On Solving the Shortest Basis Problem Based on Sequential Reduction

Shanxiang Lyu, *Member, IEEE*, Zheng Wang, *Member, IEEE*, and Ling Liu

***Abstract*— Low-complexity lattice reduction algorithms are generally not optimized for solving the shortest basis problem (SBP). We fill this blank by tweaking the recently introduced sequential reduction (SR). In a quest for developing a provable and low-complexity reduction algorithm under the SR framework, we propose to employ successive interference cancellation (SIC) as a subroutine inside SR, and the whole algorithm is referred to as SR-SIC. On the theoretical front, we prove that the upper bound on the basis length of SR-SIC is better than those of major Lenstra, Lenstra, and Lovász (LLL) based variants when the dimension of the lattice basis is no larger than 4. In practice, we show by simulations that SR-SIC yields higher information rate than LLL when designing integer-forcing linear receivers, in which SR-SIC also enjoys lower computational complexity.**

***Index Terms*— Sequential reduction, successive interference cancellation (SIC), shortest basis problem (SBP), integer-forcing.**

I. Introduction

LATTICES are discrete subgroups of $\mathbb{R}^n$. They have been used in communications to construct capacity achieving codes [1], [2] and to design integer-forcing linear receivers that achieve the optimal degrees-of-freedom [3]–[5]. Lattice reduction is a process to reach another basis with short and nearly orthogonal vectors when given an input basis. In the application to integer-forcing, lattice reduction is conceived as an essential step in the process of designing and decoding.

Lattice reduction has been studied since the time of Gauss who gave an algorithm that works for two dimensional lattices. In a celebrated result, Lenstra, Lenstra, and Lovász [6] (in short LLL) gave a polynomial time algorithm for approximating short lattice vectors. To date, a large number of generalizations of LLL have been proposed to optimize the implementation of swaps (see, *e.g.*, MLAMBDA [7], greedy LLL [8], deep LLL [9]) and size reduction (see, *e.g.*, effective LLL [10], boosted LLL [11]). A blockwise generalization of LLL is the block Korkine-Zolotarev (BKZ) reduction algorithm [9], which can generally find shorter vectors than LLL at the cost of higher computational complexity. Another framework called Seysen reduction [12] was proposed in the 1990s, it is however unable to compete with LLL variants when the problem dimension is large.

There exist many computationally hard problems over lattices. The most well known two are the shortest vector problem (SVP) and the closest vector problem (CVP), where SVP asks for the shortest non-zero vector in the lattice and CVP requires finding the closest lattice point to a given query. *While the aforementioned lattice reduction algorithms naturally fit into solving SVP or CVP, they are not the most suitable low-complexity candidates for another lattice problem termed the shortest basis problem (SBP), as SBP only pursuits making the longest basis vector short.*

To solve SBP efficiently, we resort to a flexible lattice reduction framework dubbed sequential reduction (SR) proposed in [13]. Different from LLL-based generalizations and Seysen reduction, SR resembles a series of lattice reduction algorithms that hinge on using exact/approximate CVP algorithms as subroutines. The element-based reduction [14] and the Greedy reduction [15] are both special forms of SR, in which either a pairwise nearest neighbor search or a CVP algorithm is used as a subroutine. Unfortunately, while [13] has shown the performance bounds for an SR algorithm that employs an exact CVP subroutine (*i.e.*, SR-CVP), this SR-CVP algorithm entails exponential computational complexity.

The scope of this letter to develop a version of SR that solves SBP efficiently. While the designed algorithm is expected to enjoy much lower complexity than others practically, it will also be interesting to derive its theoretical performance bounds. For this reason, we resort to the celebrated successive interference cancellation (SIC) algorithm [16] as a subroutine inside SR, which is also known as Babai's nearest plane algorithm [17]. It is noteworthy that in previous works SIC was conceived as a tool used after lattice reduction to perform lattice-reduction-aided MIMO detection (cf. [8], [16]), but here SIC is deployed inside the lattice reduction algorithm solely to reduce the length of the longest basis vector. Against this background, the contributions of this work are two-fold:

