Reduced Complexity Sphere Decoding Using Forcing Rules

Tao Cui, Chintha Tellambura and Wen Chen
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4
Email: {taocui, chintha, wenchen}@ece.ualberta.ca

Abstract— **We improve the conventional sphere decoder (SD), which is widely used for lattice codes and multiple antenna space-time communications, by showing that some of the variables can be predetermined by using bounds on partial derivatives of the cost function. This reduces the number of variables to be estimated by the SD. Moreover, we combine the forcing rules combined with the conventional SD provide several hybrid algorithms, which are found to be computationally more efficient. Forcing rules reduces complexity significantly when the dimension of the lattice is less than 12.**

I. Introduction

Many communication theory problems can be formulated as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{x} \in \mathbb{Z}^m$, $\mathbf{y}, \mathbf{n} \in \mathbb{R}^n$ denote the system input, output and additive noise, and $\mathbf{H} \in \mathbb{R}^{n\times m}$ represents the transfer matrix (i.e., the channel matrix in MIMO system), $\mathbb{Z}^m$ denotes an m-dimensional integer lattice. Generally, the noise terms n_i, $i = 1, \cdots, n$ are independent and identically distributed (i.i.d.) zero-mean Gaussian random variables with the same variance. Under such conditions and assuming $\mathbf{H}$ is perfectly known at the receiver, the optimal maximum-likelihood (ML) detector that minimizes the average error probability is given by the integer least-squares problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in Z^m} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{2}$$

$\|\cdot\|$ denotes the Euclidean norm. The lattice generated by $\mathbf{H}$ is [1]

$$\Lambda(\mathbf{H}) = \{\mathbf{H}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^m\} \tag{3}$$

he columns of $\mathbf{H}$ are called basis vectors of Λ, and the number m is said to be the dimension of Λ. The problem in (2) is known as the closest vector problem (CVP) in the lattice theory. Other applications of CVP include vector quantization and cryptography.

The complexity of lattice problems has been widely investigated since the early 1980s because of the many connections between these problems and other areas of computer science and communication. CVP was proved NP-hard to solve exactly by van Emde Boas in [2]. Micciancio gave a simple proof in [3]. The first inapproximability results for CVP are due to Arora *et al.* [4] who proved that CVP is NP-hard to approximate within any constant factor, and quasi NP-hard to approximate within factor $n^{c/\log\log n}$ for some constant $c > 0$. The latter result is improved to a proper NP-hardness result by Dinur, Kindler, and Sufra in [5]. Interestingly, CVP remains hard to solve exactly even if the lattice is known in advance and can be arbitrarily preprocessed before the target point is found [3]. Shortest vector problem (SVP), which is important in cryptography, is the counterpart of CVP: given a lattice, find the shortest lattice point in the lattice. It is a special case of CVP when $\mathbf{y} = 0$. The NP-hardness of SVP was conjectured in [2] but remained an open problem for a long time. The first result is due to Ajtai [6] who proved that solving the problem exactly is NP-hard for randomized reductions. Ajtai's result can also be adapted to show the inapproximability of SVP within certain factors $1 + o(1)$ that rapidly approach 1 as the dimension of the lattice grows. Recently, Micciancio [7] showed that approximating the shortest vector problem within any constant factor less than $\sqrt{2}$ is NP-hard for randomized reductions.

A remarkable algorithm for CVP stems from Fincke and Phost (FP) [8], which is well-known as sphere decoding (SD) in communication theory [9], [10]. The FP algorithm only tests the lattice points lying inside a hypersphere. In wireless communication, SD first appeared in [11] on lattice code decoding. Since then, SD has gained popularity in CDMA [12], space-time coding [9] and MIMO systems over frequency selective channel [10]. SD offers near-ML decoding at lower complexity compared with the exponential complexity incurred by exhaustive search.

Further improving the SD for (2) has recently received considerable interest due to the significant performance gain achieved by SD compared with other suboptimal algorithms such as VBLAST [13]. In section II, we review the basic FP and SE techniques. In section III, we propose two prepossessing techniques. The first one is by setting a lower bound on the remaining terms which are viewed as zero in original SD by using constrained least squares (CLS). Some components of the vector can be predetermined by checking the sign and the absolute value of the partial derivative at the corresponding components. In section IV, we adopt the partial derivative into the original SD so that some components can be predetermined when going down the search tree. Hence, the computational complexity can be reduced. The complexity comparison between the SD algorithms is given in section V.

