Universal Lattice Decoding: a Review and Some Recent Results

Wai Ho Mow
Department of EEE
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong S.A.R., CHINA
Email: w.mow@ieee.org

Abstract— The idea of formulating the detection of a lattice-type modulation transmitted over a linear channel as the so-called universal lattice decoding problem dates back to at least the early 1990s. The principle of universal lattice decoding can trace its roots back to the theory developed for solving the shortest/closest lattice vector problem. In this paper, such a principle and its applications in communications are reviewed. In addition, it will be shown that with some lattice preprocessing steps, impressive performance improvement and/or complexity reduction of some well-known detectors (e.g. ZF, DFE, and VBLAST) can be achieved.

I. Introduction

The idea of formulating the detection of a lattice-type modulation transmitted over a linear channel as the so-called universal lattice decoding problem dates back to at least the early 1990s [1]. The resultant universal lattice decoder is very attractive for bandwidth efficent modulations, such as M-PAM and M-QAM, due to its desirable properties [1], [2], such as:

- its decoding complexity is independent of the modulation alphabet size M;
- its performance is nearly optimal, especially for large M;
- its average complexity is quadratic, as the signal-to-noise ratio tends to infinity.

The applications of such decoders have proliferated due to the growing importance of many linear channel models, such as intersymbol interference channels [1], [2], fading channels [3], uncoded and space-time block coded multiple-antenna channels [4], multiuser CDMA channels [5], dispersive multiple-antenna channels [6] and their combinations.

The principle of universal lattice decoding can trace its roots back to the theory and algorithms developed for solving the shortest/closest lattice vector problem for integer programming and cryptoanalysis applications. The closest (lattice) vector problem (CVP) is a class of nearest neighbor searches or closest-point queries, in which the solution set to be searched consists of all the points in a lattice. A general solution for the CVP was proposed by Kannan [7]. Although his algorithm does not lead to an efficient practical solution, the underlying ideas are simple and powerful, consisting of two steps:

Step 1: For the given lattice, find a "short" and fairly "orthogonal" basis, called the reduced basis.

Step 2: Enumerate all lattice points falling inside a certain sphere centered at the query point so as to identify the closest lattice point.

The procedure that transforms a lattice basis into a reduced one is called the basis reduction algorithm, while the one achieving the second step is called the enumeration algorithm. In cryptoanalysis, it is well-known that choosing a good lattice basis is important to finding a good nearby lattice point, which can in turn greatly speed up the closest lattice point search.

The rest of the paper is organized as follows. After introducing the notation and preliminaries on lattices in Section 2, the main result on lattice basis reduction will be described in Section 3. Two algorithms for finding a nearby lattice point and their relationships with some well-known detectors will be discussed in Section 4. Extension of conventional lattice algorithms for complex lattices is described in Section 5. Section 6 introduces enumeration algorithms that find the closest lattice point and their efficient implementations. Section 7 compares the simulated performance and complexities of various presented lattice algoirhtms in a typical communication application. Finally, Section 8 contains the conclusion.

II. Notation and Preliminaries on Lattices

Let m and n be two positive integers with $n \leq m$. A subset L of $\mathbf{R}^m$ is called a lattice of dimension n if there exist n linearly independent m-dimensional vectors $b_1, \cdots, b_n \in \mathbf{R}^m$ such that

$$L = L(B) = \{\eta_1 b_1 + \cdots + \eta_n b_n : \eta_i \in \mathbf{Z}\},$$

where $B = [b_1, \cdots, b_n]$ is a $m \times n$ matrix. The set of column vectors $b_1, \cdots, b_n$ and the matrix B are said to be the basis and the basis matrix of L, respectively.

