Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Residual-Guided Boosted-LLL Refinement with Reliability-Time Analysis for GNSS Integer Ambiguity Resolution

Songyan Zhou
University of International Relations
Hong Wu
University of International Relations
Yansen Zhou
zhouys@uir.edu.cn

University of International Relations

Research Article

Keywords: GNSS integer ambiguity resolution, Integer least squares, Lattice reduction, Residual-guided refinement, Candidate budget, Reliability-time analysis

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11387577/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Residual-Guided Boosted-LLL Refinement with Reliability-Time Analysis for GNSS Integer Ambiguity Resolution

Songyan Zhou[1], Hong Wu[1], Yansen Zhou[2*]

[1]Graduate Department, University of International Relations, Beijing, China.

[2]School of Cyber Science and Engineering, University of International Relations, Beijing, China.

*Corresponding author(s). E-mail(s): zhouys@uir.edu.cn;
Contributing authors: zhousyuir@foxmail.com; p0ny@uir.edu.cn;

Abstract

Integer least-squares estimation for Global Navigation Satellite System ambiguity resolution requires balancing solution quality and computational cost. We propose residual-guided Boosted-LLL refinement, which selects coordinates with large conditional rounding residuals and evaluates a bounded neighborhood using the complete objective. Nested candidate sets guarantee non-increasing objective values. Accuracy-Time Conversion Loss and its normalized form assess whether acceptance-probability gains offset additional runtime. Controlled simulation experiments cover 70 settings with dimensions from 16 to 40. Nine-candidate refinement increases the mean probability of satisfying a 1% objective-gap criterion by approximately four and five percentage points at two matched difficulty levels, while exact-fix gains are smaller. Residual guidance produces more frequent objective improvements than random coordinate selection under the same budget. With a target probability of 0.95 and 60 preprocessing reuses, refinement lowers the conversion-cost point estimate in half of the settings.

Keywords: GNSS integer ambiguity resolution, Integer least squares, Lattice reduction, Residual-guided refinement, Candidate budget, Reliability-time analysis

1 Introduction

Carrier-phase Global Navigation Satellite System (GNSS) positioning depends on resolving the integer cycle ambiguities in the observations. The resulting integer least-squares (ILS) problem combines a covariance-weighted distance with an integer constraint. Exact ILS search can recover the minimizer, but its computational cost may become substantial when many correlated ambiguities are processed repeatedly. This motivates refinement methods that improve an initial integer estimate under a bounded additional computational budget. The equivalence between ILS and the closest vector problem (CVP) provides a natural basis for this design [9].

LAMBDA uses integer decorrelation and search to solve the ambiguity problem [22], and MLAMBDA improves the efficiency of these operations [4]. Work on LLL-based decorrelation, alternative factorizations, and stronger reduction has further developed the computational side of ambiguity resolution [6, 10, 11, 14, 17, 25]. Recent work considers constrained ILS formulations [2, 13, 24], multiple-candidate and alternative integer estimation methods [7, 16], and implementation and ambiguity-validation issues [8, 19]. These methods provide reference solutions for evaluating restricted candidate searches.

At the lower-cost end, Babai's nearest-plane algorithm computes an approximate lattice point by successive conditional rounding [1]. LLL reduction can improve its success probability [5], while Boosted LLL (BLLL) strengthens basis reduction [18]. Sampling-based lattice decoding offers another way to generate candidates beyond the Babai point [15]. Boosted LLL itself uses parallel nearest-plane candidate routes during basis reduction [18]. Our refinement acts on the decoded target after reduction: it chooses coordinates from that target's residuals and jointly scores their bounded offsets. A conditional value close to a half-integer admits a neighboring integer with a similar row cost. Because the triangular objective couples coordinates, reconsidering a few such decisions can reduce the complete objective even though the original rounding minimizes each conditional row at the time it is made.

Starting from the BLLL-Babai point, we rank coordinates by their rounding residuals and enumerate offsets in $\{-1, 0, 1\}$ on the k highest-ranked coordinates. The residual measures ambiguity on a common dimensionless scale; the complete ILS objective selects the final candidate. The candidate count is 3^k, and increasing k produces nested sets containing the initial point. Nesting guarantees non-increasing objective values; coordinate ranking determines how effectively the budget finds strict improvements.

We also evaluate whether the resulting probability gains justify the additional computation. Exact integer recovery and proximity to the ILS objective measure different outcomes [23]. We define a separate acceptance event for each and adapt the time-to-solution construction [20] to introduce Accuracy-Time Conversion Loss (ATCL). For independent instances from a fixed setting, ATCL combines mean runtime with the task count required to reach a prescribed probability of at least one acceptable output. Its normalized form, NATCL, expresses this cost relative to a reference solver.

0

This work is supported by Research Funds for NSD Construction of University of International Relations under Grant No. 2025GA05.

A discrete break-even condition compares runtime growth with task-count reduction, while a marginal accuracy-time conversion rate (MACR) evaluates the continuous return on additional search. These quantities enable offline comparisons of solvers and refinement budgets under specified quality targets.

The computational design separates reusable basis reduction from target-specific decoding and local enumeration. Once the reduced basis is available, the refinement budget is fixed by the number of selected coordinates. This separation supports performance comparisons across both single solves and repeated solves with a common covariance, using the same objective and explicit preprocessing accounting.

The contributions are a residual-guided allocation of a bounded candidate budget, a coupled-objective analysis of local improvement, and discrete and continuous cost criteria for evaluating refinement. Paired GNSS-like experiments quantify refinement gains, compare coordinate-selection rules under matched candidate budgets, and assess sensitivity to the objective-gap tolerance, target probability, preprocessing reuse, and refinement budget.

Section 2 defines the ILS model and its CVP representation. Section 3 develops the refinement and its properties. Section 4 derives the conversion analysis, Section 5 presents the experiments, and Section 6 concludes the paper.

2 Problem Formulation

The same covariance-weighted objective is used for all solvers throughout the paper.

2.1 Integer Least-Squares Formulation

Let $\widehat{\mathbf{a}} \in \mathbb{R}^n$ be the float ambiguity estimate, $\mathbf{a} \in \mathbb{Z}^n$ the integer ambiguity vector, and $\mathbf{Q}_{\widehat{a}} \in \mathbb{R}^{n\times n}$ the covariance matrix of the float ambiguities. Assuming $\mathbf{Q}_{\widehat{a}} \succ 0$, the ILS estimator is

$$\mathbf{a}_{\mathrm{ILS}} = \arg\min_{\mathbf{a}\in\mathbb{Z}^n} (\mathbf{a}-\widehat{\mathbf{a}})^{\mathsf{T}} \mathbf{Q}_{\widehat{a}}^{-1} (\mathbf{a}-\widehat{\mathbf{a}}). \tag{1}$$

For any integer candidate $\mathbf{a}$, define

$$J(\mathbf{a}) = (\mathbf{a}-\widehat{\mathbf{a}})^{\mathsf{T}} \mathbf{Q}_{\widehat{a}}^{-1} (\mathbf{a}-\widehat{\mathbf{a}}). \tag{2}$$

Let $\mathbf{a}_\Lambda$ denote the solution returned by LAMBDA. The implementation is checked against bounded brute-force enumeration in Section 5. It is then used as the fixed reference solver in the objective-gap analysis. For method m, the nonnegative relative objective gap is

$$\begin{aligned} G(m) &= \left[\frac{J(\mathbf{a}_m) - J(\mathbf{a}_\Lambda)}{J(\mathbf{a}_\Lambda) + \varepsilon}\right]_+, \\ [x]_+ &= \max(x, 0), \end{aligned} \tag{3}$$

where $\varepsilon = 10^{-12}$ stabilizes normalization near a zero reference objective. With an exact ILS reference, the numerator is nonnegative in exact arithmetic. The positive part removes numerical undershoot; Section 5 reports the reference-solver validation.

2.2 Equivalent CVP Representation of the ILS Problem

Because $\mathbf{Q}_{\widehat{a}} \succ 0$, there exists a nonsingular matrix $\mathbf{B}$ such that

$$\mathbf{Q}_{\widehat{a}}^{-1} = \mathbf{B}^{\mathsf{T}}\mathbf{B}. \tag{4}$$

Define the CVP target $\mathbf{t} = \mathbf{B}\widehat{\mathbf{a}}$. By substituting (4) into (2),

$$\begin{aligned} J(\mathbf{a}) &= (\mathbf{a}-\widehat{\mathbf{a}})^{\mathsf{T}}\mathbf{B}^{\mathsf{T}}\mathbf{B}(\mathbf{a}-\widehat{\mathbf{a}}) \\ &= \|\mathbf{B}(\mathbf{a}-\widehat{\mathbf{a}})\|_2^2 \\ &= \|\mathbf{B}\mathbf{a}-\mathbf{B}\widehat{\mathbf{a}}\|_2^2 \\ &= \|\mathbf{B}\mathbf{a}-\mathbf{t}\|_2^2 . \end{aligned} \tag{5}$$

Therefore, (1) is equivalent to

$$\min_{\mathbf{a}\in\mathbb{Z}^n} \|\mathbf{B}\mathbf{a}-\mathbf{t}\|_2^2 . \tag{6}$$

Here, $\mathbf{B}\mathbf{a}$ is a lattice point and $\mathbf{t}$ is the target. The ILS and CVP representations have identical objective values and minimizers. We use the CVP representation for lattice reduction and refinement, and express the final solution in the original integer ambiguity coordinates.

