GPS Solut
DOI 10.1007/s10291-007-0063-y

GPS TOOL BOX

MILES: MATLAB package for solving Mixed Integer LEast Squares problems

Xiao-Wen Chang · Tianyang Zhou

Received: 24 November 2006 / Accepted: 30 March 2007

Abstract In GNSS, for fixing integer ambiguities and estimating positions, a mixed integer least squares problem has to be solved. The MATLAB package MILES provides fast and numerically reliable routines to solve this problem. In the process of solving a mixed integer least squares problem, an ordinary integer least squares problem is solved. Thus this package can also be used to solve an ordinary integer least squares problem alone. An option to compute multiple solutions is provided. This paper gives a description of this package and provides a guide for using it.

Keywords MATLAB package · Mixed integer least squares estimation · GNSS · Integer ambiguities · Positioning

Introduction

MILES is a free MATLAB package for solving Mixed Integer LEast Squares problems (including ordinary integer least squares problems), and it can be used for integer ambiguity determination and position estimation in GNSS. The purpose of this paper is to introduce this package and provide a guide for using it.

First let us introduce some notation. Let the sets of all real and integer $m \times n$ matrices be denoted by $\mathbb{R}^{m\times n}$ and $\mathbb{Z}^{m\times n}$, respectively, and the sets of real and integer n-vectors by $\mathbb{R}^n$ and $\mathbb{Z}^n$, respectively. Let $\|\cdot\|$ denote the 2-norm of a vector, i.e., if $\boldsymbol{a} = (a_i) \in \mathbb{R}^n$, then $\|\boldsymbol{a}\| = \sqrt{\sum_{i=1}^n a_i^2}$.

Given $\boldsymbol{A} \in \mathbb{R}^{m\times k}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{y} \in \mathbb{R}^m$, suppose that $[\boldsymbol{A},\boldsymbol{B}]$ has full column rank. This MATLAB package provides a function to produce p optimal solutions to the mixed integer least squares (MILS) problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^k,\, \boldsymbol{z}\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z}\|^2, \tag{1}$$

in the sense that a pair $\{\boldsymbol{x}^{(j)}, \boldsymbol{z}^{(j)}\} \in \mathbb{R}^k \times \mathbb{Z}^n$ is the jth optimal solution if its corresponding residual norm $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(j)} - \boldsymbol{B}\boldsymbol{z}^{(j)}\|$ is the jth smallest (note that some of these p residual norms can be equal), i.e.,

$$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(1)} - \boldsymbol{B}\boldsymbol{z}^{(1)}\| \le \cdots \le \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(j)} - \boldsymbol{B}\boldsymbol{z}^{(j)}\| \le \cdots \le \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(p)} - \boldsymbol{B}\boldsymbol{z}^{(p)}\|.$$

Here p is a parameter to be provided by a user and its default value is 1. The MILS problem (1) arises in some aplications, such as GPS positioning using carrier phase measurements or both carrier phase and code measurements, see, e.g., Chang et al. (2004).

If the matrix $\boldsymbol{A}$ is nonexistent, (1) becomes an ordinary integer least squares (ILS) problem:

$$\min_{\boldsymbol{z}\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{z}\|^2. \tag{2}$$

This package also provides a function to produce p optimal solutions to (2).

The GPS Tool Box is a column dedicated to highlighting algorithms and source code utilized by GPS engineers and scientists. If you have an interesting program or software package you would like to share with our readers, please pass it along; e-mail it to us at gps-toolbox@ngs.noaa.gov. To comment on any of the source code discussed here, or to download source code, visit our website at http://www.ngs.noaa.gov/gps-toolbox. This column is edited by Stephen Hilla, National Geodetic Survey, NOAA, Silver Spring, Maryland, and Mike Craymer, Geodetic Survey Division, Natural Resources Canada, Ottawa, Ontario, Canada.