Let us consider the Gram-Schmidt orthogonalization of a given lattice basis. The m-dimensional orthogonalization vectors $b_1^*, \cdots, b_n^*$ and the real numbers μ_{ij}, for $1 \leq j < i \leq n$, are defined recursively by

$$b_i^* = b_i - \textstyle\sum_{j=1}^{i-1} \mu_{ij} b_j^*, \tag{1}$$

$$\mu_{ij} = (b_j^*)^T b_i / \|b_j^*\|^2, \tag{2}$$

where $(\cdot)^T$ denotes the matrix transpose of $(\cdot)$. The orthogonal vectors $b_1^*, \cdots, b_n^*$ obtained in this way depend on the ordering of $b_1, \cdots, b_n$. Note that, for $1 \leq i \leq n$, $b_1^*, \cdots, b_i^*$ and $b_1, \cdots, b_i$ span the same vector subspace. By defining $\mu_{ii} = 1$, we have

$$b_i = \textstyle\sum_{j=1}^{i} \mu_{ij} b_j^*, \tag{3}$$

for $1 \leq i \leq n$. Expressing (3) in matrix notation,

$$B = B^*[\mu_{ij}]^T, \tag{4}$$

where $B^* = [b_1^*, \cdots, b_n^*]$ and $[\mu_{ij}]$ is a $n \times n$ lower triangular matrix with all diagonal elements equal to 1.

By letting $u_i = b_i^*/\|b_i^*\|$ and $b_i(j) = \mu_{ij}\|b_j^*\|$, for $1 \leq j \leq i \leq n$, we also have

$$b_i = \textstyle\sum_{j=1}^{i} b_i(j) u_j. \tag{5}$$

[1]This work was supported by the Hong Kong RGC Grant No. HKUST6246/02E.

[2]For a full version of this paper, please refer to [18].

Clearly, $b_i(i) = \|b_i^*\|$ for all i. Expressing (5) in matrix notation,

$$B = U[b_i(j)]^T \tag{6}$$

is in fact the QR factorization of B, where $U = [u_1, \cdots, u_n]$ gives an orthonormal basis for the column space of B. We note that the upper triangular matrix $[b_i(j)]^T$ is actually the Cholesky factor of $B^T B$ satisfying

$$B^T B = [b_i(j)][b_i(j)]^T. \tag{7}$$

The latter also follows immediately from (6).

III. Lattice Basis Reduction

Lattice basis reduction is naturally associated with the problem of finding the shortest non-zero lattice vector — the shortest vector problem (SVP). The L_∞-norm SVP is known to be NP-hard, although whether the Euclidean norm SVP is NP-hard or not is still open. In 1982, Lenstra, Lenstra and Lovász [8] (LLL) achieved a breakthrough by constructing the celebrated LLL reduction algorithm, which can produce the so-called LLL-reduced basis from any given lattice basis in polynomial time and thereby approximating the shortest (non-zero) lattice vector up to a factor of $2^{(n-1)/2}$. Based on their algorithm, more efficient algorithms for solving the SVP and the CVP have been developed. In fact, their algorithm has given rise to efficient solutions to a variety of research problems, including attacks on knapsack-based crypto-systems [9], and the disproof of Mertens' century-old conjecture in number theory. All these applications were made possible by the LLL-reduction algorithm. References [8] and [10] have described the LLL-reduction algorithm with proof of its correctess and polynomial complexity. In the following, we shall rederive the reduction algorithm based on our own interpretation.

A. Size Reduction

Consider the lower triangular representation $[b_i(j)]$ of a given basis $b_1, \cdots, b_n$. If, for $j < i$, the j-th coordinate of b_i is greater in magnitude than half of the length of b_j^* (i.e. $|b_i(j)| > \frac{b_j(j)}{2}$), we can always reduce the length of b_i by projecting it onto the subspace spanned by $b_1^*, \cdots, b_j^*$ and then lifting it. The resultant vector $\bar{b}_i = b_i - \eta b_j$, for some integer η, must satisfy the condition that $|\bar{b}_i(j)| \le \frac{b_j(j)}{2}$. The process can be repeated $\frac{(n-1)(n-2)}{2}$ times for all $1 < j < i \le n$. The resultant basis consisting of shorter basis vectors is said to be size-reduced. In summary, a basis $b_1, \cdots, b_n$ is size-reduced if $|\mu_{ij}| \le \frac{1}{2}$ for $1 < j < i \le n$. Recall that $\mu_{ii} = 1$ and $\mu_{ij} = 0$ for $j > i$. Finally, we remark that $b_1^*, \cdots, b_n^*$ obtained from the orthogonalization process of the new basis is the same as before.