2.3 GNSS-Like Observation Model

To generate controlled ILS instances, consider $n+1$ observed satellites with one reference satellite. Let $\mathbf{u}_i$ and $\mathbf{u}_r$ denote the line-of-sight unit vectors of satellite i and the reference satellite, respectively. The double-difference geometry vector is $\mathbf{g}_i = \mathbf{u}_i - \mathbf{u}_r$, and stacking all $\mathbf{g}_i$ gives $\mathbf{G}_{\text{geo}} \in \mathbb{R}^{n\times 3}$. The GNSS-like observation model is

$$\mathbf{y}_{\text{obs}} = \mathbf{A}\mathbf{a} + \mathbf{G}_b\mathbf{b} + \boldsymbol{\varepsilon}. \tag{7}$$

Here $\mathbf{y}_{\text{obs}} \in \mathbb{R}^{2n}$, $\mathbf{A} \in \mathbb{R}^{2n\times n}$, and $\mathbf{G}_b \in \mathbb{R}^{2n\times 3}$. The vector $\mathbf{b} \in \mathbb{R}^3$ is the real-valued baseline and $\boldsymbol{\varepsilon}$ is zero-mean observation noise with covariance $\mathbf{Q}_y \in \mathbb{R}^{2n\times 2n}$. The scalar λ below is the carrier wavelength. For the paired carrier-phase and pseudorange-type observations used in the experiments,

$$\mathbf{A} = \begin{bmatrix} \lambda\mathbf{I}_n \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{G}_b = \begin{bmatrix} \mathbf{G}_{\text{geo}} \\ \mathbf{G}_{\text{geo}} \end{bmatrix}.$$

Let $\mathbf{x} = [\mathbf{a}^{\mathsf{T}}, \mathbf{b}^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{H} = [\mathbf{A}\ \mathbf{G}_b]$. Assuming $\mathbf{Q}_y \succ 0$ and that $\mathbf{H}$ has full column rank, the weighted least-squares float solution and covariance are

$$\begin{aligned} \widehat{\mathbf{x}} &= (\mathbf{H}^{\mathsf{T}}\mathbf{Q}_y^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{Q}_y^{-1}\mathbf{y}_{\text{obs}}, \\ \mathbf{Q}_{\widehat{x}} &= (\mathbf{H}^{\mathsf{T}}\mathbf{Q}_y^{-1}\mathbf{H})^{-1}. \end{aligned} \tag{8}$$

The ambiguity block of $\widehat{\mathbf{x}}$ gives $\widehat{\mathbf{a}}$, and the corresponding principal covariance block gives $\mathbf{Q}_{\widehat{a}}$. The experiments use this GNSS-like model to construct $\mathbf{Q}_{\widehat{a}}$, and then perform Monte Carlo sampling directly in the float-ambiguity domain. Specifically, for each trial,

$$\mathbf{a}^\star \sim \mathcal{U}\{-10,\dots,10\}^n, \qquad \widehat{\mathbf{a}} = \mathbf{a}^\star + \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\widehat{a}}). \tag{9}$$

This construction provides a controlled benchmark of the integer-solver stage while preserving the covariance structure induced by the GNSS-like double-difference observation model.

2.4 Ambiguity Dilution of Precision

For an n-dimensional ambiguity covariance matrix, the ambiguity dilution of precision (ADOP) is

$$\mathrm{ADOP} = \det(\mathbf{Q}_{\hat{a}})^{1/(2n)}. \tag{10}$$

A larger ADOP generally indicates weaker ambiguity precision. Its numerical interpretation depends on the observation model and stochastic assumptions. We use ADOP to calibrate difficulty and select settings for matched comparisons.

3 The Proposed BLLL Refinement Method

The refinement starts from a BLLL-Babai solution, selects coordinates using conditional rounding residuals, and evaluates their local candidates with the complete ILS objective.

3.1 BLLL-Babai Initialization

Apply Boosted LLL reduction to the basis in (6):

$$\mathbf{B}_r = \mathbf{B}\mathbf{Z}. \tag{11}$$

Here, $\mathbf{B}_r$ is the reduced lattice basis (subscript r denotes reduced), and $\mathbf{Z} \in \mathbb{Z}^{n\times n}$ is unimodular with $\det \mathbf{Z} = \pm 1$. Let $\mathbf{z} = \mathbf{Z}^{-1}\mathbf{a} \in \mathbb{Z}^n$ be the integer coordinates in the reduced basis, so $\mathbf{a} = \mathbf{Z}\mathbf{z}$. This bijective coordinate change preserves the lattice and gives

$$\min_{\mathbf{z}\in\mathbb{Z}^n} \|\mathbf{B}_r\mathbf{z} - \mathbf{t}\|_2^2 . \tag{12}$$

Let $\mathbf{B}_r = \mathbf{Q}\mathbf{R}$ be a QR factorization and define $\mathbf{c} = \mathbf{Q}^\mathsf{T}\mathbf{t}$. Since $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is nonsingular upper triangular,

$$\begin{aligned} J(\mathbf{Z}\mathbf{z}) &= \|\mathbf{Q}\mathbf{R}\mathbf{z} - \mathbf{t}\|_2^2 \\ &= \left\|\mathbf{R}\mathbf{z} - \mathbf{Q}^\mathsf{T}\mathbf{t}\right\|_2^2 \\ &= \|\mathbf{R}\mathbf{z} - \mathbf{c}\|_2^2 \\ &= \sum_{i=1}^{n} \left(R_{ii} z_i + \sum_{j=i+1}^{n} R_{ij} z_j - c_i \right)^2 . \end{aligned} \tag{13}$$

When $z_{i+1}, \ldots, z_n$ are fixed, the i-th conditional value is

$$\mu_i = \frac{c_i - \sum_{j=i+1}^{n} R_{ij} z_j}{R_{ii}}, \tag{14}$$

and the corresponding row contribution can be written as

$$\phi_i(z_i) = R_{ii}^2(z_i - \mu_i)^2. \tag{15}$$

Babai decoding proceeds from $i = n$ to 1 and sets $z_i^B = \lfloor \mu_i \rceil$, using nearest-integer rounding with ties to even. Let $\mathbf{z}_B = [z_1^B, \ldots, z_n^B]^\mathsf{T}$. The BLLL-Babai output is $\mathbf{a}_{\mathrm{BLLL}} = \mathbf{Z}\mathbf{z}_B$.

3.2 Residual-Guided Coordinate Selection

To allocate the search budget, we prioritize conditional rounding decisions with small normalized cost margins to adjacent integers. Define the Babai rounding residual

$$r_i = \left|\mu_i - z_i^B\right|, \qquad 0 \le r_i \le \frac{1}{2}. \tag{16}$$

The residual has a direct conditional-objective interpretation. Let z_i^A be the adjacent integer toward μ_i (choose either neighbor when $r_i = 0$). Its distance from μ_i is $1 - r_i$, whereas the Babai integer has distance r_i. Substitution into (15) gives

$$\begin{aligned} \phi_i(z_i^A) - \phi_i(z_i^B) &= R_{ii}^2\left[(1 - r_i)^2 - r_i^2\right] \\ &= R_{ii}^2(1 - 2r_i). \end{aligned} \tag{17}$$

Thus a large r_i corresponds to a small normalized margin $M_i/R_{ii}^2 = 1 - 2r_i$, where $M_i = R_{ii}^2(1 - 2r_i)$. Ranking by r_i treats rounding ambiguity on the same scale across rows. Ranking by M_i retains the row scale. Both are local rules: changing coordinate i also changes the earlier rows of the triangular objective. The analysis below makes that coupling explicit, and Section 5.4 compares the two rankings under the same candidate budget.

3.3 Local Refinement and Candidate Selection

Let $\mathcal{I}_k = \{i_1, \ldots, i_k\}$ be the indices of the k largest residuals r_i, with deterministic tie-breaking, and let $\Delta = \{-1, 0, 1\}$. The reduced-domain candidate set is

$$\begin{aligned} \mathcal{Z}_{\mathrm{ref}}^{(k)} = \big\{ \mathbf{z}_B + \boldsymbol{\delta} \;\big|\; &\delta_i \in \{-1, 0, 1\},\ i \in \mathcal{I}_k; \\ &\delta_i = 0,\ i \notin \mathcal{I}_k \big\}. \end{aligned} \tag{18}$$

The subscript "ref" denotes refinement, and (k) denotes the number of selected coordinates. For $0 \le k \le n$, the set contains exactly 3^k distinct points. Define

$$\begin{aligned} \mathbf{z}_{\mathrm{ref}}^{(k)} &= \arg\min_{\mathbf{z} \in \mathcal{Z}_{\mathrm{ref}}^{(k)}} J(\mathbf{Z}\mathbf{z}), \\ \mathbf{a}_{\mathrm{ref}}^{(k)} &= \mathbf{Z}\mathbf{z}_{\mathrm{ref}}^{(k)}. \end{aligned} \tag{19}$$

Residual ranking determines the coordinates to explore, and the complete ILS objective determines the output. Among the 3^k candidates, including the initial Babai point, refinement selects a minimum-objective candidate and maps it back to the original ambiguity coordinates.

Algorithm 1 Residual-Guided BLLL Refinement for ILS

Input: Float ambiguity estimate $\widehat{\mathbf{a}}$, covariance matrix $\mathbf{Q}_{\widehat{a}}$, refinement dimension k, and offset set $\Delta = \{-1, 0, 1\}$

1. Factorize $\mathbf{Q}_{\widehat{a}}^{-1} = \mathbf{B}^{\mathsf{T}}\mathbf{B}$ and set $\mathbf{t} = \mathbf{B}\widehat{\mathbf{a}}$.
2. Apply Boosted LLL reduction to obtain $\mathbf{B}_r = \mathbf{B}\mathbf{Z}$.
3. Compute $\mathbf{B}_r = \mathbf{Q}\mathbf{R}$ and $\mathbf{c} = \mathbf{Q}^{\mathsf{T}}\mathbf{t}$.
4. Apply Babai decoding to obtain $\mathbf{z}_B$, recording μ_i and $r_i = |\mu_i - z_i^B|$.
5. Define $\mathcal{I}_k$ as the indices of the k largest r_i, using deterministic tie-breaking.
6. Enumerate $\mathbf{z} = \mathbf{z}_B + \boldsymbol{\delta}$, with $\delta_i \in \Delta$ for $i \in \mathcal{I}_k$ and $\delta_i = 0$ otherwise. Select a minimizer of $J(\mathbf{Z}\mathbf{z})$, denoted $\mathbf{z}_{\mathrm{ref}}^{(k)}$.
7. **Return** $\mathbf{a}_{\mathrm{ref}}^{(k)} = \mathbf{Z}\mathbf{z}_{\mathrm{ref}}^{(k)}$ (the refined ambiguity vector).

