EXAMPLE AMBIGUITY RESOLUTION

Dennis Odijk and Peter J.G. Teunissen

Department of Mathematical Geodesy and Positioning
Delft University of Technology
Thijsseweg 11, 2629 JA Delft, The Netherlands

In the following a simple two-dimensional example of GPS ambiguity resolution is given. Based on a float ambiguity solution, the integer solution is computed using the three admissible estimators, i.e. integer rounding, integer bootstrapping and integer least-squares. This is done for the original (DD) ambiguities, as well as the decorrelated ambiguities. The correctness of these solutions is inferred by evaluating their ambiguity success-rates.

1 Integer estimation: original ambiguities

Consider the following two-dimensional float ambiguity solution:

$$\hat{a} = \begin{bmatrix} 2.51 \\ 2.23 \end{bmatrix}; \quad Q_{\hat{a}} = \begin{bmatrix} 0.2767 & 0.2152 \\ 0.2152 & 0.1680 \end{bmatrix} \tag{1}$$

Note that this ambiguity variance-covariance (vc-) matrix corresponds to the dual-frequency ionosphere-fixed geometry-free model for two receivers, two satellites, and two epochs, in which an undifferenced phase standard deviation of 3 mm and an undifferenced code standard deviation of 10 cm is assumed. In the following subsections for this 2D float solution the integer rounding, bootstrapping and least-squares solutions are computed.

1.1 Integer rounding

The integer rounding solution reads simply:

$$\check{a}_R = \begin{bmatrix} nint(2.51) \\ nint(2.23) \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{2}$$

where $nint(\cdot)$ denotes the rounding-to-the-nearest-integer operator. Note that the squared distance of this integer solution to the real solution, in the metric of $Q_{\hat{a}}$, is $\|\hat{a} - \check{a}_R\|^2_{Q_{\hat{a}}} = 592.81$.

1.2 Integer bootstrapping

There are two ways to compute the integer bootstrapping solution: either by starting with the first ambiguity, or by starting with the second ambiguity.

When the algorithm is started with the first ambiguity, the solution reads:

$$\check{a}_B^{(1)} = \begin{bmatrix} nint(2.51) \\ nint(2.23 - \frac{0.2152}{0.2767}(2.51-3)) \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \tag{3}$$

The squared distance of this solution to the float solution (again in the metric of $Q_{\hat{a}}$) is $\|\hat{a} - \check{a}_B^{(1)}\|^2_{Q_{\hat{a}}} = 240.62$. When the second ambiguity is used as starting point, the bootstrapped solution reads:

$$\check{a}_B^{(2)} = \begin{bmatrix} nint(2.51 - \frac{0.2152}{0.1680}(2.23-2)) \\ nint(2.23) \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{4}$$

with the following squared distance to the float solution, $\|\hat{a} - \check{a}_B^{(1)}\|^2_{Q_{\hat{a}}} = 44.96$. Note that both solutions are not equivalent, and they are also not equal to the integer rounding solution.

1.3 Integer least-squares (search)

The integer least-squares minimization problem boils in 2D down to a search over grid-points within an *ellipse* as described by:

$$(\hat{a} - a)^T Q_{\hat{a}}^{-1} (\hat{a} - a) = \chi^2 \tag{5}$$

In Figure 1 this ambiguity search space is visualized. The factor χ^2 has been set to 296.80 to have about 10 candidates lying inside the search space.

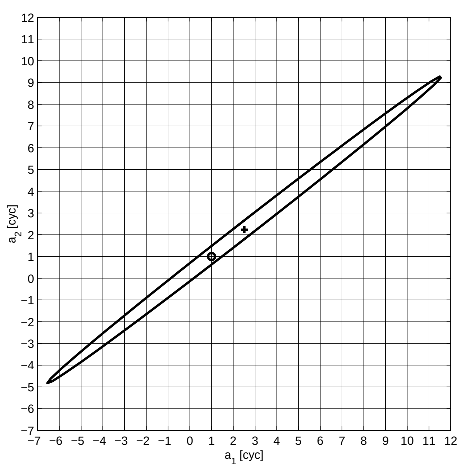

Figure 1: 2D ambiguity search space, centered around float solution (marked by a cross) and integer solution (marked by a circle).