X.-W. Chang (✉) · T. Zhou
School of Computer Science, McGill University,
3480 University St, Montreal, QC, Canada H3A 2A7
e-mail: chang@cs.mcgill.ca

Some MILS problems or ordinary ILS problems in other forms can be transformed to (1) or (2), and then can be solved by this package. For example, suppose that one wants to solve

$$\min_{x\in\mathbb{R}^k,\, z\in\mathbb{Z}^n} (\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{B}\boldsymbol{z})^T\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{B}\boldsymbol{z}), \tag{3}$$

where $\boldsymbol{V}\in\mathbb{R}^{m\times m}$ is symmetric positive definite. One can first compute the Cholesky factorization $\boldsymbol{V}=\boldsymbol{R}^T\boldsymbol{R}$, where $\boldsymbol{R}$ is nonsingular upper triangular, by MATLAB's built-in function `chol`, then solve three lower triangular linear systems $\boldsymbol{R}^T\bar{\boldsymbol{y}}=\boldsymbol{y}, \boldsymbol{R}^T\bar{\boldsymbol{A}}=\boldsymbol{A}$ and $\boldsymbol{R}^T\bar{\boldsymbol{B}}=\boldsymbol{B}$ to obtain $\bar{\boldsymbol{y}},\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$, respectively, by MATLAB's backslash command \. Thus (3) is transformed to (1), where $\boldsymbol{y}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are replaced by $\bar{\boldsymbol{y}},\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$, respectively.

The well known LAMBDA package (see De Jonge and Tiberius 1996) solves the following ILS problem:

$$\min_{z\in\mathbb{Z}^n}(\hat{z}-z)^T\boldsymbol{V}^{-1}(\hat{z}-z), \tag{4}$$

where $\hat{z}\in\mathbb{R}^n$ is a given vector and $\boldsymbol{V}\in\mathbb{R}^{n\times n}$ is symmetric positive definite. Note that (4) can be regarded as a special case of (3) and thus can be solved by the MILES package. For GNSS applications, if observation equations are available and a user wants to fix the integer ambiguities and estimate positions, it is more straightforward to use our MILES package. The method for solving the ordinary ILS problem (2) implemented by the MILES package is a modification of the MLAMBDA method presented in Chang et al. (2005), which can be much faster than the LAMBDA method implemented by the LAMBDA package (MATLAB, version 2.0).

Outlines of algorithms

In this section, we give outlines of the algorithms for solving the MILS problem (1) and the ILS problem (2). For detailed description, see Chang and Zhou (2006a). Efficiency and reliability are the two key factors we considered in designing the algorithms.

- Solving the MILS problem (1)

To solve (1), we transform it to an ILS problem and a real upper triangular linear system of equations. By solving these two sub-problems sequentially, we will obtain the MILS solution.

Suppose $\boldsymbol{A}$ has the QR factorization

$$\boldsymbol{A}=[\boldsymbol{Q}_A,\bar{\boldsymbol{Q}}_A]\begin{bmatrix}\boldsymbol{R}_A\\0\end{bmatrix}$$

where $[\underset{k}{\boldsymbol{Q}_A},\underset{m-k}{\bar{\boldsymbol{Q}}_A}]\in\mathbb{R}^{m\times m}$ is orthogonal and $\boldsymbol{R}_A\in\mathbb{R}^{k\times k}$ is nonsingular upper triangular. This factorization can be computed by Householder transformations. Then we have

$$\begin{aligned}&\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}-\boldsymbol{B}\boldsymbol{z}\|^2\\&=\left\|\begin{bmatrix}\boldsymbol{Q}_A^T\\\bar{\boldsymbol{Q}}_A^T\end{bmatrix}\boldsymbol{y}-\begin{bmatrix}\boldsymbol{R}_A\\0\end{bmatrix}\boldsymbol{x}-\begin{bmatrix}\boldsymbol{Q}_A^T\boldsymbol{B}\\\bar{\boldsymbol{Q}}_A^T\boldsymbol{B}\end{bmatrix}\boldsymbol{z}\right\|^2\\&=\|\boldsymbol{Q}_A^T\boldsymbol{y}-\boldsymbol{Q}_A^T\boldsymbol{B}\boldsymbol{z}-\boldsymbol{R}_A\boldsymbol{x}\|^2+\|\bar{\boldsymbol{Q}}_A^T\boldsymbol{y}-\bar{\boldsymbol{Q}}_A^T\boldsymbol{B}\boldsymbol{z}\|^2.\end{aligned}$$