I) We prove that, for dimensions $n \leq 4$, the proposed SR-SIC algorithm can achieve a bound on the shortness of the basis. To be concise, an SR-SIC reduced basis has $l(\mathbf{B}) \leq \frac{2}{(5-n)^{1/2}}\lambda_n(\mathbf{B})$, while that of LLL is $l(\mathbf{B}) \leq 2^{n-1}\lambda_n(\mathbf{B})$, where $l(\mathbf{B})$ denotes the length of the longest basis vector and $\lambda_n(\mathbf{B})$ denotes the nth successive minimum of the lattice. The proven result suggests that, at least for a small

Manuscript received November 21, 2020; accepted December 7, 2020. Date of publication December 9, 2020; date of current version May 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 61902149, 61801216, 61932010, and 62032009, in part by the Natural Science Foundation of Guangdong Province under Grant 2020A1515010393, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420, in part by the Fundamental Research Funds for the Central Universities under Grant 21620438, and in part by the Major Program of Guangdong Basic and Applied Research under Grant 2019B030302008. The associate editor coordinating the review of this letter and approving it for publication was B. Matuz. *(Corresponding author: Shanxiang Lyu.)*

Shanxiang Lyu is with the College of Cyber Security, Jinan University, Guangzhou 510632, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: shanxianglyu@gmail.com).

Zheng Wang is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China (e-mail: z.wang@ieee.org).

Ling Liu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: liulingcs@szu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2020.3043499

dimensional basis, the SR-SIC algorithm can nearly compete with the exponential time Korkine-Zolotarev (KZ) and Minkowski algorithms [18].

II) We show by practical examples that SR-SIC outperforms LLL when solving SBP, and this supremacy is not restricted to small-size problems. From an information-theoretic perspective, in this work we consider the application of SR-SIC to design the effective channel matrix in integer-forcing. Our simulation results show that SR-SIC attains higher information rate than LLL while costing less time.

Notation: Matrices and column vectors are denoted by uppercase and lowercase boldface letters. The ith column and the (i,j)th entry of $\mathbf{B}$ are respectively noted as $\mathbf{b}_i$ and $b_{i,j}$. $\mathbf{I}$ and $\mathbf{0}$ respectively denotes an identity matrix and a zero vector. The operation $(\cdot)^\top$ represents matrix transposition. $\lfloor x \rceil$ denotes rounding x to the nearest integer, $|x|$ denotes getting the absolute value of x, and $\|\mathbf{x}\|$ denotes the Euclidean norm of vector $\mathbf{x}$.

II. Preliminaries

A. Lattices

Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$, a full-rank lattice of dimension n is defined by

$$\Lambda = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} \mathbf{b}_i x_i \mid x_i \in \mathbb{Z} \right\}.$$

The set $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is called a lattice basis, and it is usually represented as a matrix $\mathbf{B}$ with the basis vectors $\mathbf{b}_i$ as columns.

The Gram-Schmidt vectors $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ of $\mathbf{B}$ are found by using $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$, where $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / ||\mathbf{b}_j^*||^2$ and $\mathbf{b}_1^* = \mathbf{b}_1$.

The ith successive minimum of an n dimensional lattice $\mathcal{L}(\mathbf{B})$ is the smallest real positive number r such that Λ contains i linearly independent vectors of length at most r:

$$\lambda_i(\mathbf{B}) = \inf \{ r \mid \dim(\mathrm{span}((\Lambda \cap \mathcal{B}(\mathbf{0}, r))) \geq i \},$$

in which $\mathcal{B}(\mathbf{0}, r)$ denotes a ball centered at $\mathbf{0}$ with radius r.

The process of improving the quality of a given basis by using a unimodular matrix from the general linear group $\mathrm{GL}_n(\mathbb{Z})$ is called lattice reduction. We introduce a metric termed 'basis length' that quantifies how good a reduced basis is:

$$l(\mathbf{B}) = \max_i \|\mathbf{b}_i\|. \tag{1}$$

The shortest basis problem (SBP) is closely related to basis length. The problem is to find, when given a lattice basis, a unimodular matrix that makes the longest basis vector short:

$$\min_{\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})} l(\mathbf{B} \times \mathbf{U}). \tag{2}$$

Another metric termed orthogonality defect defined by

$$\eta(\mathbf{B}) = \frac{\prod_{i=1}^{n} ||\mathbf{b}_i||}{\sqrt{|\det(\mathbf{B}^\top \mathbf{B})|}} \tag{3}$$

can quantify the overall goodness of a basis, since it reflects whether we have made all the basis vectors short.