Candidate integer solutions are found by evaluating the bounds of the two ambiguities. The first ambiguity is bounded as follows:

$$\begin{aligned} a_1 &\geq \hat{a}_1 - \sigma_{\hat{a}_1}\chi \approx -6.55 \quad \text{(lower bound)} \\ a_1 &\leq \hat{a}_1 + \sigma_{\hat{a}_1}\chi \approx 11.57 \quad \text{(upper bound)} \end{aligned} \tag{6}$$

The bounding of the second ambiguity depends on the value of the conditional ambiguity $\hat{a}_{2|1}$, and this latter value depends on the value of the candidate integer for a_1. The standard deviation of the conditional ambiguity is $\sigma_{\hat{a}_{2|1}} = 0.0251$. The bounds of the second ambiguity read:

$$\begin{aligned} a_2 &\geq \hat{a}_{2|1} - \sigma_{\hat{a}_{2|1}} \sqrt{\chi^2 - \left(\hat{a}_1 - a_1\right)^2 / \sigma^2_{\hat{a}_1}} \quad \text{(lower bound)} \\ a_2 &\leq \hat{a}_{2|1} + \sigma_{\hat{a}_{2|1}} \sqrt{\chi^2 - \left(\hat{a}_1 - a_1\right)^2 / \sigma^2_{\hat{a}_1}} \quad \text{(upper bound)} \end{aligned} \tag{7}$$

In Table 1 the candidate integer vectors are given. For all candidate integers for a_1 the lower- and upper-bounds according to Eq. (7) are given in the table as well. If one or more integer values for a_2 are within these bounds they are given. If there are no integers within the bounds, this is denoted with a '-'. In this way the complete ambiguity search space is searched for candidate integer vectors.

Table 1: Results LAMBDA search procedure: original ambiguities

a_1	$\hat{a}_{2\|1}$	lower a_2	upper a_2	a_2	$(\hat{a}-a)^T Q_{\hat{a}}^{-1}(\hat{a}-a)$
-6	-4.39	-4.54	-4.24	-	-
-5	-3.61	-3.85	-3.37	-	-
-4	-2.83	-3.13	-2.53	-3	197.33
-3	-2.06	-2.40	-1.71	-2	114.58
-2	-1.28	-1.65	-0.90	-1	195.66
-1	-0.50	-0.90	-0.10	-	-
0	0.28	-0.14	0.69	0	145.17
1	1.06	0.63	1.48	1	13.14
2	1.83	1.40	2.27	2	44.96
3	2.61	2.18	3.04	3	240.62
4	3.39	2.96	3.82	3	247.68
5	4.17	3.75	4.58	4	66.39
6	4.94	4.55	5.34	5	48.94
7	5.72	5.35	6.10	6	195.33
8	6.50	6.16	6.84	-	-
9	7.28	6.98	7.58	7	274.30
10	8.06	7.81	8.30	8	207.59
11	8.83	8.68	8.98	-	-

Table 1 shows that within the search space 13 candidate integer vectors are found. For 5 candidate integers of a_1 however no candidate integers for a_2 could be found. These are called 'dead ends'. From all found candidate vectors the vector $(1,1)^T$ is at shortest squared distance (in the metric of $Q_{\hat{a}}$) from the float solution and is therefore assigned as the *integer least-squares* solution:

$$\check{a}_{LSQ} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{8}$$

Note that this solution is not equal to both integer bootstrapped solutions, nor the integer rounding solution. The squared distance of this integer solution to the float solution is $\|\hat{a} - \check{a}_{LSQ}\|^2_{Q_{\hat{a}}} = 13.14$, which is also the shortest squared distance when compared to the integer rounding and bootstrapping solutions. Note that when the size of the ambiguity search space was set using the *bootstrapped* solution (starting with the most precise ambiguity), it would be set at $\chi^2 = 44.96$, and a much smaller search space could have been searched, but which would still contain the integer least-squares solution.