3.4 Theoretical Properties and Computational Complexity

Theorem 1 (Nested-refinement monotonicity and objective reliability) *Fix an instance and its residual ordering, including tie-breaking. The candidate sets contain the Babai point for $0 \le k \le n$. For $0 \le k < n$, they are nested and satisfy*

$$J\left(\mathbf{a}_{\mathrm{ref}}^{(k+1)}\right) \le J\left(\mathbf{a}_{\mathrm{ref}}^{(k)}\right) \le J(\mathbf{a}_{\mathrm{BLLL}}). \tag{20}$$

Thus the nonnegative objective gap to any global ILS minimizer also cannot increase.

Furthermore, for

$$G^{(k)} = \left[\frac{J(\mathbf{a}_{\mathrm{ref}}^{(k)}) - J(\mathbf{a}_\Lambda)}{J(\mathbf{a}_\Lambda) + \varepsilon}\right]_+, \tag{21}$$

and every fixed $\tau \ge 0$,

$$\Pr\left(G^{(k+1)} \le \tau\right) \ge \Pr\left(G^{(k)} \le \tau\right). \tag{22}$$

Proof A fixed ordering gives $\mathcal{I}_k \subseteq \mathcal{I}_{k+1}$. Since every selected coordinate allows a zero offset,

$$\mathbf{z}_B \in \mathcal{Z}_{\mathrm{ref}}^{(k)} \subseteq \mathcal{Z}_{\mathrm{ref}}^{(k+1)}.$$

Minimizing over these nested sets proves (20); global ILS optimality bounds all these objectives from below. The fixed positive denominator and monotone positive-part operator in (21) give $G^{(k+1)} \le G^{(k)}$. Hence $\{G^{(k)} \le \tau\} \subseteq \{G^{(k+1)} \le \tau\}$, which proves (22). □

The guarantee applies to objective quality under any fixed coordinate ordering with nested offsets. Exact fixing is evaluated against the true integer $\mathbf{a}^\star$, which can differ from the minimizer of the realized noisy objective. We therefore evaluate the two events separately. Residual guidance determines which objective improvements become available at a given budget.

The next result characterizes when the local neighborhood contains a candidate that strictly improves the BLLL-Babai objective. Let $\mathbf{e}_B = \mathbf{R}\mathbf{z}_B - \mathbf{c}$.

Proposition 1 (Condition for strict local improvement) *For an allowed refinement offset $\boldsymbol{\delta}$,*

$$\begin{aligned} &J(\mathbf{Z}(\mathbf{z}_B + \boldsymbol{\delta})) - J(\mathbf{Z}\mathbf{z}_B) \\ &\quad = 2\mathbf{e}_B^{\mathsf{T}}\mathbf{R}\boldsymbol{\delta} + \|\mathbf{R}\boldsymbol{\delta}\|_2^2. \end{aligned} \tag{23}$$

Consequently, $J(\mathbf{a}_{\mathrm{ref}}^{(k)}) < J(\mathbf{a}_{\mathrm{BLLL}})$ *if and only if the candidate set contains a nonzero allowed* $\boldsymbol{\delta}$ *such that*

$$2\mathbf{e}_B^{\mathsf{T}}\mathbf{R}\boldsymbol{\delta} + \|\mathbf{R}\boldsymbol{\delta}\|_2^2 < 0. \tag{24}$$

Proof Expanding $\|\mathbf{e}_B + \mathbf{R}\boldsymbol{\delta}\|_2^2 - \|\mathbf{e}_B\|_2^2$ gives (23). The local minimum is strictly below the Babai objective exactly when this increment is negative for at least one allowed offset, giving (24). □

To connect Proposition 1 with the rounding margin, let $\mathbf{e}_i$ be the ith standard basis vector and let $s_i = z_i^A - z_i^B \in \{-1, 1\}$ point toward the adjacent integer (choose either direction when $r_i = 0$). For the single-coordinate offset $s_i\mathbf{e}_i$, upper triangularity of $\mathbf{R}$ gives

$$\begin{aligned} &J\big(\mathbf{Z}(\mathbf{z}_B + s_i\mathbf{e}_i)\big) - J(\mathbf{Z}\mathbf{z}_B) \\ &\quad = M_i + \sum_{j<i} \left(2s_i e_{B,j} R_{ji} + R_{ji}^2\right). \end{aligned} \tag{25}$$

Indeed, rows $j > i$ do not change, row i contributes M_i by (17), and each earlier row changes from $e_{B,j}^2$ to $(e_{B,j} + s_i R_{ji})^2$. The earlier-row sum can be negative and can outweigh the nonnegative conditional margin. This explains how a locally preferred Babai rounding decision can still admit a globally better neighbor. Joint offsets additionally exploit interactions between selected columns through $\|\mathbf{R}\boldsymbol{\delta}\|_2^2$. Residual ranking identifies small normalized row penalties; enumeration resolves the actual coupled change.

The local candidate set contains exactly 3^k points. In the present implementation, the complete ILS objective is evaluated for $\mathcal{O}(3^k)$ local candidates, so the measured refinement time for $k \geq 1$ can be approximated by

$$T_{\mathrm{ref}}^{(k)} \approx T_{\text{BLLL-Babai}} + C_J 3^k, \tag{26}$$

where C_J denotes the mean mapping-and-evaluation cost per local candidate. Direct quadratic-objective evaluation gives an incremental arithmetic upper bound $\mathcal{O}(3^k n^2)$, while residual sorting requires at most $\mathcal{O}(n \log n)$. Therefore, for fixed $k \ll n$, the proposed refinement limits the candidate count, with incremental complexity $\mathcal{O}(n \log n + 3^k n^2)$, while guaranteeing non-increasing objective values. The experiments use $k = 2$ by default and explicitly measure the effect of increasing k.

4 Reliability-Based Accuracy-Time Conversion Analysis

Building on the time-to-solution construction [20], we introduce ATCL and NATCL for the selected ambiguity events and derive discrete and marginal break-even criteria for refinement. The metrics combine mean runtime with the number of independent instances required to obtain at least one acceptable output with a prescribed probability within a fixed setting.

4.1 Exact-Fixing and Objective-Gap Events

For method m in trial t, let $\mathbf{a}_{m,t}$ be its output, $\mathbf{a}_t^\star$ the simulated true ambiguity vector, and $\mathbf{a}_{\Lambda,t}$ the LAMBDA output. The exact-fixing event is

$$X_{m,t}^{\mathrm{fix}} = \mathbf{1}\{\mathbf{a}_{m,t} = \mathbf{a}_t^\star\}\,, \tag{27}$$

with probability $p_m^{\mathrm{fix}} = \Pr(\mathbf{a}_m = \mathbf{a}^\star)$. To evaluate near-reference approximate solutions separately, define

$$G_{m,t} = \left[\frac{J_t(\mathbf{a}_{m,t}) - J_t(\mathbf{a}_{\Lambda,t})}{J_t(\mathbf{a}_{\Lambda,t}) + \varepsilon}\right]_+ \tag{28}$$

and, for tolerance τ,

$$X_{m,t}^{\mathrm{obj}}(\tau) = \mathbf{1}\{G_{m,t} \le \tau\}\,. \tag{29}$$

Its probability is $p_{m,\tau}^{\mathrm{obj}} = \Pr(G_m \le \tau)$. The indicator $\mathbf{1}\{\cdot\}$ equals one when its condition holds and zero otherwise, and J_t is the objective for trial t. In the following, g denotes either exact fixing or an objective-gap event at a specified τ, and $p_m^{(g)} = \Pr(X_m^{(g)} = 1)$. The tolerance τ defines the evaluation event; k determines the algorithm's search budget. Both events use offline reference information: the simulated truth for exact fixing and LAMBDA for objective quality. Solver runtime excludes this reference evaluation. The event tolerance τ and search budget k can be varied independently.

4.2 Reliability-Constrained Accuracy-Time Conversion Loss

For a selected event g, let $T_m \in (0, \infty)$ be the mean cost per task and let $q \in (0, 1)$ be the target probability of obtaining at least one acceptable output. Define the Accuracy-Time Conversion Loss by

$$\mathrm{ATCL}_q^{(g)}(m) = T_m R_q(p_m^{(g)}), \tag{30}$$

where $p = p_m^{(g)}$ denotes the acceptance probability for one instance, and the required integer task count is

$$R_q(p) = \begin{cases} +\infty, & p = 0, \\ 1, & q \le p \le 1, \\ \left\lceil \dfrac{\ln(1-q)}{\ln(1-p)} \right\rceil, & 0 < p < q. \end{cases} \tag{31}$$

For a reference method m_0 with finite positive ATCL, its normalized form is

$$\mathrm{NATCL}_q^{(g)}(m; m_0) = \frac{\mathrm{ATCL}_q^{(g)}(m)}{\mathrm{ATCL}_q^{(g)}(m_0)}. \tag{32}$$

ATCL assigns the fixed task count $R_q(p)$ a cost under the chosen per-task runtime model; smaller values indicate a lower modeled cost for the prescribed target. It equals expected batch runtime when T_m includes the full cost charged to each task. With amortized preprocessing, it is a reuse-normalized cost, as clarified in Section 5. NATCL is dimensionless: values below, equal to, or above one indicate lower, equal, or higher costs than the reference method m_0. We use LAMBDA as the reference and compare methods under the same event, target probability, and experimental setting.

For independent tasks with $0 < p < q$, solving $1 - (1-p)^r \geq q$ for the smallest positive integer r gives (31). The case $p \geq q$ requires one task. This is the standard probability-to-count transformation used in time-to-solution analysis [20].