Finally, the closest vector problem (CVP) is, given a vector $\mathbf{y} \in \mathbb{R}^n$ and a lattice $\mathcal{L}(\mathbf{B})$, find a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that:

$$\|\mathbf{y} - \mathbf{v}\|^2 \leq \|\mathbf{y} - \mathbf{w}\|^2, \quad \forall \mathbf{w} \in \mathcal{L}(\mathbf{B}).$$

B. Sequential Reduction

The principle of sequential reduction is to reduce a basis vector by using all other vectors that span a sublattice. Since we are interested in solving SBP, we present a degraded version of sequential reduction in this work, *which only reduces the longest vector.* To be concise, given a set of basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, we sort these vectors such that their lengths are in ascending order. Then it goes into a while loop which checks whether the longest vector can be made shorter by using a sublattice spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$. The algorithm terminates if no more reduction can be made on $\mathbf{b}_n$. The pseudocodes of this type of reduction are shown in Algorithm 1.

Algorithm 1 The Degraded Form of an SR Algorithm

Input: Lattice basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$.
Output: Reduced lattice basis $\mathbf{B}$.

```
1 while TRUE do
2     Sort b_1, b_2, ..., b_n such that
      ||b_1||^2 ≤ ||b_2||^2 ... ≤ ||b_n||^2;
3     Let b_n be the query point, and use an
      exact/approximate CVP algorithm to find the closest
      vector (noted as s_n) in basis b_1, ..., b_{n-1};
4     if ||b_n − s_n||^2 < ||b_n||^2 then
5         b_n ← b_n − s_n;
6     else
7         Break;
```

III. The Proposed SR-SIC for SBP

For the sake of solving SBP effectively, we resort to the celebrated SIC algorithm in this section, fully leveraging the fact that the subroutine in Line 3 of Algorithm 1 can be chosen from any exact/approximate CVP algorithm.

A. The SIC Subroutine

We first review how SIC works. Given a query point $\mathbf{y}$ and a lattice basis $\tilde{\mathbf{B}}$ of rank k (*e.g.*, $\mathbf{b}_1, \ldots, \mathbf{b}_k$), the corresponding CVP is to solve $\min_{\mathbf{x} \in \mathbb{Z}^k} \left\| \mathbf{y} - \tilde{\mathbf{B}}\mathbf{x} \right\|^2$. We perform a QR factorization to get $\tilde{\mathbf{B}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$, where $\tilde{\mathbf{Q}}$ is unitary and $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times k}$ is upper triangular. Without loss of generality, we assume that $\tilde{r}_{1,1} > 0, \tilde{r}_{2,2} > 0, \ldots, \tilde{r}_{k,k} > 0$. Define $\tilde{\mathbf{y}} = \tilde{\mathbf{Q}}^\top \mathbf{y} \in \mathbb{R}^n$, since $\tilde{\mathbf{Q}}$ is unitary, the following two forms of solving a CVP are equivalent:

$$\min_{\mathbf{x} \in \mathbb{Z}^k} \left\| \mathbf{y} - \tilde{\mathbf{B}}\mathbf{x} \right\|^2 = \min_{\mathbf{x} \in \mathbb{Z}^k} \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{R}}\mathbf{x} \right\|^2. \tag{4}$$

Not enumerating the exact solution of (4), SIC estimates the symbols of the signal vector $\mathbf{x}$ recursively, thus canceling the

effect of those symbols already decoded and nulling those yet unknown. If a symbol $\hat{x}_k$ has been estimated, the decoder will exploit this decision to further estimate the remaining symbols $\hat{x}_{k-1}, \ldots, \hat{x}_1$, forming a nonlinear decoding structure. To be concise, it estimates the integer coefficients by using:

$$\hat{x}_k = \lfloor \frac{\tilde{y}_k}{\tilde{r}_{k,k}} \rceil, \tag{5}$$

$$\hat{x}_m = \lfloor \frac{\tilde{y}_m - \sum_{l=m+1}^{n} \tilde{r}_{m,l}\hat{x}_l}{\tilde{r}_{m,m}} \rceil \tag{6}$$

for $m = k-1, \ldots, 1$.