B. LLL Reduction

Given a basis and a specific orthogonalization $b_1^*, \cdots, b_n^*$, we can always size-reduce it into a "better" basis. One may then ask how to find a "better" orthogonalization of a given basis.

For any orthogonalization of a given lattice, the product of the lengths of b_i^*'s must be a constant. Intuitively, it is desirable that the lengths of b_i^*'s are distributed as even as possible so that the basis, after size reduction, appears to be "shorter". Lovász [10] observed that typically the shorter vectors among $b_1^*, \cdots, b_n^*$ are at the end of the sequence. So it is desirable to make the orthogonalization sequence $b_1(1) = \|b_1^*\|, \cdots, b_n(n) = \|b_n^*\|$ lexicographically as small as possible.

For $1 \le j \le i \le n$, define $b(i,j)$ as the projection of b_i onto the orthogonal complement of the subspace spanned by $u_1, \cdots, u_{j-1}$, or mathematically, $b(i,j) = \sum_{k=j}^{i} b_i(k) u_k$. For some $i < n$, consider the lengths of the projections of b_i and b_{i+1} onto $u_i, \cdots, u_n$, i.e., $b(i,i) = b_i(i)u_i$ and $b(i+1,i) = b_{i+1}(i)u_i + b_{i+1}(i+1)u_{i+1}$. If $b(i,i)$ is longer than $b(i+1,i)$ (i.e., $b_i(i) > \|b(i+1,i)\|$), we can always swap b_i and b_{i+1} to get a lexicographically smaller orthogonalization sequence $b_1(1), \cdots, b_{i-1}(i-1), \|b(i+1,i)\|, \cdots, b_n(n)$. Hence a better orthogonalization is resulted.

After swapping some basis vectors, we can further size reduce the basis without changing the orthogonalization sequence. The two processes, namely, finding a better basis via size reduction for a given orthogonalization sequence and finding a better orthogonalization sequence via swapping basis vectors for a given basis, can be iterated until no further improvement is achievable. This is in essence the LLL-reduction algorithm in its most preliminary form.

Algorithm LLL_Reduction($b_1, \cdots, b_n$)

Step 1: Size-reduce the given basis.

Step 2: Check if there exists any i such that $\delta \cdot \|b(i,i)\|^2 > \|b(i+1,i)\|^2$. If found, swap b_i and b_{i+1}, update the orthogonalization sequence, and go to step 1. Otherwise, stop.

Here $\delta \le 1$ is set to achieve faster convergence.

IV. Finding a Nearby Lattice Point

Babai [11] proved that given a LLL-reduced basis, a nearby lattice point that is closest, within a factor exponential in n, to the query point can be found by two simple polynomial-time algoirhtms, called Procedure Rounding Off and Procedure Nearest Plane, respectively.

Denote the query vector $q = B\vartheta = \vartheta_1 b_1 + \cdots + \vartheta_n b_n \in \mathbf{R}^m$, where $\vartheta \in \mathbf{R}^n$ is a n-dimensional column vector. Procedure Rounding Off simply finds a nearby lattice point by rounding ϑ_i's to their nearest integers, i.e.,

$$B \cdot \text{round}(\vartheta) = \text{round}(\vartheta_1) b_1 + \cdots + \text{round}(\vartheta_n) b_n.$$

We observe that the procedure is in fact algorithmically equivalent to the well-known zero-forcing (ZF) detector, which takes the received output vector q and detects the transmitted input vector as $\text{round}(B^\dagger q) = \text{round}(\vartheta)$, where $B^\dagger = (B^T B)^{-1} B^T$ is the pseudo-inverse of B with $m \ge n$ such that $B^\dagger B = I_n$.

We also note that Babai's Procedure Nearest_Plane [11] is equivalent to a specific type of decision feedback equalization (DFE) algorithms, called the nulling and cancellation detector. By adopting a detection order (i.e., $\eta_n, \cdots, \eta_1$, in our notation) corresponding to the descending order of signal-to-noise ratios (SNR) of different elements in a received vector, the latter becomes the well-known VBLAST detector [12].