Notice that for any fixed $\boldsymbol{z}$, we can choose $\boldsymbol{x}\in\mathbb{R}^k$ such that the first term on the right hand side of the last equation is equal to zero. Therefore, to solve the MILS problem (1), we first solve the ordinary ILS problem

$$\min_{z\in\mathbb{Z}^n}\|\bar{\boldsymbol{Q}}_A^T\boldsymbol{y}-\bar{\boldsymbol{Q}}_A^T\boldsymbol{B}\boldsymbol{z}\|^2 \tag{5}$$

to obtain the integer solution $\hat{z}\in\mathbb{Z}^n$, and then solve the upper triangular system

$$\boldsymbol{R}_A\boldsymbol{x}=\boldsymbol{Q}_A^T\boldsymbol{y}-\boldsymbol{Q}_A^T\boldsymbol{B}\hat{\boldsymbol{z}} \tag{6}$$

to obtain the real solution $\hat{\boldsymbol{x}}\in\mathbb{R}^k$. If we find p optimal integer solutions to (5), then we can obtain the corresponding p real solutions by solving (6). Thus the key is to solve the ILS problem (5) and it is addressed in the following section.

- Solving the ILS problem (2)

The entire algorithm to solve (2) consists of two processes: reduction and search. The purpose of the reduction process is to make the search process easier and more efficient. In this package, the LLL reduction (Lenstra et al. 1982) is used for the reduction process, but our algorithm is based on the modified LLL algorithms proposed in Zhou (2006). The search algorithm used in this package is based on the one presented in Chang et al. (2005), which is a modification of the Schnorr-Euchner enumeration strategy based algorithm presented in Agrell et al. (2002).

The reduction process transforms the given ILS problem (2) into a new ILS problem and its essential part is the LLL reduction which transforms $\boldsymbol{B}$ into an upper triangular matrix. We can cast the LLL reduction as a matrix factorization, which we refer to as the QRZ factorization:

$$\boldsymbol{Q}^T\boldsymbol{B}\boldsymbol{Z}=\begin{bmatrix}\boldsymbol{R}\\0\end{bmatrix},\quad\text{or}\quad \boldsymbol{B}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\0\end{bmatrix}\boldsymbol{Z}^{-1}=\boldsymbol{Q}_1\boldsymbol{R}\boldsymbol{Z}^{-1}, \tag{7}$$

where $\boldsymbol{Q}=[\underset{n}{\boldsymbol{Q}_1},\underset{m-n}{\boldsymbol{Q}_2}]\in\mathbb{R}^{m\times m}$ is orthogonal, $\boldsymbol{Z}\in\mathbb{Z}^{n\times n}$ is unimodual (i.e., $\boldsymbol{Z}$ is an integer matrix and $|\det(\boldsymbol{Z})|=1$ thus $\boldsymbol{Z}^{-1}$ is also an integer matrix), and $\boldsymbol{R}\in\mathbb{R}^{n\times n}$ is nonsingular upper triangular and satisfies the following two LLL reduction criteria:

$$|r_{ij}| \leq \frac{1}{2}|r_{ii}|, \quad r_{ii}^2 \leq r_{i,i+1}^2 + r_{i+1,i+1}^2 \tag{8}$$

for $i = 1,...,n-1$, $j = i + 1,...,n$. Here the second criterion is a special case of the more general criterion $\delta r_{ii}^2 \leq r_{i,i+1}^2 + r_{i+1,i+1}^2$, where $1/4 < \delta \leq 1$. For the QRZ factorization (7) where $\boldsymbol{R}$ satisfies (8), we refer to it as the LLL-QRZ factorization.