Given the above $\hat{\mathbf{x}}$ in Eqs. (5) and (6), the estimated lattice point is written as $\tilde{\mathbf{B}}\hat{\mathbf{x}}$. The SIC subroutine is summarized in Algorithm 2, and the resulted Algorithm 1 is referred to as SR-SIC.

Algorithm 2 The SIC Subroutine in SR

Input: A query point $\mathbf{y}$, and the lattice basis $\tilde{\mathbf{B}} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$.

Output: The closest lattice point to $\mathbf{y}$.

1 Perform QR factorization on $\tilde{\mathbf{B}}$ to obtain $\tilde{\mathbf{B}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$;
2 $\tilde{\mathbf{y}} = \tilde{\mathbf{Q}}^\top \mathbf{y}$;
3 $\hat{x}_k = \lfloor \frac{\tilde{y}_k}{\tilde{r}_{k,k}} \rceil$;
4 **for** $m = k-1, \ldots, 1$ **do**
5 $\quad \hat{x}_m = \lfloor \frac{\tilde{y}_m - \sum_{l=m+1}^{n} \tilde{r}_{m,l}\hat{x}_l}{\tilde{r}_{m,m}} \rceil$;
6 Return $\tilde{\mathbf{B}}\hat{\mathbf{x}}$.

To employ SIC in Line 3 of Algorithm 1, we set $k = n-1$, $\tilde{\mathbf{B}} = [\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}]$, and $\mathbf{y} = \mathbf{b}_n$ for Algorithm 2. Thus, the estimated closest lattice point to the query point $\mathbf{b}_n$ is

$$\mathbf{s}_n = \tilde{\mathbf{B}}\hat{\mathbf{x}}. \tag{7}$$

It is noteworthy that $\mathbf{y}$ is not inside the space spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$, so we can equivalently project $\mathbf{y}$ onto this linear space and then find its closest lattice point in $\mathcal{L}(\tilde{\mathbf{B}})$.

B. Performance Bounds

The following theorem provides upper bounds for the basis length $l(\mathbf{B})$ and the orthogonality defect $\eta(\mathbf{B})$.

Theorem 1: Given any lattice basis in dimensions $n \leq 4$ as input, the SR-SIC reduced basis satisfies:

$$l(\mathbf{B}) \leq \frac{2}{(5-n)^{1/2}} \lambda_n(\mathbf{B}), \tag{8}$$

$$\eta(\mathbf{B}) \leq \frac{2^{2n-1}}{(5-n)^{n/2}} \frac{\lambda_n^n(\mathbf{B})}{\lambda_1^n(\mathbf{B})}, \tag{9}$$

where λ_1 and λ_n respectively denotes the 1st and nth successive minimum of lattice $\mathcal{L}(\mathbf{B})$.

Proof: Since the last vector is the longest, we have

$$\begin{aligned} \|\mathbf{b}_1\|^2 &\leq \|\mathbf{b}_n\|^2, \\ \|\mathbf{b}_2\|^2 &\leq \|\mathbf{b}_n\|^2, \\ &\vdots \\ \|\mathbf{b}_{n-1}\|^2 &\leq \|\mathbf{b}_n\|^2. \end{aligned} \tag{10}$$

As an SIC algorithm with query point $\mathbf{b}_n$ over basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}\}$ can no longer make $\|\mathbf{b}_n\|^2$ smaller, we have

$$\begin{aligned} \underbrace{\|\mathbf{b}_n\|^2}_{\text{term 1}} &\leq \|\mathbf{b}_n - \mathbf{s}_n\|^2 \\ &= \underbrace{\|\mathbf{b}_n^*\|^2}_{\text{term 2}} + \underbrace{\|\mathbf{b}_n - \mathbf{b}_n^* - \mathbf{s}_n\|^2}_{\text{term 3}}, \end{aligned} \tag{11}$$

where the second equality is due to Pythagoras' theorem.