Denote the $m \times n$ basis matrix of the lattice L' by $[b_1', \cdots, b_n'] = (B^\dagger)^T$, where $L' = L((B^\dagger)^T)$ is called the dual lattice of $L(B)$. The VBLAST detection order is achieved by ordering (or re-indexing) the basis vectors such that $\|b_n'\| \le \cdots \le \|b_1'\|$. In other words, the detection order ensures that the transmitted vector element corresponding to the shortest dual basis vector is to be detected first, and so on. It is interesting to note that the VBLAST detection ordering as a preprocessing step for finding a nearby lattice point actually coincides with a preprocessing step proposed by Fincke and Pohst [13] for

finding the closest lattice point. In fact, Fincke and Pohst also suggested that the dual lattice should be reduced first.

V. Extension to Complex Lattices

If the modulation scheme involves both in-phase and quadrature-phase components (such as QPSK and QAM), the resultant (complex) lattice is a complex integer linear combination of some complex-valued basis vectors. Mathematically, a n-dimensional complex lattice $L \subset \mathbf{C}^m$ is defined as

$$L = \{\eta_1 b_1 + \cdots + \eta_n b_n : \eta_i \in \mathbf{Z} + \mathbf{Z}\sqrt{-1}\},$$

where $b_1, \cdots, b_n \in \mathbf{C}^m$ are n linearly independent m-dimensional complex basis vectors.

Yao and Wornell [14] extended the famous Gauss reduction algorithm for 2-dimensional complex lattices and demonstrated that the Gauss reduction step can significantly improves the performance of ZF and VBLAST detectors for a 2-transmit 2-receive antenna system with QAM. Note that Gauss redution is identical to the 2-D LLL reduction with $\delta = 1$.

Since the complex number field is well known to be an extension of the real number field, one may wonder how easy it is, if possible at all, to extend the general theory on lattices from the real to the complex case. In fact, the part of the theory reviewed up to here has already been extended in a careful manner in Sections II, III and IV. In particular, the results presented therein are valid, as long as the operations involved are replaced by their complex arithmetics counterparts. When interpreting the aforementioned results for complex lattices, there are a few points deserve special attention:

- $(\cdot)^T$ becomes the complex conjugate transpose operator.
- $|\cdot|$ gives the magnitude of the possibly complex-valued argument, and $|\cdot|^2$ should not be confused with $(\cdot)^2$.
- The real and imaginary parts returned by round$(\cdot)$ are the integers nearest to the real and imaginary parts of the possibly complex-valued argument respectively.
- In (2), the role of b_j* and b_i are not exchangeable in the complex case, unlike the real case.

Since an n-dimensional complex lattice is isomorphic to a $2n$-dimensional real lattice, every decoding problem with a complex lattice formulation can be re-formulated as a real lattice decoding problem. This approach was suggested in e.g. [1], [4] and has now become a standard approach. However, working directly on the complex lattice can result in decoding algorithms with lower complexity, because the exploitation of complex lattice structure allows the lattice dimension involved to be half of that of the equivalent real lattice. The complexity reduction is great especially for high-dimensional complex lattices. In general, conventional lattice algorithms and their complex counterparts are algorithmically inequivalent, even when applied to equivalent real and complex lattices.

VI. Finding the Closest Lattice Point

A straightforward method to find the closest lattice point is to enumerate all lattice points falling inside a sphere centered at the query point so as to identify the closest lattice point in the Euclidean metric. To avoid enumerating an unnecessarily large number of points, it is important to determine a reasonably small radius of the sphere.

A. Choice of Initial Radius

To avoid enumerating an unnecessarily large number of points, it is useful to determine a sufficiently small radius of the sphere which is sufficiently large to contain at least one lattice point. One suggestion from [5] is to use the Rogers upper bound on covering radius, which requires the knowledge about the dimension and determinant of the lattice. An alternative upper bound on the covering radius is $\frac{1}{2}\left(\sum_{k=1}^{n} \|b_k^*\|^2\right)^{1/2}$, which follows from proposition 4.2 in [7]. Although the values of $\|b_k^*\|$'s are required for calculating the bound, they are typically required by other lattice algorithms and do not induce additional complexity. In fact, the nearby lattice point returned by Babai's Procedure Nearest_Plane, called the Babai point in [16], always satisfies this bound. A better choice of the initial radius is $\|b - q\|$, where q and b denote the query point and the Babai point respectively.