4.3 ATCL Properties and Refinement Break-Even Analysis

The following result establishes ATCL monotonicity and the refinement break-even condition.

Theorem 2 (Pareto consistency and refinement break-even condition) *For a fixed acceptability event g and target reliability q, the following statements hold.*

(i) Pareto consistency: *let m_1 and m_2 be any two methods evaluated in the same setting. If $T_{m_1} \leq T_{m_2}$ and $p_{m_1}^{(g)} \geq p_{m_2}^{(g)}$, then*

$$\mathrm{ATCL}_q^{(g)}(m_1) \leq \mathrm{ATCL}_q^{(g)}(m_2). \tag{33}$$

(ii) Refinement break-even: *consider BLLL-Babai and BLLL-refinement under the same event, with probabilities p_B, p_R and mean runtimes T_B, T_R. For finite ATCL values, refinement has ATCL no greater than BLLL-Babai if and only if*

$$\frac{T_R}{T_B} \leq \frac{R_q(p_B)}{R_q(p_R)}. \tag{34}$$

Proof The function $R_q(p)$ is non-increasing in p. Hence

$$T_{m_1} R_q(p_{m_1}^{(g)}) \leq T_{m_2} R_q(p_{m_1}^{(g)}) \leq T_{m_2} R_q(p_{m_2}^{(g)}),$$

which proves (i). For (ii), dividing $T_R R_q(p_R) \leq T_B R_q(p_B)$ by the positive quantity $T_B R_q(p_R)$ gives (34). □

For the objective-gap event, nesting gives $p_R \geq p_B$. Equation (34) compares the runtime multiplier T_R/T_B with the reduction factor $R_q(p_B)/R_q(p_R)$ in equivalent task count. Refinement strictly reduces ATCL when the task-count reduction factor exceeds the runtime multiplier.

Since $R_q(p)$ is integer-valued, probability gains reduce the task count only at specific thresholds. For each integer $r \geq 2$, the equality $R_q(p) = r$ holds precisely on the interval

$$1 - (1-q)^{1/r} \leq p < 1 - (1-q)^{1/(r-1)}. \tag{35}$$

Within one such band, a larger probability does not change the task count, so a slower method has larger ATCL. Crossing a boundary can offset a modest runtime increase. Once $p_B \geq q$, both nested methods require one task and ATCL reduces to runtime. The preferred refinement budget therefore depends on the objective-gap tolerance τ and target probability q.

4.4 Continuous Diagnostic and Marginal Gain

To assess the incremental benefit of increasing k, we introduce a continuous relaxation of the task count. Removing the ceiling reveals gains within each constant-count interval of discrete ATCL while retaining the minimum of one task. An overbar denotes the continuous relaxation. The marginal criterion assesses whether the gain from k to $k+1$ compensates for its added runtime. Define

$$\overline{R}_q(p) = \begin{cases} +\infty, & p = 0, \\ \max\left\{1, \dfrac{\ln(1-q)}{\ln(1-p)}\right\}, & 0 < p < 1, \\ 1, & p = 1. \end{cases} \tag{36}$$

and

$$\overline{\mathrm{ATCL}}_q^{(g)}(m) = T_m \overline{R}_q\left(p_m^{(g)}\right). \tag{37}$$

For the non-saturated region $0 < p < q$, define the reliability exponent

$$I(p) = -\ln(1-p). \tag{38}$$

Because $(1-p)^r = \exp[-rI(p)]$, one equivalent task reduces the log-failure probability by $I(p)$. Substitution into (36) yields, for $0 < p < q$,

$$\overline{\mathrm{ATCL}}_q^{(g)}(m) = \ln\left(\frac{1}{1-q}\right) \frac{T_m}{I(p_m^{(g)})}. \tag{39}$$

Thus, within the non-saturated region, comparing methods is equivalent to minimizing the target-independent diagnostic loss

$$\mathcal{L}^{(g)}(m) = \frac{T_m}{-\ln(1-p_m^{(g)})}. \tag{40}$$

When p is small, $-\ln(1-p) = p + \mathcal{O}(p^2)$, and therefore $\mathcal{L}^{(g)}(m) \approx T_m / p_m^{(g)}$. For two methods whose event probabilities both lie in $(0, q)$, the normalized diagnostic is

$$\frac{\mathcal{L}^{(g)}(m)}{\mathcal{L}^{(g)}(m_0)} = \frac{T_m}{T_{m_0}} \frac{-\ln(1-p_{m_0}^{(g)})}{-\ln(1-p_m^{(g)})}. \tag{41}$$

This ratio equals the continuous ATCL ratio when both probabilities lie in $(0, q)$, where the common target factor cancels. At or above q, the one-task floor in (36) applies. We use $\mathcal{L}$ to interpret the local probability–time trade-off and discrete ATCL to report fixed-target comparisons.

For consecutive refinement dimensions, let $p_k^{(g)}$ and T_k denote the event probability and mean runtime at dimension k. When $T_{k+1} > T_k$, define the marginal accuracy-time conversion rate (MACR), measured in inverse time, by

$$\mathrm{MACR}_{k \to k+1}^{(g)} = \frac{I(p_{k+1}^{(g)}) - I(p_k^{(g)})}{T_{k+1} - T_k}. \tag{42}$$

A positive MACR means that the added computation reduces log-failure probability. Its comparison with the current average gives a decision condition. For $0 < p_k^{(g)}, p_{k+1}^{(g)} < q$, write $\Delta I = I(p_{k+1}^{(g)}) - I(p_k^{(g)})$ and $\Delta T = T_{k+1} - T_k > 0$. The strict improvement $\mathcal{L}_{k+1}^{(g)} < \mathcal{L}_k^{(g)}$ holds if and only if

$$\frac{\Delta I}{\Delta T} > \frac{I(p_k^{(g)})}{T_k}, \tag{43}$$

by cross-multiplying the positive denominators in $(T_k + \Delta T)/(I(p_k^{(g)}) + \Delta I) < T_k/I(p_k^{(g)})$.

Therefore,

$$\mathrm{MACR}_{k\to k+1}^{(g)} > \frac{I(p_k^{(g)})}{T_k} \tag{44}$$

is the condition under which increasing k strictly improves the continuous loss. It compares the additional reliability exponent per unit time with its current average $I(p_k^{(g)})/T_k$. This is the local counterpart of the discrete break-even condition: the former measures marginal exchange, while the latter also accounts for integer task-count boundaries. If measured runtimes fail to increase, the losses can be compared directly without dividing by ΔT.

4.5 Empirical Estimation and Uncertainty Quantification

We estimate acceptance probabilities and mean runtimes from paired trials within each setting and propagate probability uncertainty into ATCL. A hat denotes an empirical estimate. For N trials,

$$\widehat{p}_m^{(g)} = \frac{1}{N}\sum_{t=1}^{N} X_{m,t}^{(g)}, \qquad \widehat{T}_m = \frac{1}{N}\sum_{t=1}^{N} T_{m,t}. \tag{45}$$

Within each setting, we compute $\widehat{\mathrm{ATCL}}_q^{(g)}(m) = \widehat{T}_m R_q(\widehat{p}_m^{(g)})$ and normalize by the reference estimate for NATCL, before summarizing across settings. Zero observed successes give an infinite plug-in ATCL, with uncertainty in the underlying probability quantified below. NATCL requires a finite positive reference cost. Computing each ratio before aggregation preserves the task distribution of the comparison.

To quantify probability uncertainty, let p_L and p_U denote, respectively, the lower and upper endpoints of the 95% Wilson confidence interval for $p_m^{(g)}$, computed from the N trial indicators. On the event that this interval contains $p_m^{(g)}$, monotonicity gives $R_q(p_U) \le R_q(p_m^{(g)}) \le R_q(p_L)$. Multiplication by the measured mean runtime yields the probability-induced ATCL interval

$$\left[\widehat{T}_m R_q(p_U),\ \widehat{T}_m R_q(p_L)\right]. \tag{46}$$

This propagates event-probability uncertainty conditional on the measured mean time. If $p_L = 0$, the upper endpoint is infinite. Close to a boundary in (35), an interval may cover several task counts, even when the probability interval is narrow.

For a paired objective comparison, Theorem 1 excludes transitions from an acceptable Babai output to an unacceptable refined output. The probability gain therefore equals the probability that refinement moves an initially unacceptable output across the selected objective-gap threshold. This interpretation connects the objective improvement mechanism to the break-even condition in (34): paired probability differences measure recovered acceptable outputs, while ATCL evaluates their additional computational cost.

5 Experiments

The experiments address three questions: whether refinement improves BLLL-Babai at a small computational cost, whether residual guidance contributes beyond additional candidate evaluations, and when probability gains justify their runtime costs.

5.1 Experimental Setup and Evaluation Metrics

All metrics are computed within each setting from common paired trials.
Solution quality. Exact-fix probability p^{fix} is the fraction of outputs equal to the simulated true integer vector. Objective-gap probability p_{τ}^{obj} is the fraction whose relative ILS objective gap G to LAMBDA does not exceed τ; a larger value indicates more near-reference outputs. Their definitions are given in (27)–(29). The absolute gain $\Delta p = p_{\mathrm{refine}} - p_{\mathrm{BLLL}}$ measures the paired refinement effect; $100\Delta p$ expresses it in percentage points (pp).
Direct objective improvement. Strict-improvement frequency is the fraction of trials with $J_{\mathrm{BLLL}} - J_s > 10^{-12}$, where $J_{\mathrm{BLLL}} = J(\mathbf{a}_{\mathrm{BLLL}})$ and $J_s = J(\mathbf{a}_s)$ for strategy s. Mean relative objective reduction averages $(J_{\mathrm{BLLL}} - J_s)/J_{\mathrm{BLLL}}$ over all paired trials, including unchanged outputs, and is reported as a percentage. Larger values are preferable. These measures distinguish the occurrence and magnitude of improvement from crossing a particular objective-gap threshold.
Cost and conversion. Runtime T_m includes mean solve time and preprocessing amortized over a common reuse count H, as specified in (48). ATCL multiplies this runtime by the independent-task count required for at least one acceptable output at target probability q; NATCL divides that cost by LAMBDA's cost. Smaller values are preferable, and NATCL below one favors the tested method. MACR measures the additional reliability exponent per unit added time; it is compared with the current average in (44). The independent-task model in Section 4 supplies the basis for these comparisons.

