated in the $\mathbf{L}^T\mathbf{DL}$ factorisation can improve the search efficiency. The purpose is to show that correlation coefficients or condition numbers should not be used as measures of effectiveness of reduction. In our numerical tests, we first computed the $\mathbf{L}^T\mathbf{DL}$ factorisation without and with minimum symmetric pivoting, respectively, and then applied the search routine to find the ILS solution. So no decorrelation or size reduction was involved. In our test, we took dimensions $n=5$, 6,…, 40 and performed 40 runs for each n. The test result was given in Fig. 3, which indicates that the search efficiency can increase up to about five times by using this permutation strategy.

Comparison of the reduction algorithms

We use Cases 1–5 to give comparisons of the three different reduction algorithms. For Cases 1, 2 and 4, we took dimensions $n=5$, 6,…, 40 and performed 40 runs for each n. For Case 3, we performed 40 runs for each k. For Case 5, we performed the tests on 50 different instances. For each case we give three plots, corresponding to the average reduction time (s), the average search time (s), and the average relative backward error; see Figs. 4–8.

From the simulation results for Cases 1–4, we observe that new reduction algorithm PReduction is faster than both LAMBDA Reduction and MReduction for all cases. Usually, the improvement becomes more significant when the dimension n increases. For example, in Case 3, PReduction has about the same running time as MReduction and LAMBDA reduction when $n=5$, but is almost 10 times faster when $n=40$. In Case 1, even when $n=5$, PReduction is slightly faster than MReduction and more than 6 times faster than LAMBDA's reduction algorithm. Except in Case 4 for $n=33,35$, we also