2 Integer estimation: decorrelated ambiguities

Integer solutions can also be obtained using *decorrelated* ambiguities. In 2D the procedure for constructing the decorrelating Z^T-matrix consists of an alternating application of two basic transformation matrices, Z_a^T and Z_b^T, to the vc-matrix $Q_{\hat{a}}$. These basic transformation matrices read:

$$Z_a^T = \begin{bmatrix} 1 & -nint(\sigma_{\hat{a}_1\hat{a}_2}\sigma_{\hat{a}_2}^{-2}) \\ 0 & 1 \end{bmatrix}; \quad Z_b^T = \begin{bmatrix} 1 & 0 \\ -nint(\sigma_{\hat{a}_2\hat{a}_1}\sigma_{\hat{a}_1}^{-2}) & 1 \end{bmatrix} \tag{9}$$

The procedure ends when the matrices Z_a^T and Z_b^T simplify into identity matrices. In Table 2 the results of the procedure of constructing the Z^T-matrix can be found. According to the table, the last two matrices Z_3^T and Z_4^T are

Table 2: Stepwise construction of the Z-matrix.

i	Z_i^T			$Q_{\hat{z}_i} = Z_i^T Q_{\hat{z}_{i-1}} Z_i$
0	-			$\begin{bmatrix} 0.2767 & 0.2152 \\ 0.2152 & 0.1680 \end{bmatrix}$
1	$\begin{bmatrix} 1 & -nint(\frac{0.2152}{0.2767}) \\ 0 & 1 \end{bmatrix}$	$=$	$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.0143 & 0.0472 \\ 0.0472 & 0.1680 \end{bmatrix}$
2	$\begin{bmatrix} 1 & 0 \\ -nint(\frac{0.0472}{0.0143}) & 1 \end{bmatrix}$	$=$	$\begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.0143 & 0.0043 \\ 0.0043 & 0.0135 \end{bmatrix}$
3	$\begin{bmatrix} 1 & -nint(\frac{0.0043}{0.0135}) \\ 0 & 1 \end{bmatrix}$	$=$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.0143 & 0.0043 \\ 0.0043 & 0.0135 \end{bmatrix}$
4	$\begin{bmatrix} 1 & 0 \\ -nint(\frac{0.0043}{0.0143}) & 1 \end{bmatrix}$	$=$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.0143 & 0.0043 \\ 0.0043 & 0.0135 \end{bmatrix}$

equivalent to the identity matrix, such that the Z^T-matrix is only based on the matrices Z_1^T and Z_2^T. The matrix Z^T plus its inverse read:

$$Z^T = Z_2^T Z_1^T = \begin{bmatrix} 1 & -1 \\ -3 & 4 \end{bmatrix}, \quad Z^{-T} = Z_1^{-T} Z_2^{-T} = \begin{bmatrix} 4 & 1 \\ 3 & 1 \end{bmatrix} \tag{10}$$

Application of the constructed Z^T-matrix to the original ambiguities results in the following decorrelated ambiguities:

$$\hat{z} = Z^T \hat{a} = \begin{bmatrix} 0.28 \\ 1.39 \end{bmatrix}; \quad Q_{\hat{z}} = Z^T Q_{\hat{a}} Z = \begin{bmatrix} 0.0143 & 0.0043 \\ 0.0043 & 0.0135 \end{bmatrix} \tag{11}$$

Although the decorrelating transformation was originally designed to be used in combination with the integer least-squares estimator in the LAMBDA-method, it can also be applied to decorrelate the ambiguities to improve integer estimation based on *rounding* or *bootstrapping*. This is illustrated in the following subsections, in which the decorrelated float solution of Eq. (11) serves as input for the integer rounding, bootstrapping and least-squares estimators.

2.1 Integer rounding

The integer rounding solution in the decorrelated domain is simply obtained as:

$$\check{z}_R = \begin{bmatrix} nint(0.28) \\ nint(1.39) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{12}$$

The squared distance of this rounding solution to the decorrelated float solution in the metric of $Q_{\hat{z}}$ reads $\|\hat{z} - \check{z}_R\|^2_{Q_{\hat{z}}} = 13.14$, which is a much shorter squared distance than the squared distance of the integer rounding solution in the original domain. Back-transforming the integer rounding solution to the original ambiguity domain, results in the following integer solution $\check{a}'_R$:

$$\check{a}'_R = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{13}$$

Note that this transformed solution does *not* correspond to the integer rounding solution obtained using the original ambiguities in Eq. (2): $\check{a}'_R \neq \check{a}_R$.