The default objective-gap tolerance is $\tau = 1\%$, the target is $q = 0.95$, and $H = 60$. Additional tolerances and targets test sensitivity. ADOP, measured in cycles, describes covariance-based instance difficulty.

We compare Rounding, Direct Babai, LLL-Babai, BLLL-Babai, BLLL-Klein-9, BLLL-refinement, LAMBDA, and MLAMBDA [4]. Rounding ignores the covariance; Direct Babai uses the unreduced CVP basis. The reduction and refinement parameters are listed in Table 2. BLLL-Klein-9 retains Babai and makes eight sampling attempts on the same reduced basis [15], with duplicates evaluated once. At row i, integer z_i has probability proportional to $\exp[-\alpha R_{ii}^2(z_i - \mu_i)^2]$, where $\alpha = \ln(n)/\min_j R_{jj}^2$, over integers within six of the rounded conditional center. It matches the default

Table 1 Comparison of ADOP Settings in Representative GNSS Studies

Study	Observation model and data	Reported or selected ADOP	Main purpose
Odijk and Teunissen [21]	Multi-frequency single-baseline GNSS models; analytical evaluation	0.12 cycles as a strong-model reference	Relating ADOP to formal ambiguity success probability
Castro-Arvizu et al. [3]	GPS-only and GPS–Galileo RTK; real static and dynamic observations	Mean ADOP 0.086–0.338 cycles	Studying the effects of constellation and elevation mask
Khodabandeh [12]	Single-station PPP-RTK and geometry-free models	RTK approximately 0.06–0.20 cycles; weaker GF models approximately 0.31–2 cycles	Demonstrating model- and latency-dependent ambiguity strength
This work	Controlled GNSS-like Monte Carlo ILS model	Calibration approximately 0.15–0.40; matched targets 0.225 and 0.270	Comparing approximate and accurate solvers under moderate and difficult conditions

refinement's nominal nine-candidate budget. LAMBDA is the common objective-gap and NATCL reference.

ADOP characterizes the realized ambiguity precision. Table 1 places our matched targets, 0.225 and 0.270 cycles, alongside representative GNSS studies [3, 12, 21]. These targets select moderate and difficult conditions with measurable approximate-solver gaps; ADOP values depend on the observation model.

The GNSS-like generator uses wavelength 0.1902937 m and stress factor $s = \max\{0, (n-16)/8\}$. Elevations are uniform between $e_{\min} = \max\{5, 10 - s\}$ and $e_{\max} = \max\{e_{\min} + 8, 42 - 1.5s\}$ degrees; azimuths are uniform in a sector of width $\max\{45, 130 - 6s\}$ degrees with a uniform center. The highest-elevation satellite is the reference. For observation type $v \in \{\phi, c\}$ and non-reference elevation e_i in radians,

$$\begin{aligned} d_{v,i} &= \frac{\sigma\sigma_v(s)}{\max\{\sin(e_i), 0.12\}}, \\ \mathbf{Q}_v &= \mathbf{D}_v[(1-c_v)\mathbf{I}_n + c_v\mathbf{1}_n\mathbf{1}_n^{\mathsf{T}}]\mathbf{D}_v, \end{aligned} \tag{47}$$

where $\mathbf{D}_v = \mathrm{diag}(d_{v,1}, \ldots, d_{v,n})$, $\mathbf{1}_n$ is the all-ones vector, $\sigma_\phi(s) = 0.015(1 + 0.06s)$ m, and $\sigma_c(s) = 0.60(1 + 0.20s)$ m. The correlations are

$$\begin{aligned} c_\phi &= \min\{0.30, 0.05 + 0.03s + 0.10(\rho - 0.90)_+\}, \\ c_c &= \min\{0.45, 0.10 + 0.05s + 0.15(\rho - 0.90)_+\}. \end{aligned}$$

Table 2 Main Experimental Parameters

Category	Parameter setting
Evaluation dimensions	$n \in \{16, 20, 24, 28, 32, 36, 40\}$
Scenario and noise	$\rho \in \{0.90, 0.95\}$; $\sigma \in \{0.50, 0.60, 0.70, 0.80, 0.85\}$
Trials per setting	100 for $n = 16$; 80 for $n = 24, 32$; 60 otherwise
Scalability stress test	$n \in \{40, 48, 56\}$; $\rho = 0.95$; $\sigma \in \{0.70, 0.80\}$; 40, 25, 15 trials, respectively
Lattice reduction	$\delta_{\mathrm{LLL}} = 0.99$; 9 Boosted-LLL routes
Refinement	$k = 2$, $\Delta = \{-1, 0, 1\}$; 9 candidates
Additional baselines	BLLL-Klein-9 (8 randomized samples + Babai); MLAMBDA
Quality events	Exact fixing; $G \leq 1\%, 3\%, 5\%$
ATCL/NATCL	$q = 0.95$; LAMBDA reference
Runtime amortization	Common horizon $H = 60$ for all methods/settings

Phase and code blocks are independent. The tunable correlations define the GNSS-like covariance surrogate used in the experiments. We set $\mathbf{Q}_y = \mathrm{diag}(\mathbf{Q}_\phi, \mathbf{Q}_c) + 10^{-12}\mathbf{I}_{2n}$ and regularize the extracted ambiguity covariance by $10^{-12}\mathbf{I}_n$. The scenario label ρ selects the geometry seed and enters c_v. Both labels use the same geometry sampling regime; the resulting ambiguity correlations follow from the constructed covariance.

Table 2 specifies the 70 evaluation settings and the scalability stress test. Within each setting, all methods share the covariance, true ambiguities, and float perturbations. Exact fixing and objective-gap events are evaluated separately. Sensitivity analysis uses $\tau \in \{0.5\%, 1\%, 2\%, 3\%, 5\%\}$ and $q \in \{0.90, 0.95, 0.99\}$ over all evaluation settings.

For repeated solves with a fixed covariance, runtime is measured by a high-resolution performance counter and amortized as

$$T_m^{\mathrm{amort}}(H) = \overline{T}_m^{\mathrm{solve}} + \frac{T_m^{\mathrm{prep}}}{H}, \tag{48}$$

where T_m^{prep} is the one-time preprocessing cost, $\overline{T}_m^{\mathrm{solve}}$ is the mean solve-stage time, and H is the common reuse count. We instantiate T_m in Section 4 with $T_m^{\mathrm{amort}}(60)$. The three BLLL-based solvers share the reduced basis; the budget study also shares the measured $k = 0$ baseline time. Here H is an externally specified reuse horizon, independent of R_q. Consequently, ATCL charges $R_q T_m^{\mathrm{prep}}/H$ for preprocessing. A standalone batch with one preprocessing step costs $T_m^{\mathrm{prep}} + R_q \overline{T}_m^{\mathrm{solve}}$; the two accounting models coincide when $H = R_q$. The reported comparisons use the common-horizon model.

The numerical implementation accepts updates exceeding an objective decrease of 10^{-12} and uses the same tolerance to record strict improvement. The nesting analysis in Section 3 applies to exact candidate-set minima; independently evaluated implementation outputs can differ at the 10^{-12} tolerance scale. We report 95% Wilson probability intervals. The separate high-sample confirmation uses 95% paired bootstrap intervals for $\Delta p = p_{\mathrm{refine}} - p_{\mathrm{BLLL}}$ and exact McNemar tests. All runtime

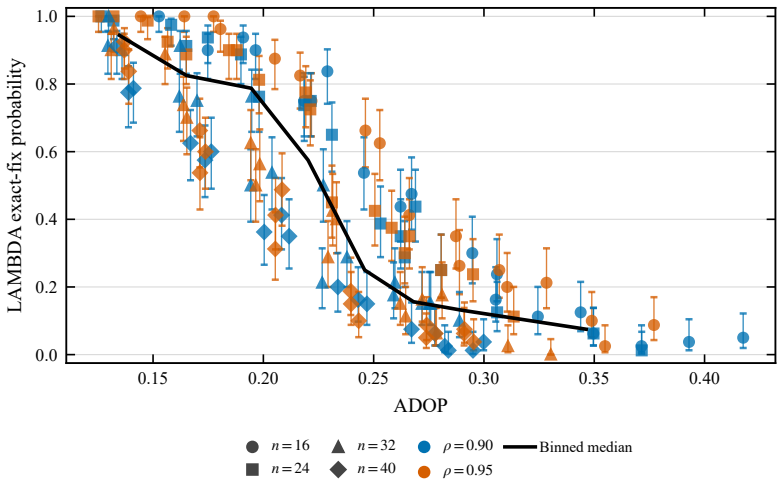


Fig. 1 LAMBDA exact-fix probability versus ADOP over the calibration settings.

measurements use Windows, an Intel Core Ultra 9 185H processor, 16 GB RAM, and Python 3.12.

5.2 Validation and Difficulty Calibration

LAMBDA and MLAMBDA were checked against bounded brute-force enumeration on low-dimensional general ILS and GNSS-like instances. Across 640 validation trials, both methods returned matching solutions with zero recorded objective violations. Their objective values also agreed within the 10^{-8} numerical audit tolerance. Fig. 1 uses LAMBDA as the ILS reference to calibrate the ADOP–difficulty relation before evaluating refinement.

Fig. 1 confirms that the generated settings cover a broad difficulty range and that the LAMBDA exact-fix probability decreases as ADOP increases. The binned medians summarize the observed success-probability trend. Across the corresponding low-, medium-, and high-ADOP regimes, the median exact-fix probability decreases from approximately 0.90 to 0.41 and 0.10, respectively. ADOP therefore provides a suitable difficulty coordinate for the subsequent matched comparisons.