We then show upper bounds for terms 2 and 3 in (11), respectively. The residue distance of SIC is upper bounded by the lengths of Gram-Schmidt vectors, so we have for term 3 that

$$\begin{aligned} \|\mathbf{b}_n - \mathbf{b}_n^* - \mathbf{s}_n\|^2 &\leq \frac{1}{4}\sum_{j=1}^{n-1} |\tilde{r}_{j,n}|^2 \\ &\leq \frac{1}{4}\sum_{j=1}^{n-1} \|\mathbf{b}_j\|^2. \end{aligned} \tag{12}$$

Regarding term 2, we deploy a QR decomposition to get $\mathbf{B} = \mathbf{QR}$, from which we obtain $\|\mathbf{b}_n^*\|^2 = |r_{n,n}|^2$. We use the same technique as that in [13] to proceed: No matter how worse the basis quality of $\mathbf{B}$ or $\mathbf{R}$ is, there exists one coefficient vector $\bar{\mathbf{x}} \in \mathbb{Z}^n$ and $\bar{x}_n \neq 0$ such that

$$|r_{n,n}\bar{x}_n|^2 + |r_{n-1,n-1}\bar{x}_{n-1} + r_{n-1,n}\bar{x}_n|^2 + \underbrace{\ldots}_{\geq 0} \leq \lambda_n^2(\mathbf{B}). \tag{13}$$

So we have

$$\|\mathbf{b}_n^*\|^2 = |r_{n,n}|^2 \leq |r_{n,n}\bar{x}_n|^2 \leq \lambda_n^2(\mathbf{B}). \tag{14}$$

Substituting (12) and (14) into (11), we have

$$\|\mathbf{b}_n\|^2 \leq \lambda_n^2(\mathbf{B}) + \frac{1}{4}\sum_{j=1}^{n-1} \|\mathbf{b}_j\|^2. \tag{15}$$

Now we reach upper bounds for $\|\mathbf{b}_1\|^2, \ldots, \|\mathbf{b}_n\|^2$. It follows from plugging (10) into (15) that

$$\|\mathbf{b}_n\|^2 \leq \lambda_n^2(\mathbf{B}) + \frac{n-1}{4}\|\mathbf{b}_n\|^2. \tag{16}$$

We can therefore obtain non-trivial result when $n \leq 4$:

$$\|\mathbf{b}_n\|^2 \leq \frac{4}{5-n}\lambda_n^2(\mathbf{B}). \tag{17}$$

$$l(\mathbf{B}) = \|\mathbf{b}_n\| \leq \frac{2}{\sqrt{5-n}}\lambda_n(\mathbf{B}). \tag{18}$$

Regarding the orthogonal defect, the product of vectors' lengths can be bounded based on using the arithmetic mean-geometric mean inequality, *i.e.*,

$$\prod_{i=1}^{n} \|\mathbf{b}_i\| \leq \left(\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{b}_i\|^2\right)^{n/2} \leq \left(\frac{4}{5-n}\right)^{n/2} \lambda_n^n(\mathbf{B}). \tag{19}$$

In addition, $\sqrt{|\det(\mathbf{B}^\top\mathbf{B})|}$ is larger than the volume of a ball with radius $\lambda_1(\mathbf{B})/2$, then for $n \leq 4$,

$$\sqrt{|\det(\mathbf{B}^\top\mathbf{B})|} \geq V_n \times \left(\frac{\lambda_1(\mathbf{B})}{2}\right)^n \geq 2\left(\frac{\lambda_1(\mathbf{B})}{2}\right)^n, \tag{20}$$