With the LLL-QRZ factorization (7), we have

$$\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{z}\|^2 = \|\boldsymbol{Q}_1^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{Z}^{-1}\boldsymbol{z}\|^2 + \|\boldsymbol{Q}_2^T\boldsymbol{y}\|^2.$$

Then with $\bar{\boldsymbol{y}} \triangleq \boldsymbol{Q}_1^T\boldsymbol{y}$ and $\bar{\boldsymbol{z}} \triangleq \boldsymbol{Z}^{-1}\boldsymbol{z}$, we see that (2) is equivalent to

$$\min_{\bar{z} \in \mathbb{Z}^n} \|\bar{\boldsymbol{y}} - \boldsymbol{R}\bar{\boldsymbol{z}}\|^2. \tag{9}$$

Note that if $\bar{\boldsymbol{z}}$ is a solution to (9), then $\boldsymbol{z} = \boldsymbol{Z}\bar{\boldsymbol{z}}$ is a solution to (2).

The reduction process has reduced the ILS problem (2) to (9). To solve (9), a search process is used to enumerate possible $\boldsymbol{z} \in \mathbb{Z}^n$. Our package has the option to find p optimal solutions to (2).

The search algorithm first finds p integer points, which are in order of increasing residual norms. It then defines

$$\beta = \|\bar{\boldsymbol{y}} - \boldsymbol{R}\bar{\boldsymbol{z}}^{(p)}\|^2, \tag{10}$$

where $\bar{\boldsymbol{z}}^{(p)}$ is the pth integer point. Then the algorithm searches the ellipsoid

$$\|\bar{\boldsymbol{y}} - \boldsymbol{R}\bar{\boldsymbol{z}}\|^2 < \beta$$

to find a new integer point. When a new integer point is found, $\bar{\boldsymbol{z}}^{(p)}$ is removed and the new point is inserted into the sequence of the left $p-1$ integer points so that they are still in order of increasing residual norms. The algorithm then defines a new β by using the last integer point in the sequence as before [see (10)]. Notice that the ellipsoid has been shrunk. Then the search algorithm searches for a new integer point within the new ellipsoid and so on. Finally when no new integer point can be found, the search process is finished and the latest found p integer points are the solutions to (9).

System requirements

The package has been fully tested on Windows XP, Linux and Macintosh with MATLAB 7.x, and should work on any platform supporting MATLAB. For the system requirements for running MATLAB, please refer to:

`http://www.mathworks.com`

Obtaining and installing MILES

The package can be downloaded from the website

`http://www.cs.mcgill.ca/~chang`

The package is provided as a compressed file with extension ''zip'' or ''tar.gz''. To extract the package, an uncompress tool should be used, such as ''winzip'' on Windows and ''gzip'' on Linux. Suppose that the package is extracted to ''C:/MILES'' on Windows, then you can change to the directory and use the provided routines. If you want to use the package in a directory other than ''C:/MILES'', you have to add ''C:/MILES'' to the MATLAB path.

Commericial use and citation

MILES is a freely available software package and can be included in commercial packages. We ask users to give proper credit to the authors by citing this paper, Chang and Zhou (2006a) or Chang and Zhou (2006b) as the official reference in their software packages or publications. This package is copyrighted but not trademarked. However, if any modifications affect the interface, functionality, or accuracy of the resulting software, the name of the routine should be changed. Any modification to our package should be noted in the modifier's documentation.

Support

The MILES project supports the package in the sense that reports of errors or poor performance will gain immediate attention from the developers. Any comments and suggestions for improvement of the code or the document are also welcome. It may still be possible to improve the efficiency of the code by using some programming tricks or MATLAB built-in functions, but for research and educational purpose we have tried to keep the code simple and clear. Error reports and also descriptions of interesting applications and other comments should be sent to:

Prof. Xiao-Wen Chang
School of Computer Science
McGill University
3480 University Street
Montreal, Quebec
Canada H3A 2A7
Email: chang@cs.mcgill.ca
Telephone: 1-514-398-8259

Routines

In this section, we describe the input and output of some routines in this package. Each routine can be used separately without invoking the main function `mils.m`.