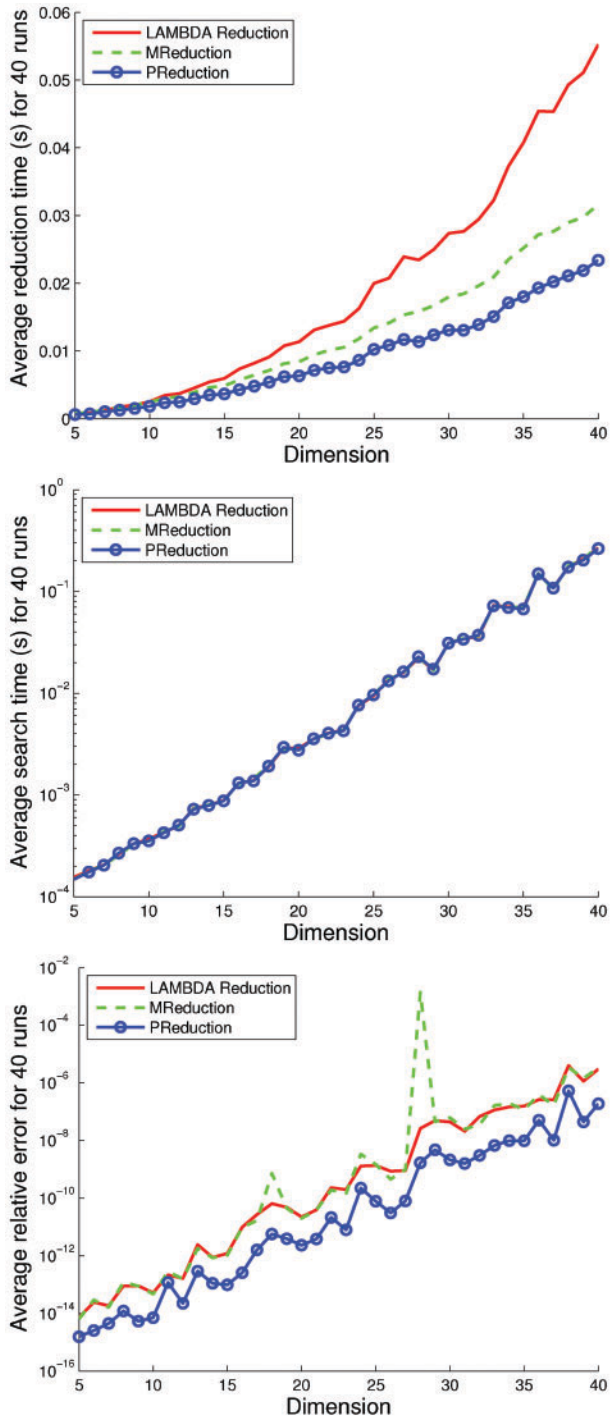


4 Case 1

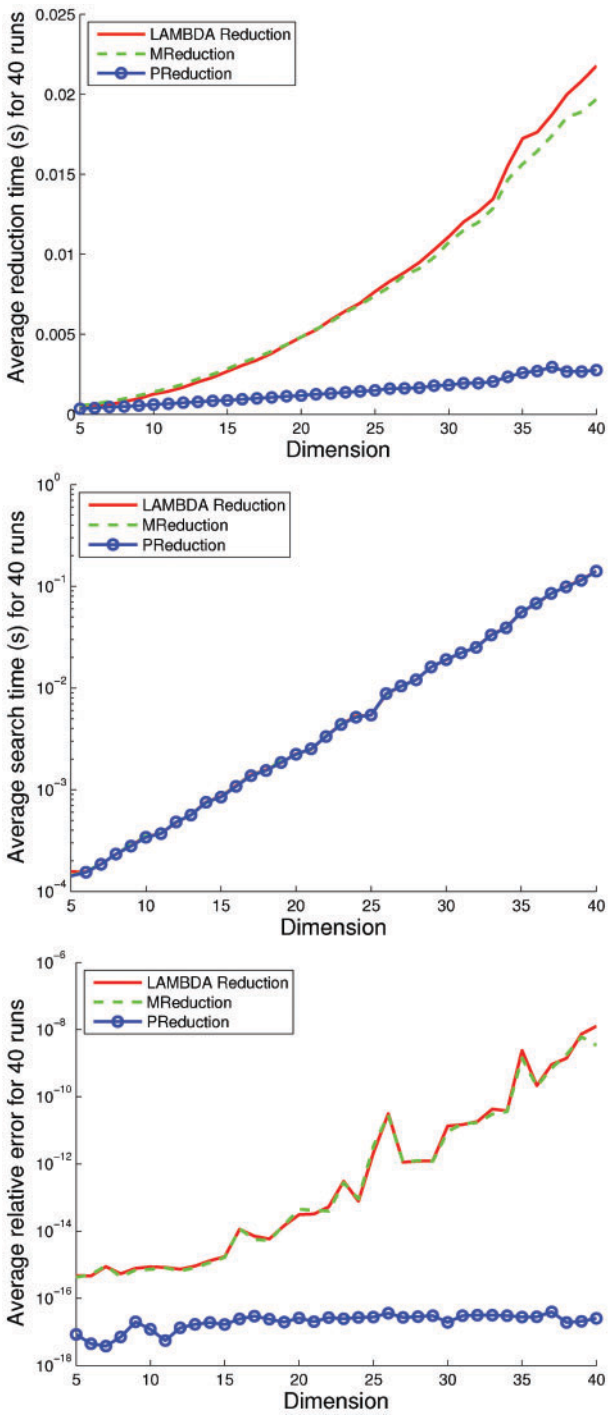
observe that the three different reduction algorithms have almost the same effect on the efficiency of the search process (the three curves for the search time in the plots often coincide). This confirms what we argued at the end of the previous section. For the exceptional situations we will give an explanation later.

In Case 5, we observe that PReduction is about 20% faster then MLAMBDA and is about four times faster than LAMBDA's reduction. Consider that the reduction time is comparable to the search time in this case, the increase of the reduction efficiency is very helpful to improve the overall efficiency.