B. Enumerating Lattice Points in a Sphere

Let $q = \vartheta_1 b_1 + \cdots + \vartheta_n b_n \in \mathbf{R}^m$ be the query point. If a lattice point $a = \eta_1 b_1 + \cdots + \eta_n b_n \in L(B)$ is inside the sphere of radius r centered at q, it satisfies the sphere constraint $\|a - q\| \leq r$. The enumeration problem is to determine all valid combinations of $\eta_1, \cdots, \eta_n$ under the sphere constraint, which can be expressed in terms of $b_1^*, \cdots, b_n^*$ as $\sum_{i=1}^{n}(\sum_{k=i}^{n}(\eta_k - \vartheta_k)\mu_{k,i})^2\|b_i^*\|^2 \leq r^2$. This suggests a recursive enumeration algorithm based on the following relationships:

$$r_n = r, \tag{8}$$

$$|\eta_n - \vartheta_n| \leq r_n/\|b_n^*\|, \tag{9}$$

and for $i = n-1, n-2, \cdots, 1$,

$$r_i = (r_{i+1}^2 - |\sum_{i+1}^{n}(\eta_k - \vartheta_k)\mu_{k,i+1}|^2\|b_i^*\|^2)^{1/2}, \tag{10}$$

$$|\eta_i + (\sum_{k=i+1}^{n}(\eta_k - \vartheta_i)\mu_{k,i})| \leq r_i/\|b_i^*\|. \tag{11}$$

The algorithm recursively divides an i-dimensional enumeration problem with radius r_i into $(\lfloor \frac{2r_i}{\|b_i^*\|} \rfloor + 1)$ $(i-1)$-dimensional similar problems with radii r_{i-1}'s. Eventually, the actual enumeration process occurs in many one-dimensional lattices. In the communications literature, the Pohst-Fincke enumeration algorithm without the LLL-reduction preprocessing is often called the sphere decoding algorithm.

C. Some Improvements

An improvement on the Pohst-Fincke enumeration algorithm can be obtained by updating the values of $r_1, \cdots, r_n$ with r' replacing r whenever a lattice point b with $r' = \|b - q\| < r$ is enumerated. As suggested by Mow in [1], to avoid an unnecessarily large amount of updating operations, we can enumerate the values of η_i from its mid-value to its upper bound and then from its mid-value to its lower bound, instead of from the lower bound to the upper bound. In this way, the short vectors are likely to be enumerated first. Based on a similar observation, Schnorr and Euchner [15] suggested an even better enumeration order, which enumerates the value of η_i in the order of increasing distance from the mid-value. As pointed out by Agrell et al. [16], the first lattice point enumerated using the Schnorr-Euchner ordering is the Babai point. Thus the choice of the initial radius is naturally set according to the Babai point, as suggested in Section VI-A, but without an explicit execution of the nearest plane algorithm and the associated complexity.

In [1] and [2], Mow suggested that the average complexity can be reduced by adding a simple stopping test for detecting early if the closest lattice point has been found. Let $\gamma = \gamma(L)$ denote the packing radius of the lattice L (i.e., half the length of the shortest lattice vector). If an enumerated lattice point is found to be at a distance less than γ from the query point (i.e., the packing radius stopping test is satisfied), it is clearly a nearest lattice point and thus the enumeration process can be terminated right away. In lattice decoding applications, the query point is a noisy version of a lattice point. At a sufficiently large SNR, most of the query points are located very close to the original lattice point. Therefore, a nearby lattice point, such as the Babai point, is likely to be the nearest point as well. It was suggested in [2] that the Babai point is first found by applying the nearest plane algorithm and if it passes the packing radius test, the whole enumeration process is skipped. As the computational cost of running the stopping test is very low, we apply the test whenever a new lattice point known to be currently the nearest is found.