2.2 Integer bootstrapping

Like in the original ambiguity domain, using the integer bootstrapping estimator two solutions can be obtained, depending on with which ambiguity the process is started. They both read:

$$\begin{aligned} \check{z}_B^{(1)} &= \begin{bmatrix} nint(0.28) \\ nint(1.39 - \frac{0.0043}{0.0143}(0.28 - 0)) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \check{z}_B^{(2)} &= \begin{bmatrix} nint(0.28 - \frac{0.0043}{0.0135}(1.39 - 1)) \\ nint(1.39) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{14}$$

with the following squared distances to the float solution: $\|\hat{z} - \check{z}_B^{(1)}\|^2_{Q_{\hat{z}}} = \|\hat{z} - \check{z}_B^{(2)}\|^2_{Q_{\hat{z}}} = 13.14$. So using decorrelated ambiguities both bootstrapped solutions turn out to be equivalent, in contrast to the solutions in the original domain, see Eq. (3) and (4). Transformation of both decorrelated solutions back to the original domain, results in the following integer solutions, denoted as $\check{a}_B^{(1)'}$ and $\check{a}_B^{(2)'}$:

$$\check{a}_B^{(1)'} = \check{a}_B^{(2)'} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{15}$$

These back-transformed bootstrapped solutions are equivalent since they are obtained from equivalent decorrelated solutions. However they are *not* equal to the bootstrapped solutions obtained in the original domain, see Eq. (3) and (4): $\check{a}_B^{(1)'} \neq \check{a}_B^{(1)}$ and $\check{a}_B^{(2)'} \neq \check{a}_B^{(2)}$.

2.3 Integer least-squares (search)

Figure 2 shows the *transformed* ambiguity search space (using $\chi^2 = 296.80$; the same as in the case with the original ambiguities), which is much less elongated than the original search space in Figure 1.

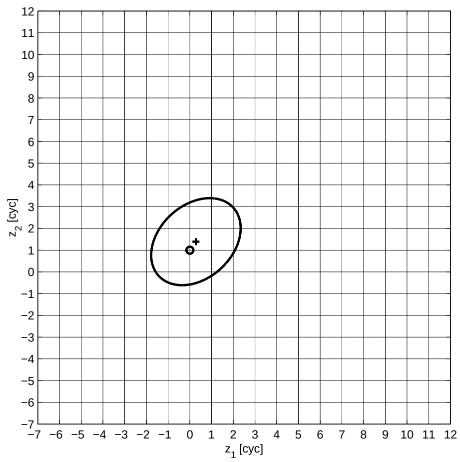

Figure 2: LAMBDA-transformed 2D ambiguity search space, centered around float solution (marked by a cross) and integer solution (marked by a circle).

As result of the LAMBDA-search on the transformed ambiguities, in Table 3 the candidate integer vectors are given. From the table one can see that for each z_1 integer candidate there are *more than one* integer candidates for z_2. Using the original ambiguities however, at most just one candidate for a_2 could be found for a certain candidate integer a_1 (see Table 1). Moreover, using the decorrelated ambiguities no dead ends are found. From this example we may therefore conclude that the search for the integer least-squares solution in the transformed domain is performed in a much more efficient way than in the original domain. The integer least-squares solution in the transformed domain reads:

$$\check{z}_{LSQ} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{16}$$

Note that this solution is equal to the bootstrapped and rounding solutions in the same domain, and thus it also holds that $\|\hat{z} - \check{z}_{LSQ}\|^2_{Q_{\hat{z}}} = 13.14$. Back-transforming the solution in the transformed domain to the original DD-domain, results in the following solution:

$$\check{a}'_{LSQ} = Z^{-T}\check{z}_{LSQ} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{17}$$

This back-transformed solution is exactly the solution as was found using the search in the original domain: $\check{a}'_{LSQ} = \check{a}_{LSQ}$.