From the simulation results, we also observe that PReduction is usually more numerically stable than MReduction and LAMBDA Reduction (to less extent



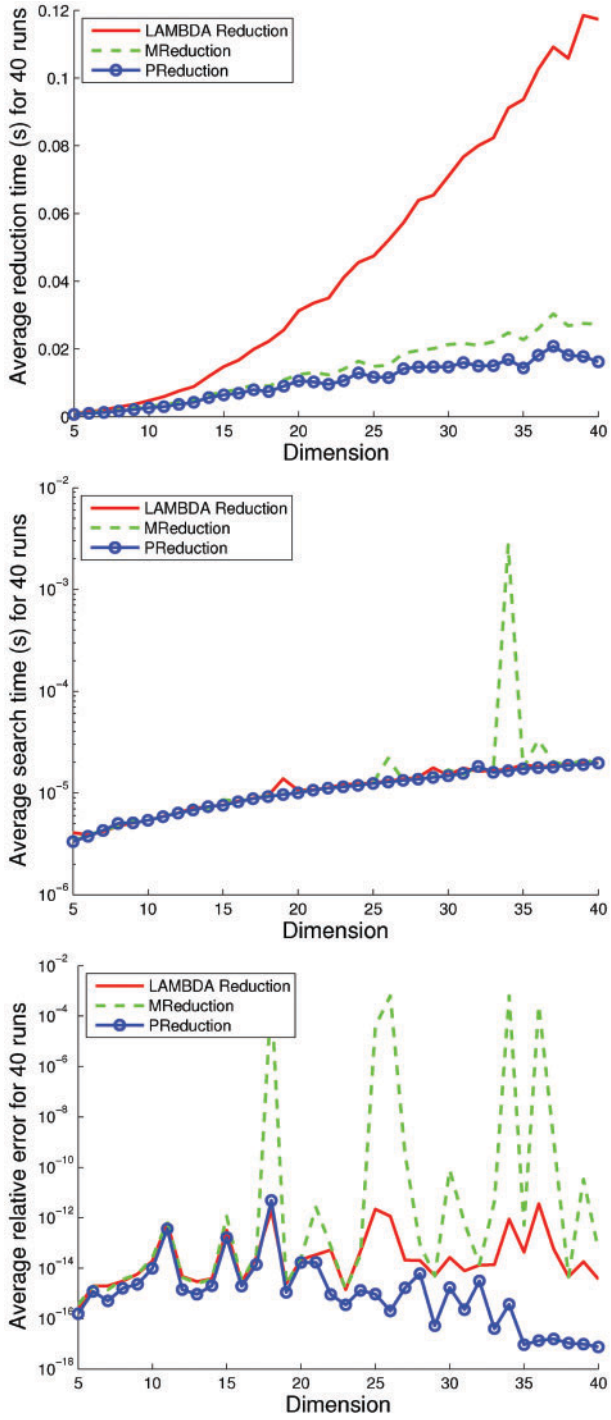
5 Case 2



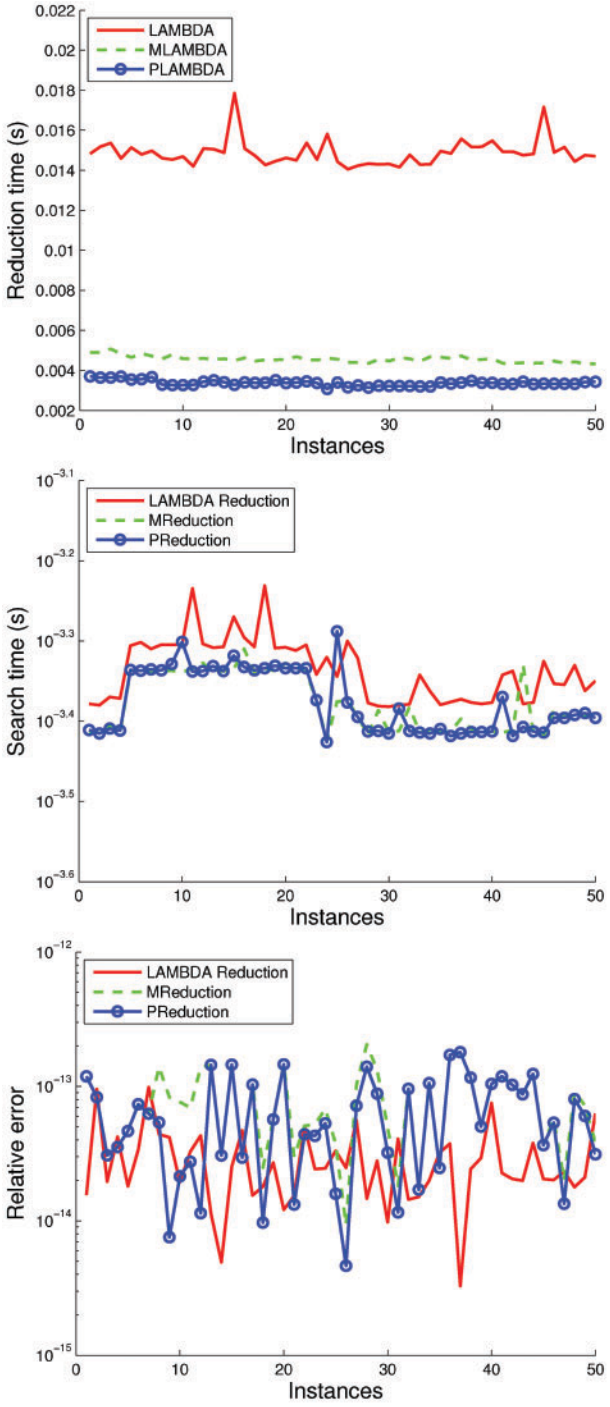
6 Case 3

and occasions). From the third plot of Fig. 7, we see that MReduction can be much worse than PReduction and LAMBDA Reduction in terms of stability. The reason is that IGTs for reducing the entries below the subdiagonal of **L** in MReduction were deferred to the last step to make the algorithm more efficient, but this can result in large off-diagonal entries in **L** and consequently large entries in **Z**, which may lead to big rounding errors. As we said in the previous section, such a problem is avoided in PReduction. The reason that PReduction can be much more stable than LAMBDA Reduction (see the third plot of Fig. 6) is probably that the former involves less computations. For the real data (see Fig. 8), the three algorithms' stability is more or less the same.

Now we can explain the spikes in the second plot of Fig. 7. Owing to the numerical stability problem, the transformed ILS problem after MReduction is applied has large rounding errors in each of the instances which produce two large spikes and the resulted ILS solution is not identical to the ILS solution obtained by using PReduction or LAMBDA Reduction in one of the 40 runs. Here we would like to point out that the MILES package [**5**], which solves a mixed ILS problem in the standard form, uses a reduction algorithm which is different from MReduction and has no such numerical stability problem.



7 Case 4



8 Case 5

Summary