TABLE I
UPPER BOUNDS ON BASIS LENGTH $l(\mathbf{B})$

Algorithms	This work: SR-SIC	LLL [6]	boosted LLL [11]	KZ [18]	boosted KZ [11]	Minkowski [18]
Dimension 3	$1.414\lambda_3(\mathbf{B})$	$4\lambda_3(\mathbf{B})$	$4\lambda_3(\mathbf{B})$	$1.225\lambda_3(\mathbf{B})$	$\lambda_3(\mathbf{B})$	$\lambda_3(\mathbf{B})$
Dimension 4	$2\lambda_4(\mathbf{B})$	$8\lambda_4(\mathbf{B})$	$8\lambda_4(\mathbf{B})$	$1.323\lambda_4(\mathbf{B})$	$\lambda_4(\mathbf{B})$	$\lambda_4(\mathbf{B})$

where V_n refers to the volume of an n-dimensional unit ball with $V_1 = 2$, $V_2 \approx 3.142$, $V_3 \approx 4.189$, $V_4 \approx 4.935$. Combining (19) and (20), we obtain (9). □

We compare the bounds of SR-SIC with those of major algorithms in Table I. The effective subroutines of SR-SIC, LLL and boosted LLL all enjoy polynomial-time complexity, while those of KZ, boosted KZ, and Minkowski have exponential-time complexity [11], [18]. We can observe from Table I that SR-SIC yields the best upper bounds among the polynomial-time algorithms, with $l(\mathbf{B}) \leq 1.414\lambda_3(\mathbf{B})$ for dimension 3 and $l(\mathbf{B}) \leq 2\lambda_4(\mathbf{B})$ for dimension 4.

Remark 1: To extend Theorem 1 for dimensions larger than 4, we shall revise the algorithm to achieve sharper bounds in (10). Hereby the loose bounds in (10) is due to the fact that all basis vectors are only sorted in an ascending order, which enjoys the advantage of low-complexity. In essence, our SR-SIC algorithm is using a simple length-sorting condition to replace the Lovász condition of LLL.

Remark 2: We observe from Theorem 1 that (8) outperforms [13, Thm. 2, Eq. (8)] which has an additional $\sqrt{n}$ factor. The improved analysis in this letter is, actually, due to targeting the longest vector when showing bounds on terms 2 and 3. An interesting observation is that the low-complexity SR-SIC can reach the same bound on $l(\mathbf{B})$ that SR-CVP promises.

C. Complexity

First, in the SIC subroutine, the computational complexity of the QR factorization is $O(n^3)$, and that of completing n successive cancellation and quantization is in the order of $O(n^2)$. Second, regarding sorting the lengths of the basis vectors, a full sorting only needs to be implemented for the first time, whose worst case complexity is $O(n^2)$. After that, we only need to insert the newly computed vector into a position, and its complexity is $O(n)$. Therefore, with N_{iter} loops in Algorithm 1, the computational complexity of SR-SIC is

$$N_{iter} \times O(n^3). \tag{21}$$

Define $\delta = \min_{\mathbf{b}_n, \mathbf{s}_n}\{||\mathbf{b}_n||^2 - ||\mathbf{b}_n - \mathbf{s}_n||^2\}$ (note that $\delta = 1$ for integer lattices). If the while loop in Algorithm 1 continues, then $||\mathbf{b}_n||^2$ is decreasing at least by δ in each iteration. Thus we have

$$N_{iter} \leq \sum_{i=1}^{n} \left(||\mathbf{b}_i||^2 - \lambda_i^2(\mathbf{B})\right)/\delta.$$

We argue that obtaining an explicit upper bound for N_{iter} is hard, although simple bounds exist when we have the input distribution of entries in $\mathbf{b}_i$, together with using Gaussian Heuristics [19] for $||\lambda_i||$.

IV. SIMULATIONS

This section examines the performance and complexity of the proposed SR-SIC algorithm. Major benchmark algorithms considered are: LLL [6], boosted LLL [11], and Minkowski reduction [18] that provides the length upper bound of SBP. To foster reproducible research, all programs in the simulations are of open source and freely available at GitHub.[1]

A. Basis Quality in Small Dimensions

While lattice reduction algorithms have very similar performances in small dimensions, there exist examples in small dimensions that SR-SIC performs better than LLL. In dimension 3, consider

$$\widetilde{\mathbf{B}} = \begin{bmatrix} 1 & 0.4 & 0 \\ 0 & 1 & 0.52 \\ 0 & 0 & 1 \end{bmatrix}.$$