Table 3: Results LAMBDA search procedure: decorrelated ambiguities

z_1	$\hat{z}_{2\|1}$	lower z_2	upper z_2	z_2	$(\hat{z}-z)^T Q_{\hat{z}}^{-1}(\hat{z}-z)$
-1	1.01	-0.49	2.50	0	197.33
				1	114.58
				2	195.66
0	1.31	-0.58	3.19	0	145.17
				1	13.14
				2	44.96
				3	240.62
1	1.61	-0.18	3.39	0	247.68
				1	66.39
				2	48.94
				3	195.33
2	1.91	0.86	2.95	1	274.30
				2	207.59

In contrast to the integer rounding and bootstrapping estimators, the integer least-squares estimator is the only estimator for which its solution in the transformed domain is equal to the solution as obtained using the ambiguities in the original domain. This is not surprising, since both minimization problems, in the original as well as the Z-transformed domain, are exactly equivalent. This equivalence does however not hold for the integer rounding and integer bootstrapping estimators. The cause for this phenomenon is explained in the following section, by considering the *pull-in regions* of the different integer solutions.

3 Pull-in regions

Before showing the pull-in regions, assume that the *correct* integer solution for our 2D example corresponds to the estimated integer least-squares solution, denoted as a in the original domain, and denoted as z in the LAMBDA-transformed domain:

$$a = \check{a}_{LSQ} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad z = \check{z}_{LSQ} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{18}$$

In Figures 3 and 4 the pull-in regions of the integer *rounding* estimator are plotted, in the original as well as in the transformed domain. The figures show that in both domains the pull-in regions are simple squares, where each square is centered around a grid-point. The pull-in regions corresponding to the assumed correct integer solution are grey-shaded. Also in the figures the float solution is depicted (marked by a dot), and it can be seen that in the original domain the float vector is lying *outside* the pull-in region of the correct solution. In the Z-domain, it is however *inside* the pull-in region. So in this case it is better to decorrelate the ambiguities first, in order to obtain the correct integer solution.

A similar phenomenon is visible for the integer *bootstrapping* estimator. In Figures 5 and 6 for this estimator the pull-in regions are shown (starting with

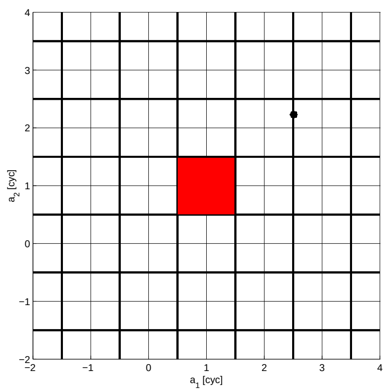

Figure 3: *Integer rounding* in the original domain.

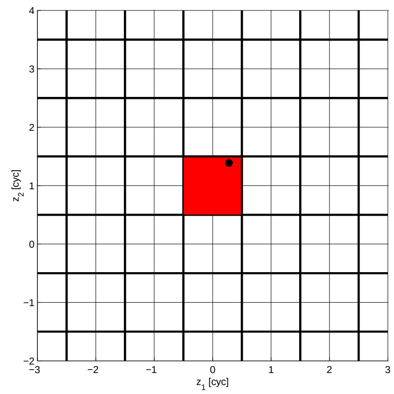

Figure 4: *Integer rounding* in the decorrelated domain.

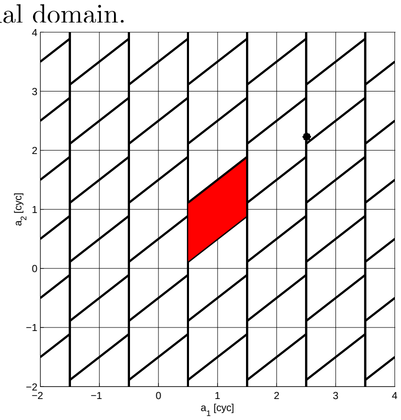

Figure 5: *Integer bootstrapping* in the original domain.