We have shown that there are two misconceptions about the reduction or decorrelation process in solving ILS problems by discrete search in the GNSS literature. The first is that the reduction process should decorrelate the ambiguities as much as possible. The second misconception is that the reduction process should reduce the condition number of the covariance matrix. We showed by theoretical arguments and numerical examples that both are not right objectives a reduction process should try to achieve. The right objective is to pursue the order of the diagonal entries of **D** given in order (5). The new understanding on the role of decorrelation in the reduction process led to PReduction, a reduction algorithm. The numerical test results indicate that PReduction is more computationally efficient than LAMBDA's reduction algorithm and MLAMBDA's reduction algorithm (to less extent) and is usually more numerically stable than MLAMBDA's reduction algorithm and LAMBDA's reduction algorithm (to less extent). For real GPS data, we found that the three reduction algorithms have more or less the same stability. The new version of the Matlab package MLAMBDA has used the new reduction algorithm presented in this paper. The package can be downloaded from http://www.cs.mcgill.ca/~chang/software.php.

Acknowledgements

This research was supported by NSERC of Canada Grant no. RGPIN217191-12. Part of the ideas given in sections on 'Impact of igts on the search process' and 'A new reduction algorithm' was given in [**24**]. We would like to thank Dr Yang Gao and Mr Junbo Shi from University of Calgary and Mr Xiankun Wang from The Ohio State University for providing real GPS data for our numerical tests. And we also would like to thank the two anonymous reviewers for providing valuable suggestions to improve this paper.