5.3 Refinement Gains and Computational Cost

Fig. 2 compares BLLL-Babai and nine-candidate refinement across dimensions at matched ADOP levels. To control variation in ambiguity precision, we select independently for each dimension the available setting whose ADOP is closest to 0.225 or 0.270 cycles. The selected ranges are approximately 0.217–0.234 and 0.262–0.272 cycles. ADOP matching controls the geometric mean of the covariance-ellipsoid axis scales; covariance shape and dimension remain sources of difficulty variation.

Within each selected setting, BLLL-Babai and refinement share the instances and reduced basis, so their paired comparison measures the effect of the added local search.

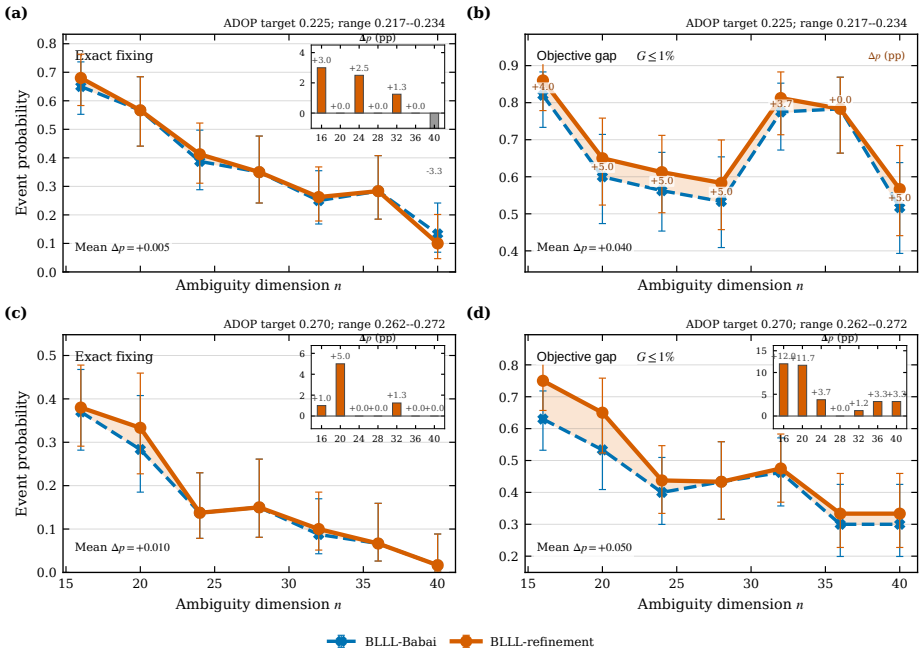


Fig. 2 ADOP-matched event probabilities: exact fixing in (a), (c), and the 1% objective-gap event in (b), (d). Insets or labels show the absolute probability difference $\Delta p = p_{\text{refine}} - p_{\text{BLLL}}$; error bars are 95% Wilson intervals.

Panels (a) and (c) report exact-fix probability, whereas panels (b) and (d) report the 1% objective-gap probability; labels indicate Δp.

At ADOP 0.225, the mean exact-fix difference is about +0.005, while the 1% objective-gap probability increases by about +0.040. At ADOP 0.270, the corresponding differences are approximately +0.010 and +0.050. The exact-fix changes vary by dimension. The marginal Wilson intervals describe each probability separately; inference on the paired difference uses the confirmation experiment below.

A separate paired confirmation experiment uses larger sample sizes for $n \in \{16, 28, 40\}$ at both ADOP targets, with 500 trials for $n = 16$ and 28, and 300 trials for $n = 40$. Across these six settings, the difference in 1% objective-gap probability ranges from +0.010 to +0.068. For $n = 16$ and 28, all paired 95% confidence intervals for this difference lie strictly above zero; the gains are smaller at $n = 40$. The exact-fix difference ranges from −0.004 to +0.028, and its 95% confidence interval often includes zero. The objective-gap probability gains count instances moved across the selected threshold; the exact-fix results measure recovery of the true ambiguity vector.

Runtime and broader baselines.

Fig. 3 compares common-horizon amortized runtime at $\rho = 0.95$ and $\sigma = 0.80$. BLLL-refinement remains close to BLLL-Babai, whereas BLLL-Klein-9 incurs a larger sampling overhead. In the present Python implementations, LAMBDA and MLAMBDA are competitive at lower dimensions but become substantially slower than BLLL-refinement at higher dimensions.

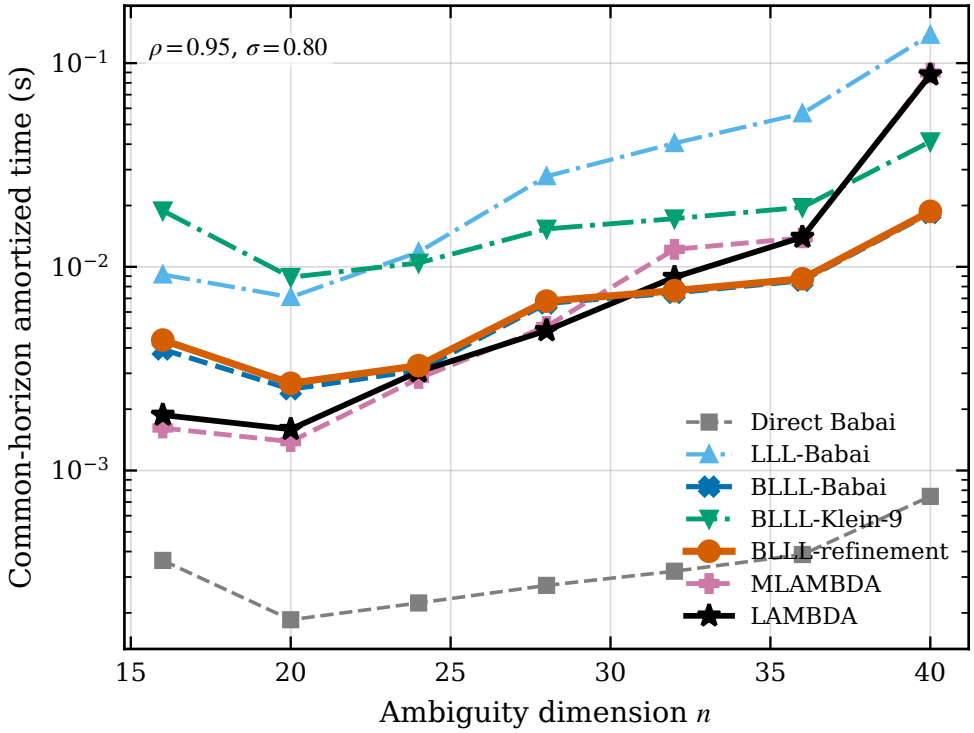


Fig. 3 Common-horizon amortized runtime at $\rho = 0.95$ and $\sigma = 0.80$, with $H = 60$.

At $n = 40$, the measured runtimes of BLLL-Babai, BLLL-refinement, and BLLL-Klein-9 are approximately 0.0184, 0.0187, and 0.0412 s, respectively. Their 1% objective-gap probabilities are 0.317, 0.350, and 0.433. Thus, BLLL-Klein-9 obtains a larger 1% objective-gap probability in this setting but uses more than twice the runtime of the proposed refinement. LAMBDA and MLAMBDA both satisfy the objective-gap criterion and require approximately 0.0876 and 0.0894 s.

A separate stress test uses $n \in \{40, 48, 56\}$, $\rho = 0.95$, and $\sigma \in \{0.70, 0.80\}$. At $n = 56$, the average runtimes of BLLL-refinement, LAMBDA, and MLAMBDA are approximately 0.133, 0.500, and 0.463 s. ADOP also increases with n in this test. The stress test therefore measures the combined effect of larger systems and weaker covariance conditions.

5.4 Contribution of Residual-Guided Coordinate Selection

To isolate residual guidance, we compare coordinate-selection strategies under the same BLLL-Babai initialization and nominal nine-candidate budget. All strategies use $k = 2$, the same three-offset rule, and complete-objective selection. Residual guidance chooses the two largest r_i; the random baseline selects two coordinates uniformly;

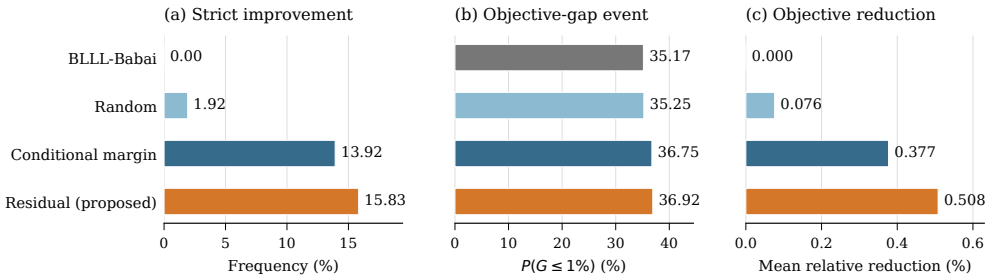


Fig. 4 Matched-budget coordinate-selection ablation: (a) strict-improvement frequency, (b) the 1% objective-gap probability, and (c) mean relative objective reduction. The refinements share the BLLL-Babai initialization and nine-candidate budget. Values aggregate four equally sized groups of 300 paired trials; all axes start at zero and larger values are preferable. Labels give aggregate point estimates at the reported precision.

and the conditional-margin baseline chooses the two smallest $M_i = R_{ii}^2(1 - 2r_i)$. Four paired groups use $n \in \{32, 40\}$, $\rho = 0.95$, and $\sigma \in \{0.70, 0.80\}$, with 300 common trials per group.

Fig. 4 shows strict-improvement frequencies of 1.92% for random coordinate selection and 15.83% for residual guidance. The paired residual-versus-random difference is positive in every group, with exact McNemar $p < 10^{-8}$ in all four comparisons. Conditional-margin selection also improves on random selection. Its aggregate strict-improvement frequency, 1% objective-gap probability, and mean relative objective reduction are slightly lower than those of residual guidance; this comparison is descriptive. All three refinements contain the Babai point and retain its objective bound.