Mow [2] argued that as the SNR tends to infinity, the average decoding complexity becomes quadratic for $m = n$ (or in general, $O(nm)$), namely, the complexity of the nearest plane algorithm with preprocessing. This result on the asymptotic average complexity of a universal lattice decoder is apparently consistent with the recent theoretical analysis of Hassibi and Vikalo [17] (see Figure 2 therein). The result remains valid for our implementation of the enumeration algorithm here, because the complexity of the Pohst-Fincke enumeration algorithm with the Schnorr-Euchner ordering up to the first enumerated lattice point (i.e., the Babai point) is also $O(nm)$, ignoring the preprocessing complexity.

Although the enumeration algorithm can always find a closest lattice point, it may sometimes return an invalid transmitted vector. As all practical modulation schemes have a finite symbol alphabet, the set of valid transmitted vectors are located inside a certain finite region of the infinite lattice. If the closest lattice point is outside the finite region, the so-called boundary error is resulted. As pointed out by Viterbo and Boutros [3], for important cubic-shaped modulation schemes (such as PAM and QAM), it is easy to incorporate the alphabet constraint by restricting the range of every vector component (i.e., η_i in our notation). This simple modification totally eliminates the occurrence of boundary errors leading to an exact MLD algorithm. The modified enumeration region is in general not spherical, and might be empty even if the initial radius is chosen according to the Babai point. In the latter case, it is necessary to restart the enumeration process with a larger initial radius chosen according to some heuristics. In addition, to enable the cubic-shaped alphabet constraint to be easily incorporated into the enumeration algorithm, the original lattice basis must be used. It means that the MLD modification is incompatible with the complexity reduction technique of applying the LLL reduction in the preprocessing phase.

VII. Simulation Experiments

In this section, the performance and complexity of various lattice decoding algorithms introduced in Sections IV to VI are evaluated and compared by computer simulation. The elements in the $m \times n$ basis matrix B are assumed to be independently and identically distributed complex Gaussian random variables with mean 0 and variance 1. It corresponds to the channel gain matrix in a typical multiple antenna communication system with n-transmit and m-receive antennas. We consider a 4-transmit 4-receive antenna system with 64-QAM whose alphabet is defined as $\mathbf{Z}_8 + \mathbf{Z}_8\sqrt{-1}$, without loss of generality. The system gives rise to a 4-dimensional complex lattice. As discussed in Section V, we may perform the complex LLL reduction (abbreviated as CLLL) directly on the complex lattice, or the conventional LLL reduction to an equivalent 8-dimensional lattice. The i-th element η_i of the transmitted input vector η represents a 64-QAM symbol to be transmitted by the i-th antenna. The query point (corresponding to the received output vector) is the transmitted input vector transformed by the channel matrix and corrupted by an additive noise vector w, or in symbols, $q = B\eta + w$. The elements of the noise vector w are independently and identically distributed white complex Gaussian random variables with mean 0 and variance 1, independent of the channel gain coefficients B_{ij}'s.

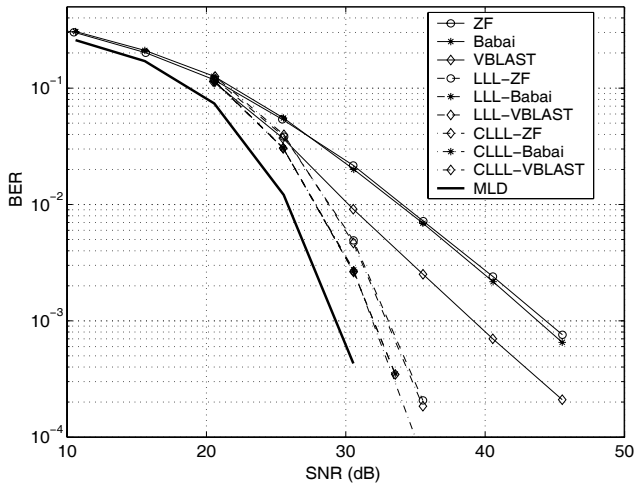

Fig. 1. Performance of the zero-forcing detector, the Babai nearest plane algorithm, and the VBLAST detector, with and without the real or complex LLL reduction preprocessing, as well as the MLD in a 4-transmit 4-receive antenna system with 64-QAM.