qrmcp.m routine

Given $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $\boldsymbol{F} \in \mathbb{R}^{m\times p}$, this routine computes the QR factorization of $\boldsymbol{C}$ with minimum-column pivoting: $\boldsymbol{Q}^T\boldsymbol{CP} = \begin{bmatrix}\boldsymbol{R}\\0\end{bmatrix}$ and computes $\boldsymbol{F}:=\boldsymbol{Q}^T\boldsymbol{F}$. The Q-factor is not produced. The permutation matrix $\boldsymbol{P}$ is encoded in an integer vector $\boldsymbol{piv}$, i.e., $\boldsymbol{P} = \boldsymbol{P}_1...\boldsymbol{P}_{n-1}$, where $\boldsymbol{P}_j$ is the identity with rows j and $\boldsymbol{piv}(j)$ interchanged. The MATLAB function is:

`function [R,F,piv] = qrmcp(C,F)`

Input arguments

- `C` m by n real matrix to be QR factorized
- `F` m by p real matrix to be transformed to `Q' F`

Output arguments

- `R` n by n real upper triangular matrix
- `F` m by p matrix transormed from the input `F` by `Q'`, i.e., `F:= Q' * F`
- `piv` n-vector storing the information of the permutation matrix `P`

reduction.m routine

Given $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ with full column rank and $\boldsymbol{y} \in \mathbb{R}^m$, this routine computes the LLL-QRZ factorization of $\boldsymbol{B} : \boldsymbol{Q}^T\boldsymbol{BZ} = \begin{bmatrix}\boldsymbol{R}\\0\end{bmatrix}$ and computes $\boldsymbol{y}:=\boldsymbol{Q}^T\boldsymbol{y}$. The Q-factor is not produced. Its goal is to reduce a general integer least squares problem to an upper triangular one. The MATLAB function is:

`function [R,Z,y] = reduction(B,y)`

Input arguments

- `B` m by n real matrix with full column rank
- `y` m-dimensional real vector to be transformed to `Q'y`

Output arguments

- `R` n by n LLL-reduced upper triangular matrix
- `Z` n by n unimodular matrix, i.e., an integer matrix with $|\det(\mathrm{Z})| = 1$
- `y` m-vector transformed from the input `y` by `Q'`, i.e., `y:= Q' * y`

search.m routine

Given a nonsingular upper triangular $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{y} \in \mathbb{R}^n$, this routine produces p optimal solutions to the ILS problem $\min_{z\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{Rz}\|^2$ by a search algorithm. The MATLAB function is:

`function Zhat = search(R,y,p)`

Input arguments

- `R` n by n real nonsingular upper triangular matrix
- `y` n-dimensional real vector
- `p` the number of optimal solutions to be found and its default value is 1

Output argument

- `Zhat` n by p integer matrix (in double precision). Its jth column is the jth optimal solution, i.e., its corresponding residual norm is the jth smallest

ils.m routine

Given $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ with full column rank and $\boldsymbol{y} \in \mathbb{R}^{m\times n}$, this routine produces p optimal solutions to the ILS problem $\min_{z\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{Bz}\|^2$. The MATLAB function is:

`function Zhat = ils(B,y,p)`

Input arguments

- `B` m by n real matrix with full column rank
- `y` m-dimensional real vector
- `p` the number of optimal solutions to be found and its default value is 1

Output arguments

- `Zhat` n by p integer matrix (in double precision). Its jth column is the jth optimal solution, i.e., its corresponding residual norm is the jth smallest

mils.m routine

Given $\boldsymbol{A} \in \mathbb{R}^{m\times k}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{y} \in \mathbb{R}^m$, suppose that $[\boldsymbol{A},\boldsymbol{B}]$ is of full column rank. This routine produces p optimal solutions to $\min_{\boldsymbol{x}\in\mathbb{R}^k, z\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{Ax} - \boldsymbol{Bz}\|^2$. The MATLAB function is:

`function [Xhat,Zhat] = mils(A,B,y,p)`

Input arguments

- `A` m by k real matrix
- `B` m by n real matrix, `[A,B]` has full column rank
- `y` m-dimensional real vector
- `p` the number of optimal solutions to be found and its default value is 1

Output arguments

- `Xhat` k by p real matrix
- `Zhat` n by p integer matrix (in double precision). `{Xhat(:,j),Zhat(:,j)}` is the jth optimal solution, i.e., its corresponding residual norm is the jth smallest

Examples

Here we give two examples of using the routines.