The reduced bases of different algorithms have lengths

$$\begin{aligned} &||\mathbf{b}_1||^2, ||\mathbf{b}_2||^2, ||\mathbf{b}_3||^2 \\ &= 1, 1.16, 1.3904\,(\text{LLL}) \\ &\quad 1, 1.16, 1.2704\,(\text{SR-SIC}, \text{BoostedLLL}, \text{Minkowski}) \end{aligned}$$

In dimension 4, consider

$$\widetilde{\mathbf{B}} = \begin{bmatrix} 2 & 0.4 & 0.4 & 0.4 \\ 0 & 1 & 0.4 & 0 \\ 0 & 0 & 1 & 0.52 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The reduced bases of different algorithms have lengths

$$\begin{aligned} &||\mathbf{b}_1||^2, ||\mathbf{b}_2||^2, ||\mathbf{b}_3||^2, ||\mathbf{b}_4||^2 \\ &= 0.64, 1.01, 1.17, 1.4004\,(\text{LLL}) \\ &\quad 0.64, 1.01, 1.17, 1.3104\,(\text{SR-SIC}, \text{BoostedLLL}, \\ &\quad \text{Minkowski}) \end{aligned}$$

As Boosted LLL and Minkowski reduction feature higher complexity, the examples show that SR-SIC is a better low-complexity alternative of solving SBP.

B. Application to Integer-Forcing

As an application, we consider a real-valued Multiple-Input Multiple-Output (MIMO) channel model. Letting the channel matrix $\mathbf{H}$ be random, the actual achievable sum rate of the integer-forcing linear receivers can be quantitatively evaluated by the ergodic rate R_E [4], [5],

$$R_E \triangleq \mathbb{E}_{\mathbf{H}}\left(\max_{\substack{\mathbf{C}\in\mathbb{Z}^{n\times n},\\ \text{rank}(\mathbf{C})=n}} \min_{i\in\{1,\ldots,n\}} \frac{n}{2}\log_2\left(\frac{\text{SNR}}{||\bar{\mathbf{B}}\mathbf{c}_i||^2}\right)\right), \tag{22}$$

[1] https://github.com/shx-lyu/boosted-lll

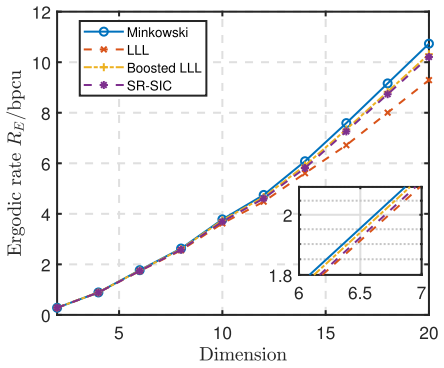

Fig. 1. The ergodic rates of different reduction algorithms in integer-forcing.

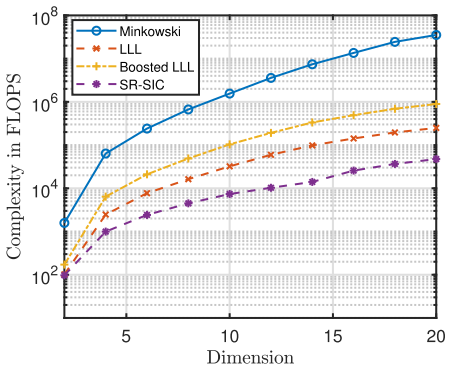

Fig. 2. The complexity in FLOPS of different algorithms.

where $\mathbf{H}^\top\mathbf{H}+1/\mathrm{SNR}\mathbf{I} = \mathbf{V}\Sigma\mathbf{V}^\top$ is the eigen-decomposition, SNR refers to the signal to noise ratio (SNR), $\bar{\mathbf{B}} \triangleq \Sigma^{-1/2}\mathbf{V}^\top$, and $\mathbf{C} \triangleq [\mathbf{c}_1, \ldots, \mathbf{c}_n]$. Considering a typical model for MIMO communication, hereby we assume that entries of $\mathbf{H}$ are i.i.d. standard normal $\mathcal{N}(0,1)$ distributed, and apply lattice reduction algorithms to reduce the basis $\bar{\mathbf{B}}$.