5.5 ATCL-Based Cost Comparison

For each setting and fixed (g, q, H), we compute ATCL from the estimated acceptance probability and amortized runtime, and normalize by LAMBDA to obtain NATCL. Fig. 5 compares refinement with its Babai baseline and illustrates the probability–time trade-off. Sensitivity analysis varies τ, q, and H.

Fig. 5(a) reports $\log_{10}(\mathrm{NATCL}_{\mathrm{refine}} / \mathrm{NATCL}_{\mathrm{BLLL}})$ for the 1% objective-gap event at $\rho = 0.95$. The common LAMBDA normalization cancels, so the displayed quantity is also the ATCL log-ratio. Negative entries identify settings where the reduction in equivalent task count outweighs the refinement overhead. Positive entries indicate that the probability gain does not reduce the required task count sufficiently to offset the added runtime.

At the setting in Fig. 5(b), refinement increases the objective-gap probability from approximately 0.317 to 0.350, while adding 0.337 ms, or about 2%, to the runtime. At $q = 0.95$, the required task count decreases from eight to seven. The runtime multiplier is approximately 1.018, below the task-count reduction factor of 1.143, so the break-even condition in (34) is satisfied. The ATCL ratio is approximately 0.891: a 3.3-percentage-point probability gain reduces the ATCL point estimate by about 11%.

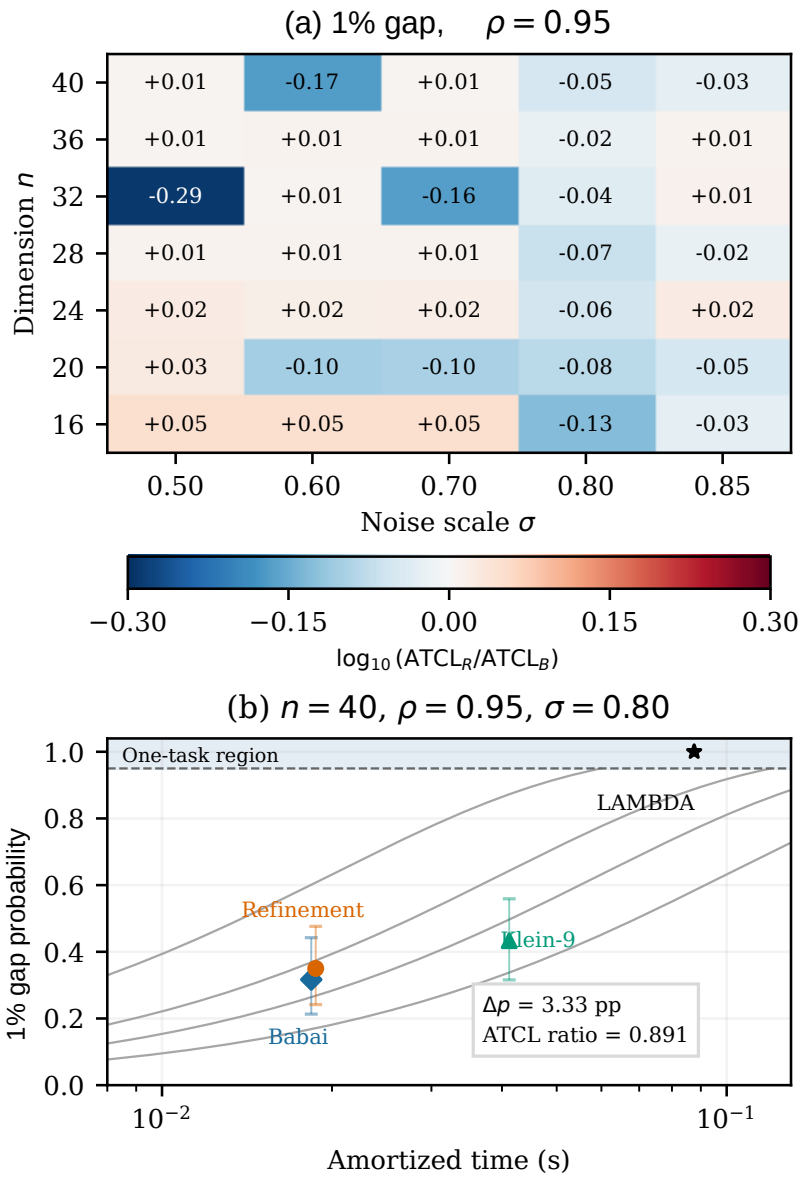


Fig. 5 Accuracy–time conversion with $H = 60$ and $q = 0.95$: (a) ATCL log-ratio for refinement versus BLLL-Babai at $\rho = 0.95$; (b) the probability–time plane at $n = 40$, $\rho = 0.95$, and $\sigma = 0.80$. Curves show $\mathcal{L} = T/[-\ln(1-p)]$ below $p = q$; the shaded region uses the one-task floor. Error bars in (b) are 95% Wilson intervals; the objective-gap probability of LAMBDA is 1 by definition.

The continuous-loss ratio is approximately 0.90, reflecting the corresponding smooth change in the reliability exponent.

Applying the secant comparison underlying (44) to the move from $k = 0$ to $k = 2$ gives an incremental reliability exponent per unit time of approximately 148.6 s^{-1}, compared with the baseline average of 20.7 s^{-1}. The added search therefore improves continuous conversion efficiency. This comparison assesses the combined two-coordinate budget; individual adjacent increments are examined in Section 5.6.

For comparison, BLLL-Klein-9 has a higher probability, 0.433, but requires about 0.0412 s per task and six equivalent tasks. Its ATCL is approximately 0.247 s, versus 0.131 s for refinement, 0.147 s for BLLL-Babai, and 0.0876 s for LAMBDA. The extra sampling improves quality but costs more under this target; LAMBDA attains the objective-gap event in one task and has the lowest ATCL among these four methods. The probability-induced ATCL intervals from (46) are $[0.110, 0.239]$ s for BLLL-Babai and $[0.094, 0.206]$ s for refinement, conditional on their measured mean times. The task-count reduction explains the point-estimate gain; its uncertainty reflects the 60 trials in this setting.

A sensitivity analysis over all 70 evaluation settings examines how the operating target affects this trade-off. For the 1% objective-gap event, refinement has lower ATCL than BLLL-Babai in 42.9%, 50.0%, and 57.1% of the settings for $q = 0.90, 0.95, 0.99$, respectively. When the objective-gap tolerance is relaxed to 5%, these fractions fall to 20.0%, 22.9%, and 35.7%. At $q = 0.95$, the median refinement-to-BLLL ATCL ratio is 0.942, 0.982, 1.025, 1.026, and 1.026 for $\tau = 0.5\%, 1\%, 2\%, 3\%, 5\%$, respectively. The trend agrees with the break-even analysis: a looser tolerance more often places the baseline in, or near, the one-task saturation regime, leaving less reduction in the required task count to offset the refinement overhead.

Reuse changes the relative weight of preprocessing and local enumeration. At the 1% objective-gap event and $q = 0.95$, the median refinement-to-Babai ATCL ratio across the 70 settings is 0.965, 0.982, and 1.092 for $H = 1, 60, 300$, respectively. The corresponding fractions favoring refinement are 50.0%, 50.0%, and 31.4%. As reduction is amortized over more tasks, local search occupies a larger fraction of the per-task cost; its probability gain must compensate for a larger relative overhead. These values reuse the recorded preprocessing and solve times in (48).

5.6 Refinement-Budget Analysis

The budget study evaluates the cost and quality gains from enlarging the residual-guided neighborhood. We compare discrete ATCL across budgets and apply (44) to adjacent increments to interpret their continuous exchange efficiency. The comparison with $k = 0$ removes refinement; positive k values form a budget-sensitivity study.

The refinement dimension k controls the local-search size. We evaluate $k \in \{0, 1, 2, 3, 4, 5\}$ on four paired groups with $n \in \{32, 40\}$, $\rho = 0.95$, and $\sigma \in \{0.70, 0.80\}$. Each group has 100 trials. All values of k use the same instances. The case $k = 0$ is BLLL-Babai.

Fig. 6(a) shows that the first one or two selected coordinates capture most of the measured gain in objective-gap probability. The nondecreasing objective-gap probability trend agrees with Theorem 1, while the nearly flat exact-fix curve indicates little net change in correct recovery.

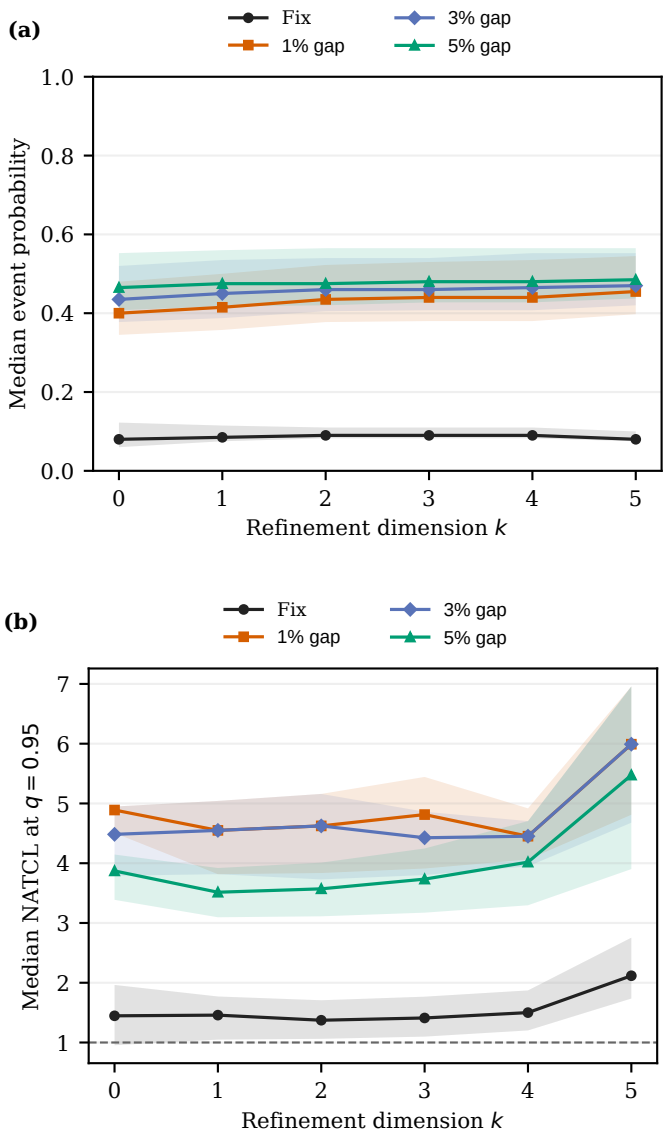


Fig. 6 Paired refinement-budget analysis: (a) median event probabilities for exact fixing and the 1%, 3%, and 5% objective-gap events; (b) median NATCL at $q = 0.95$ on a linear axis. Lines show medians across paired settings; shading gives the between-setting 25th–75th percentile range. Each NATCL is computed within its setting using $H = 60$ before aggregation.