Figure 1 shows the BER performance of the ZF detector, the Babai nearest plane algorithm and the VBLAST detector with and without applying LLL reduction of B as a preprocessing step, as well as the 3 complex lattice based detectors (i.e., CLLL-ZF, CLLL-Babai and CLLL-VBLAST). The MLD performance is also shown therein. For the ease of comparison, we shall take the performance of the Babai nearest plane algorithm at a BER of 10^{-3} as the reference point. The ZF detector performs only a fraction of a dB worse. The VBLAST detector is in fact the nearest plane algorithm with the basis ordering preprocessing (c.f. Section IV). It can be observed that the basis ordering can provide almost 5dB gain. With the LLL reduction preprocessing, the three detectors (i.e., LLL-ZF, LLL-Babai and LLL-VBLAST) have similar performance that offers about 12dB gain. It implies that the LLL reduction can provide an additional 7dB gain over the VBLAST basis ordering as a preprocessing step. Note that the LLL-VBLAST does not perform better than the LLL-Babai, in spite of its higher complexity. The performance gap between the LLL-Babai and LLL-ZF is about 1dB. It is interesting to note that the complex LLL reduction can provide the same performance gain as the traditional LLL reduction, in spite of its lower complexity. Also, the CLLL-Babai (as well as LLL-Babai, CLLL-VBLAST and

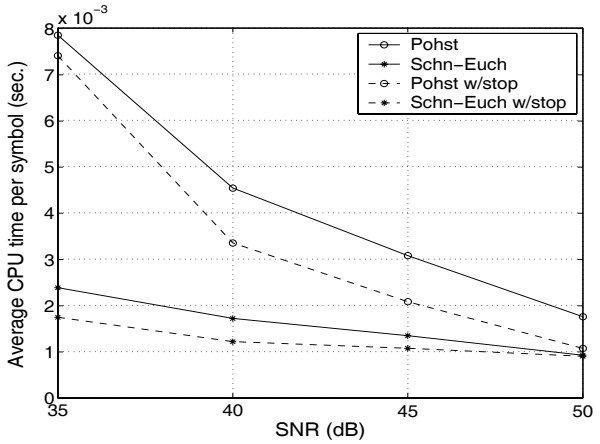

Fig. 2. Time complexity of the Pohst-Fincke enumeration algorithm with the original Pohst or the Schnorr-Euchner ordering, with or without the packing radius test, as a function of the SNR.

LLL-VBLAST) is only about 2.5dB from the MLD performance and can achieve the optimal diversity order. The CLLL-Babai is thus a very attractive low-complexity suboptimal detector.

The time complexities of different implementations of the optimal lattice decoder applied to the same 4-transmit 4-receive antenna 64-QAM system are compared in Figure 2. For simplicity, no LLL reduction preprocessing has been performed. Figure 2 shows the average CPU time per symbol of the Pohst-Fincke enumeration algorithm and its Schnorr-Euchner variant, with and without the packing radius stopping test as described in Section VI. The initial radius is set according to the Babai point, explicitly for the Pohst-Fincke algorithm and implicitly for the Schnorr-Euchner variant.

Figure 2 shows that the Schnorr-Euchner ordering can speed up the Pohst-Fincke algorithm by about 3 times at 35dB. However, the speedup factor becomes less than 2 as the SNR increases to 50dB. This can be explained by the fact that as the SNR increases, the enumeration sphere contains only a few lattice points and hence the order of enumerating them become less significant. It can also be seen from the figure that the use of the packing radius test can greatly speed up the Pohst-Fincke algorithm with both the Pohst and the Schnorr-Euchner orderings, but at different SNRs. In particular, the speedup factor for the former increases from 6% at 35dB to over 60% at 50dB, while that for the latter decreases from 36% at 35dB to only 2% at 50dB. Following the discussion in Section VI-C, with the packing radius test, the complexity of both the Pohst-Fincke algorithm and its Schnorr-Euchner variant should converge to that of the Babai nearest plane algorithm, as the SNR tends to infinity. It can be concluded from Figure 2 that the convergence has already occurred at 50dB. However, at practical values of SNR (i.e., $\leq$ 40dB), the speedup factor due to the packing radius test is significant only when combined with the Schnorr-Euchner ordering. Finally, we note that the Pohst-Fincke algorithm with both the Schnorr-Euchner ordering and the packing radius test appears to be the most efficient implementation of the optimal lattice decoder known, especially at moderate SNRs.