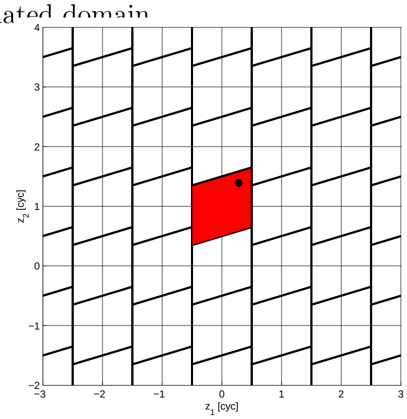

Figure 6: *Integer bootstrapping* in the decorrelated domain.

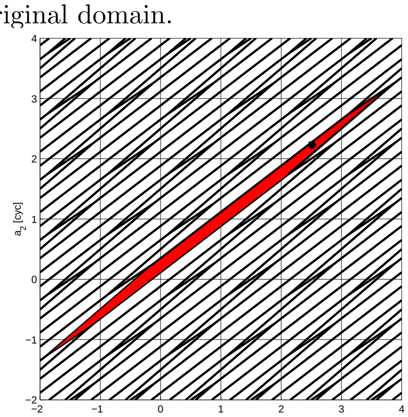

Figure 7: *Integer least-squares* in the original domain.

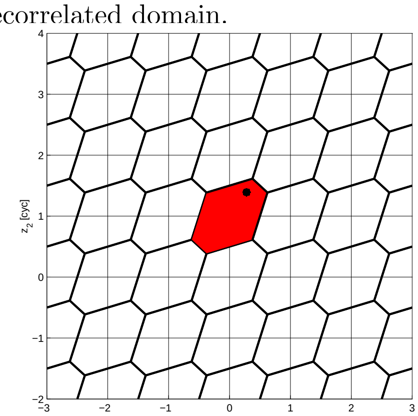

Figure 8: *Integer least-squares* in the decorrelated domain.

the first ambiguity). In 2D these pull-in regions are parallelograms centered around the grid points. Like for integer rounding, in the original domain the float ambiguity vector is *not* mapped to the correct integer solution, however in the Z-domain it is.

Finally, Figures 7 and 8 show the pull-in regions of the integer *least-squares* estimator. In 2D these turn out to be *hexagons* centered around the grid points. As can be seen from the figures, for this integer estimator it makes no difference whether the ambiguities are first decorrelated or not, since its integer solution was assumed as the correct one.

So in this example the float ambiguity solution lies in the pull-in region of the (assumed) correct solution for all the three integer estimators, provided that the ambiguities are first decorrelated. This beneficial effect of decorrelation is also visible in the ambiguity *success-rate*, which is discussed in the next section.

4 Ambiguity success-rates

Table 4 shows for our 2D example the computed ambiguity success-rates plus some lower- and upper-bounds. This has been done for the ambiguities in the original domain, as well as for the decorrelated ambiguities.

Table 4: Example ambiguity success-rates.

success-rate	original amb.	decorrelated amb.
lower-bound rounding	0.51171	0.99995
bootstrapping (1st ambiguity)	0.65816	0.99996
bootstrapping (2nd ambiguity)	0.77749	0.99997
ADOP upper-bound bootstrapping	0.99997	0.99997
simulated least-squares	not computed	0.99998
ADOP upper-bound least-squares	0.99999	0.99999

The tables shows that the probability of integer least-squares is largest: the simulated success-rate using the decorrelated ambiguities is 0.99998. Note that the value of 0.99999 is only an ADOP-based *upper-bound* for the least-squares success-rate, and cannot be used when inferring the correctness of the integer solution. Although the simulated success-rate is largest, it is sharply lower-bounded by the bootstrapping success-rates and the lower-bound for rounding, provided that the ambiguities are decorrelated. In Figures 3-6 this beneficial effect of decorrelation was already visible.

Note that for the success-rate of integer bootstrapping, which can be computed exactly, it makes sense to start the bootstrapping with the most precise ambiguity (denoted with (2) in the table) as this yields a higher success-rate, though there is only a marginal difference in the decorrelated case.

5 Conclusion

From the considered example the following conclusion can be drawn. Although for the integer least-squares solution it makes no difference whether the ambiguities are decorrelated or not, the estimation based on the decorrelated ambiguities turns out to be more efficient (faster) than using the ambiguities in the original domain. Besides, with decorrelated ambiguities it is for the considered example also possible to obtain the correct integer solution using the more simpler integer rounding and bootstrapping estimators.