The candidate count rises from 9 to 27, 81, and 243 for $k = 2, 3, 4, 5$. Fig. 6(b) shows little sustained reduction in conversion cost beyond the first few coordinates and a marked increase at $k = 5$. The median NATCL values for the objective-gap events are near their minima at $k = 1$, while the exact-fix NATCL reaches its minimum near

Table 3 Adjacent-Budget Conversion at the 1% Objective-Gap Event

Step	Candidates	ΔT (ms)	Δp (pp)	MACR met	ATCL reduced
0→1	3	0.097	2.0	4/4	2/4
1→2	9	0.109	2.5	4/4	0/4
2→3	27	0.298	0.5	1/4	0/4
3→4	81	0.686	0.0	0/4	1/4
4→5	243	3.078	1.0	0/4	0/4

ΔT and Δp are medians across the four paired settings. The last two columns count settings satisfying (44) and reducing discrete ATCL, respectively, at $q = 0.95$ and $H = 60$.

$k = 2$. In these four tested settings, $k = 2$ captures most of the observed gain in objective-gap probability with nine candidates and serves as a compact default budget. The discrete ATCL optimum depends on the event and setting.

Table 3 evaluates the consecutive increments within each of the four paired settings. Both $0 \to 1$ and $1 \to 2$ satisfy the continuous marginal condition in every setting. Only one setting satisfies it for $2 \to 3$, and none does for the last two increments. The median added time grows from about 0.10 ms in each of the first two steps to 3.08 ms in the final step, while the added objective-gap probability remains small.

The discrete decisions differ in a way predicted by (35). The $1 \to 2$ move improves continuous loss in all four settings but lowers discrete ATCL in none, because it leaves their task counts unchanged. Conversely, the $3 \to 4$ move at $n = 32$, $\sigma = 0.70$ reduces the count from four to three: the probability rises from 0.62 to 0.64, enough to cross the $r = 3$ boundary, even though the continuous marginal condition fails. The task-count criterion and the marginal criterion thus answer complementary budget questions. The first two coordinates give consistent continuous benefit; a later isolated discrete gain can result from crossing a task-count boundary.

The median NATCL curves exceed one, indicating median normalized costs above the LAMBDA reference. Within the refinement family, the first two coordinates provide consistent continuous-loss improvements in the tested settings. Discrete ATCL selects budgets according to whether those probability gains cross task-count boundaries and offset the added runtime.

6 Conclusion

We developed a residual-guided local refinement of the BLLL-Babai solution for GNSS integer ambiguity resolution. The method explores bounded offsets on coordinates with large conditional rounding residuals and selects the candidate with the lowest complete ILS objective. Nested candidate sets retain the Babai point and guarantee non-increasing objective values as the budget grows.

Controlled GNSS-like experiments show that refinement primarily improves proximity to the reference ILS objective, while gains in exact integer recovery are smaller and less consistent. Under matched nine-candidate budgets, residual guidance produces more frequent objective improvements than random selection. Its aggregate improvement frequency, objective-gap probability, and mean relative objective reduction are also slightly higher than those of conditional-margin selection; the latter comparison is descriptive.

ATCL quantifies whether acceptance-probability gains offset the additional runtime under a specified event, target probability, and preprocessing reuse count. In the tested budget-sensitivity settings, $k = 2$ captures most of the observed objective-gap probability gain with nine candidates. The discrete ATCL optimum varies with the event and setting, whereas the first two budget increments consistently improve the continuous loss. The evaluation uses controlled GNSS-like Monte Carlo instances; future work will examine real observations and instance-adaptive budgets with ambiguity validation.

Declarations

Funding

This work is supported by Research Funds for NSD Construction of University of International Relations under Grant No. 2025GA05.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contributions

S.Z. conceived the study, developed the methodology, implemented the software, performed the formal analysis and investigation, and wrote the original draft. H.W. prepared the manuscript formatting and template, coordinated the submission, and reviewed and edited the manuscript. Y.Z. contributed to the conceptualization and supervision, validated the results, and reviewed and edited the manuscript. All authors read and approved the final manuscript.

Data availability

The data supporting the findings of this study are generated by the simulation code described in the Code Availability statement. The simulation settings, summary statistics and trial-level result files are available from the corresponding author on reasonable request.

Code availability

The lattice-reduction, refinement and simulation code used to generate the results is available from the corresponding author on reasonable request.

References

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

[2] Brack A, He S (2026) Inequality-constrained integer least-squares and its application to troposphere-bounded GNSS PPP. GPS Solutions 30:23

[3] Castro-Arvizu JM, Medina D, Ziebold R (2020) Impact of Satellite Elevation Mask in GPS+Galileo RTK Positioning. In: Proc. 2020 Int. Tech. Meeting Inst. Navig., San Diego, CA, USA, pp 487–498

[4] Chang XW, Yang X, Zhou T (2005) MLAMBDA: A modified LAMBDA method for integer least-squares estimation. J Geodesy 79:552–565. https://doi.org/10.1007/s00190-005-0004-x

[5] Chang XW, Wen J, Xie X (2013) Effects of the LLL reduction on the success probability of the Babai point and on the complexity of sphere decoding. IEEE Trans Inf Theory 59(8):4915–4926

[6] Grafarend EW (2000) Mixed integer-real valued adjustment (IRA) problems: GPS initial cycle ambiguity resolution by means of the LLL algorithm. GPS Solutions 4(2):31–44

[7] Gu S, Fang L, Jiang W (2024) Multiple integer candidates ambiguity resolution: a unification ambiguity resolution algorithm. Satell Navig 5:21

[8] Guo J, Geng J (2025) Integer ambiguity validation through machine learning for precise point positioning. Satell Navig 6:14

[9] Hassibi A, Boyd SP (1998) Integer parameter estimation in linear models with applications to GPS. IEEE Trans Signal Process 46(11):2938–2952

[10] Jazaeri S, Amiri-Simkooei AR, Sharifi MA (2012) Fast integer least-squares estimation for GNSS high-dimensional ambiguity resolution using lattice theory. J Geodesy 86(2):123–136. https://doi.org/10.1007/s00190-011-0501-z

[11] Jiao Y, Li K, Tian C, et al (2023) Improved GNSS integer ambiguity resolution method based on the column oriented Cholesky decomposition. Sci Rep 13:4454

[12] Khodabandeh A (2021) Single-station PPP-RTK: correction latency and ambiguity resolution performance. J Geodesy 95:42

[13] Khodabandeh A, Teunissen PJG (2024) Bias-constrained integer least squares estimation: distributional properties and applications in GNSS ambiguity resolution. J Geodesy 98:40

[14] Li X, Xiong Y, Chen W, et al (2023) Improved GNSS Ambiguity Fast Estimation Reduction Algorithm. Sensors 23(20):8568

[15] Liu S, Ling C, Stehlé D (2011) Decoding by sampling: A randomized lattice algorithm for bounded distance decoding. IEEE Trans Inf Theory 57(9):5933–5945

[16] Liu Y, Liu W, Zhang X, et al (2024) An improved GNSS ambiguity best integer equivariant estimation method with Laplacian distribution for urban low-cost RTK positioning. Satell Navig 5:12

[17] Lou L, Grafarend EW (2003) GPS integer ambiguity resolution by various decorrelation methods. Z Vermessungswesen 128(3):203–210

[18] Lyu S, Ling C (2017) Boosted KZ and LLL algorithms. IEEE Trans Signal Process 65(18):4784–4796. https://doi.org/10.1109/TSP.2017.2708020

[19] Massarweh L, Verhagen S, Teunissen PJG (2025) New LAMBDA toolbox for mixed-integer models: estimation and evaluation. GPS Solutions 29:14. https://doi.org/10.1007/s10291-024-01738-z

[20] nnow TFR, Wang Z, Job J, et al (2014) Defining and detecting quantum speedup. Science 345(6195):420–424. https://doi.org/10.1126/science.1252319

[21] Odijk D, Teunissen PJG (2008) ADOP in closed form for a hierarchy of multi-frequency single-baseline GNSS models. J Geodesy 82(8):473–492. https://doi.org/10.1007/s00190-007-0197-2

[22] Teunissen PJG (1995) The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation. J Geodesy 70:65–82. https://doi.org/10.1007/BF00863419

[23] Wu Z, Bian S, Liu Y, et al (2025) A study on three commonly used GNSS integer estimators with probability of correct estimation. GPS Solutions 29:18

[24] Zhang C, Dong D, Kubo N, et al (2024) Evaluation of different constrained LAMBDAs for low-cost GNSS attitude determination in an urban environment. GPS Solutions 28:42

[25] Zhi G, Shi J (2023) Resolve integer ambiguity based on the global deep grid-based algorithms. Sci Rep 13:20597