VIII. Conclusion

Several important lattice algorithms for performing the lattice basis reduction and for finding a nearby or the closest lattice point were reviewed. Specifically, the LLL reduction algorithm and the Pohst-Fincke enumeration algorithm with the Schnorr-Euchner ordering were discussed. The use of LLL reduction can provide impressive performance gain for the ZF, DFE and VBLAST detectors at the expense of affordable preprocessing complexity. To attain the MLD performance at low complexity, it is promising to apply the Pohst-Fincke enumeration algorithm with the Schnorr-Euchner ordering and the packing radius stopping test.

Our general treatment allows the LLL reduction algorithm to be extended for complex lattices with unnecessarily square basis matrices. This generalization is important for designing low-complexity universal lattice decoders for typical communications applications, in which the passband quadrature phase modulation schemes (such as QPSK and QAM) are used. Our simulation result verified the the complex LLL reduction algorithm, if applicable, can further reduce complexity without degrading performance.

References

[1] W. H. Mow. Maximum Likelihood Sequence Estimation from the Lattice Viewpoint. M.Phil. Thesis, Dept. of Information Engineering, the Chinese University of Hong Kong, June 1991; downloadable at http://www.ee.ust.hk/~eewhmow.

[2] W. H. Mow. Maximum Likelihood Sequence Estimation from the Lattice Viewpoint. IEEE Trans. Inform. Theory 1994; 40(5): 1591–1600.

[3] E. Viterbo, J. Boutros. A universal lattice code decoder for fading channels. IEEE Trans. Inform. Theory 1999; 45(5): 1639–1642.

[4] M. O. Damen, A. Chkeif, J.-C. Belfiore. Lattice code decoder for space-time codes. IEEE Commun. Lett. 2000; 4(5): 161–163.

[5] L. Brunel, J. J. Boutros. Lattice Decoding for Joint Detection in Direct-Sequence CDMA Systems. IEEE Trans. Inform. Theory 2003; 49(4): 1030–1037.

[6] H. Vikalo, B. Hassibi. Maximum-Likelihood Sequence Detection of Multiple Antenna Systems over Dispersive Channels via Sphere Decoding. EURASIP Journal on Applied Signal Processing 2002; 2002(5), 525–531.

[7] R. Kannan. Minkowski's Convex Body Theorem and Integer Programming. Math. Oper. Research 1987; 12: 415–440.

[8] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring Polynomials with Rational Coefficients. Math. Ann. 1982, 261: 513–534.

[9] A. Shamir. A Polynomial Time Algorithm for Breaking the Merkle-Hellman Cryptosystem. In Proc. 23rd IEEE Symp. on Foundations of Computer Science, 1982; 145–152.

[10] L. Lovász. An Algorithmic Theory of Numbers, Graphs and Convexity. Capital City Press, Montpelier, Vermont, 1986.

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

[12] G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. Bell Labs Technical Journal 1996; 1(2): 41–59.

[13] U. Fincke, M. Pohst. Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis. Math. Comp. 1985; 44:463–471.

[14] H. Yao, G. W. Wornell. Lattice-Reduction-Aided Detectors for MIMO Communication Systems. In Proc. IEEE Globecom, Taipei, November 17-21, 2002.

[15] C. P. Schnorr, M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming 1994; 66: 181–191.

[16] E. Agrell, T. Eriksson. A. Vardy, K. Zeger. Closest Point Search in Lattices. IEEE Trans. Inform. Theory 2002, 48: 2201–2213.

[17] B. Hassibi, H. Vikalo. On the Expected Complexity of Sphere Decoding. In Proc. Asilomar Conf. Signals, Systems and Computers 2001; 2: 1051–1055.

[18] W. H. Mow. Universal Lattice Decoding: Principle and Recent Advances. Wireless Communications and Mobile Computing 2003; 3(5): 553–569.