In Fig. 1, we plot the integer-forcing rate performance of different lattice reduction algorithms in dimensions 2 to 20 at $\mathrm{SNR} = -5\mathrm{dB}$. It reveals that both SR-SIC and Boosted LLL attain rates that are close to the upper bounds provided by Minkowski reduction, while LLL suffers a performance gap especially when the dimensions become large than 10.

C. Complexity

We further study the computational complexity of the reduction algorithms by counting the number of floating-point operations (FLOPS). With the same settings as those in Fig. 1, we plot the dimension versus FLOPS relationships in Fig. 2. It shows that SR-SIC has the lowest complexity among the four algorithms, and it is around 10 times faster than LLL in the dimension range of 10 to 20.

V. Conclusion

To summarize, we have proposed a low-complexity yet provable version of sequential reduction for solving SBP. The algorithm only reduces the longest vector. For a small dimensional reduced basis, the longest vector is shown to be no longer than $\frac{2}{\sqrt{5-n}}\lambda_n(\mathbf{B})$. Considering SBP, the simulation results show that the proposed SR-SIC outperforms LLL without the need of a complexity-performance tradeoff.

References

[1] C. Ling and J.-C. Belfiore, "Achieving AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5918–5929, Oct. 2014.
[2] L. Liu, Y. Yan, C. Ling, and X. Wu, "Construction of capacity-achieving lattice codes: Polar lattices," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 915–928, Feb. 2019.
[3] R. F. H. Fischer, S. Stern, and J. B. Huber, "Lattice-reduction-aided and integer-forcing equalization: Structures, criteria, factorization, and coding," *Found. Trends Commun. Inf. Theory*, vol. 16, nos. 1–2, pp. 1–155, 2019.
[4] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7661–7685, Dec. 2014.
[5] A. Sakzad, J. Harshan, and E. Viterbo, "Integer-forcing MIMO linear receivers based on lattice reduction," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4905–4915, Oct. 2013.
[6] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, Dec. 1982.
[7] X.-W. Chang, X. Yang, and T. Zhou, "MLAMBDA: A modified LAMBDA method for integer least-squares estimation," *J. Geodesy*, vol. 79, no. 9, pp. 552–565, Dec. 2005.
[8] Q. Wen and X. Ma, "Efficient greedy LLL algorithms for lattice decoding," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3560–3572, May 2016.
[9] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, nos. 1–3, pp. 181–199, Aug. 1994.
[10] C. Ling, W. H. Mow, and N. Howgrave-Graham, "Reduced and fixed-complexity variants of the LLL algorithm for communications," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1040–1050, Mar. 2013.
[11] S. Lyu and C. Ling, "Boosted KZ and LLL algorithms," *IEEE Trans. Signal Process.*, vol. 65, no. 18, pp. 4784–4796, Sep. 2017.
[12] M. Seysen, "Simultaneous reduction of a lattice basis and its reciprocal basis," *Combinatorica*, vol. 13, no. 3, pp. 363–376, Sep. 1993.
[13] S. Lyu, J. Wen, J. Weng, and C. Ling, "On low-complexity lattice reduction algorithms for large-scale MIMO detection: The blessing of sequential reduction," *IEEE Trans. Signal Process.*, vol. 68, pp. 257–269, 2020.
[14] Q. Zhou and X. Ma, "Element-based lattice reduction algorithms for large MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 274–286, Feb. 2013.
[15] P. Q. Nguyen and D. Stehlé, "Low-dimensional lattice basis reduction revisited," *ACM Trans. Algorithms*, vol. 5, no. 4, p. 46, 2009.
[16] C. Ling, W. H. Mow, and L. Gan, "Dual-lattice ordering and partial lattice reduction for SIC-based MIMO detection," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 6, pp. 975–985, Dec. 2009.
[17] L. Babai, "On Lovász'lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.
[18] W. Zhang, S. Qiao, and Y. Wei, "HKZ and minkowski reduction algorithms for Lattice-Reduction-Aided MIMO detection," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5963–5976, Nov. 2012.
[19] L. Luzzi, D. Stehle, and C. Ling, "Decoding by embedding: Correct decoding radius and DMT optimality," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2960–2973, May 2013.