Example 1. The following script gives a simple example of solving an ILS problem by using M-file `ils.m`:

```
% Construct data
m = 5; n = 3;
B = randn(m, n);
z_true = [1; -2; 3];
y = B* z_true + 1.e-3* randn(m,1);
p = 2;
% Find p optimal solutions to the ILS
% problem min_{z} ||y-Bz||
Z = ils(B,y,p)
```

After running the above script, we got the following output:

```
Z =
   1    2
  -2   -1
   3    3
```

We see that the first optimal ILS estimate is equal to the true parameter vector.

Example 2. The following script gives a simple example of solving an MILS problem by using routine `mils.m`:

```
% Construct data
m = 7;  k = 2;  n = 3;
A = randn(m,k); B = randn(m,n);
y = randn(m,1);
p = 3;
% Find p pairs of the optimal solutions to
% the MILS problem min_{ x, z} ||y-Ax-Bz||
[ X,Z] = mils(A,B,y,p)
```

After running the above script, we got the following output:

```
X =
    1.0783   1.0649   1.1304
    1.4820   1.0027   2.0476
Z =
    0   -1    0
    2    2    2
    0    0   -1
```

Troubleshooting

Here we list some problems a user may encounter using our routines.

Problem 1: common errors in calling MILES routines

A user should always carefully read the leading comments of a routine before using it. The leading comments describe what the routine can do and give a detailed description of all input/output arguments and they can be viewed at the MATLAB command prompt by typing ''help'' followed by the routine name. For the benefit of users, we list the most common programming errors in calling a routine. These errors may cause the MILES routines or MATLAB to report a failure, or may lead to wrong results without a warning message.

- Wrong number of arguments
- Arguments in wrong order
- Wrong dimensions for an array argument
- The input matrix is rank deficient
- MATLAB path is not set up appropriately.

Problem 2: poor performance in efficiency

One should note that the integer least squares problem is NP-hard. If the dimension of the integer least squares problem is large or the residual norms corresponding the optimal solutions are large, then the computation can be very time-consuming. Another thing we should mention is that MATLAB is slower than some high level programming languages, such as C/C++.

Problem 3: integer overflow

If an integer number produced in the computation is outside of the interval $[-2^{53}+1, 2^{53}-1]$, then its floating point representation in double precision may not be accurate. This may lead wrong integer solutions. Version 1.0 does not check integer overflow and so does not give a warning message if this occurs. For many practical applications, however, this integer overflow phenomena may not be a concern.

Acknowledgments This research was supported by NSERC of Canada Grant RGPIN 217191-03. The authors would like to thank Mike Craymer and Stephen Hilla for their helpful comments and suggestions.

References

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

Chang X-W, Zhou T (2006a) MILES: MATLAB package for solving Mixed Integer LEast Squares problems, Theory and Algorithms. (http://www.cs.mcgill.ca/~chang/software.php)

Chang X-W, Zhou T (2006b) MILES: MATLAB package for solving Mixed Integer LEast Squares problems, Users' Guides (http://www.cs.mcgill.ca/~chang/software.php)

Chang X-W, Paige CC, Yin L (2004) Code and carrier phase based short baseline GPS positioning: computational aspects. GPS Solut 7:230-240

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

De Jonge P, Tiberius CCJM (1996) LAMBDA method for integer ambiguity estimation: implementation aspects. In: Delft Geodetic Computing Center LGR-Series, No.12

Lenstra AK, Lenstra HW, Lovasz L (1982) Factoring polynomials with rational coefficients. Math Ann 261:515–534

Zhou T (2006) Modified LLL Algorithms. Master's Thesis, McGill University, School of Computer Science