References

1. Agrell, E., Eriksson, T., Vardy, A. and Zeger, K., 2002. Closest point search in lattices. IEEE Transactions *on Information Theory*, 48(8): 2201–2214.
2. Babai, L.,1986. On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica, 6(1): 1–13.
3. Chang, X. W., Wen, J. and Xie, X., 2012. Effects of the LLL reduction on the success probability of the Babai point and on the complexity of sphere decoding. http://arxiv.org/abs/1204·2009.
4. Chang, X. W., Yang, X. and Zhou, T., 2005. MLAMBDA: a modified LAMBDA method for integer least-squares estimation. *Journal of Geodesy*, 79(9): 552–565.
5. Chang, X. W. and Zhou, T.,2007, MILES: Matlab package for solving Mixed Integer LEast Squares problems. GPS Solutions, 11(4): 289–294.
6. de Jonge, P. and Tiberius, C. C. J. M., 1996. LAMBDA method for integer ambiguity estimation: implementation aspects. In: Delft Geodetic Computing Center LGR-Series, No.12.
7. Higham, N. J., 2002. *Accuracy and Stability of Numerical Algorithms*, Philadelphia, PA, SIAM, 2nd edn.
8. Jazaeri, S., Amiri-Simkooei, A. R. and Sharifi, M. A., 2012. Fast integer least-squares estimation for GNSS high-dimensional ambiguity resolution using lattice theory. *Journal of Geodesy*, 86(2): 123–136.
9. Joosten, P. and Tiberius, C. C. J. M., 2002. LAMBDA: FAQs. *GPS Solutions*, 6(1): 109–114.
10. Lenstra, A. K., Lenstra, H. W. and Lovász, L., 1982. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4): 515–534.
11. Li, B. and Teunissen, P. J. G., 2011. High dimensional integer ambiguity resolution: a first comparison between LAMBDA and Bernese. *Journal of Navigation*, 64(S1): S192–S210.
12. Ling, C. and Howgrave-Graham, N., 2007. Effective LLL reduction for lattice decoding. *Proceedings of IEEE International Symposium on Information Theory*, 24–29 June, Nice, France: 196–200.
13. Liu, L.T., Hsu, H. T., Zhu, Y. Z. and Ou, J. K.,1999. A new approach to GPS ambiguity decorrelation. Journal of Geodesy, 73(9): 478–490.
14. Lou, L. and Grafarend, E. W., 2003. GPS integer ambiguity resolution by various decorrelation methods. *Zeitschrift fur Vermessungswesen* 128(3): 203–210.
15. Schnorr, C. P. and Euchner, M., 1994. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1): 181–199.
16. Strang, G. and Borre, K.,1997. Linear algebra, geodesy, and GPS. Wellesley, MA, Wellesley Cambridge Press.
17. Teunissen, P. J. G., 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.
18. Teunissen, P. J. G.,1995. The invertible GPS ambiguity transformation. Manuscripta Geodaetica, 20(6): 489–497.
19. Teunissen, P. J. G., 1995. The least-squares ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation. *Journal of Geodesy*, 70(1): 65–82.
20. Teunissen, P. J. G., 1997. A canonical theory for short GPS baselines. Part III: the geometry of the ambiguity search space. *Journal of Geodesy*, 71(8): 486–501.
21. Teunissen, P. J. G., 1998. GPS carrier phase ambiguity fixing concepts. In: GPS for geodesy, New York, Springer-Verlag, Teunissen, P. and Kleusberg, A. (eds), 2nd edn: 319–388.
22. Teunissen, P. J. G., 1999. An optimality property of the integer least-squares estimate. *Journal of Geodesy*, 73(11): 587–593.
23. Teunissen, P. J. G., 2001. GNSS ambiguity bootstrapping: theory and applications. *Proceedings of International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation*, 5–8 June, Banff, Alberta, Canada: 246–254.
24. Xie, X., Chang, X. W. and Al Borno, M., 2011. Partial LLL reduction. *Proceedings of IEEE GLOBECOM 2011*, 5–9 December, Houston, TX, USA.
25. Xu, P., 2001. Random simulation and GPS decorrelation. *Journal of Geodesy*, 75(7): 408–423
26. Xu, P., 2011. Parallel Cholesky-based reduction for the weighted integer least squares problem. *Journal of Geodesy*, 86(1): 35–52.