0-7803-8622-1/04/$20.00 ©2004 IEEE

Notation: Bold symbols denote matrices or vectors. $(\cdot)^T$ and $(\cdot)^H$ denote transpose and conjugate transpose, $(\cdot)^\dagger$ denotes pseudo-inverse. The set of all complex $K \times 1$ vectors is denoted by $\mathcal{C}^K$. For 2^q-ary phase shift keying (PSK), the signal constellation $\mathcal{Q}_{2^q} = \{e^{j2\pi k/2^q}, k = 0, 1, \cdots, 2^q - 1\}$ and all PSK $N \times 1$ vectors are denoted by $\mathcal{Q}_{2^q}^N$. 4-ary PSK is commonly known as quadrature PSK (QPSK). If x and y are Gaussian with $E[x] = \mu_x$, $E[y] = \mu_y$ and $E\left[(x - \mu_x)^2\right] = E\left[(y - \mu_y)^2\right] = \sigma^2/2$, then $z = x + jy$ (where $j = \sqrt{-1}$) is Complex Gaussian. We write $z \sim \mathcal{CN}(\mu_x + j\mu_y, \sigma^2)$ in this case. The $N \times N$ identity matrix is denoted by $\mathbf{I}_N$.

II. Predetermination

The original SD algorithm by Fincke and Phost [8] (FPSD hereafter) is proposed to solve the SVP. FPSD restricts the search space to the lattice points within a sphere instead of searching all the lattice points. Each time a valid lattice point is found, the search space is restricted further by updating the radius. Although its worst case complexity was shown to be exponential, FPSD has been widely used in CVP due to its efficiency on many communication problems.

Before using FPSD or exhaustive search, if some components of $\mathbf{s}$ can be determined, then some benefits accrue. The integer least squares problem (1) is equivalent to the following quadratic optimization problem

$$\min_{\mathbf{s} \in \mathcal{A}^M} f(s) = \frac{1}{2}\mathbf{s}^T\mathbf{A}\mathbf{s} + \mathbf{B}^T\mathbf{s} \tag{4}$$

where $\mathbf{A} = 2\mathbf{H}^T\mathbf{H}$ and $\mathbf{B} = -2\mathbf{H}^T\mathbf{y}$. If $\mathcal{A} = \{0, 1\}$. Eq.(4) is known as Quadratic Bivalent Programming in *Operations Research*, which belongs to a class of NP-complete combinatorial optimization problems. Hansen [14] gives a survey of the techniques used prior to 1980. Most of the proposed algorithms are of the branch and bound type. In [15]–[17], a preprocessing technique to fix variables by using the gradient of the quadratic function is proposed. This technique takes advantage of the special structure of the quadratic problem. From II C, QPSK modulated complex system can be decoupled to BPSK real system. Hence, we first apply the predetermination technique to BPSK systems and then generalize it to other constellations in this subsection.

Theorem 1 Let $\mathbf{s}^*$ be the global minimum of the problem

$$\min f(s) = \frac{1}{2}\mathbf{s}^T\mathbf{A}\mathbf{s} + \mathbf{B}^T\mathbf{s}, \quad -1 \le s_i \le 1 \tag{5}$$

Then $\mathbf{s}^*$ is also optimal for the linear problem

$$\min(\nabla f(\mathbf{s}^*))^T\mathbf{s}, \quad -1 \le s_i \le 1 \tag{6}$$

Proof For any feasible point $\mathbf{y}$ and $\lambda \in (0, 1)$, the point $\mathbf{z} = (1-\lambda)\mathbf{s}^* + \lambda\mathbf{y}$ is also feasible and therefore $f(\mathbf{z}) - f(\mathbf{s}^*) \ge 0$, or $\lambda(\mathbf{B} + \mathbf{A}\mathbf{s}^*)^T(\mathbf{y} - \mathbf{s}^*) + \frac{1}{2}\lambda^2(\mathbf{y} - \mathbf{s}^*)^T\mathbf{A}(\mathbf{y} - \mathbf{s}^*) \ge 0$. Since $\lambda \ne 0$, this implies that $(\mathbf{B} + \mathbf{A}\mathbf{s}^*)^T(\mathbf{y} - \mathbf{s}^*) \ge -\frac{1}{2}\lambda(\mathbf{y} - \mathbf{s}^*)^T A(\mathbf{y} - \mathbf{s}^*)$. If $\lambda \to 0$, we have $(\mathbf{B} + \mathbf{A}\mathbf{s}^*)^T(\mathbf{y} - \mathbf{s}^*) \ge 0$. That is $f(\mathbf{s}^*)^T\mathbf{y} \ge f(\mathbf{s}^*)^T\mathbf{s}^*$ for all feasible $\mathbf{y}$.

¿From Theorem 1, we can get the following corollary directly.

Corollary 1 Suppose that $m_i \le \partial f(\mathbf{s})/\partial s_i \le M_i$ for all $\mathbf{s} \in \{-1, 1\}^M$ and $i = 1, 2, \cdots, M$. If $m_i > 0$, then $s_i^* = -1$. If $M_i < 0$, then $s_i^* = 1$.

Corollary 1 can be interpreted as if the partial derivative of $f(\mathbf{s})$ with respect to variable s_i is always larger than zero (or less than zero), $f(\mathbf{s})$ will be monotonously increasing (or decreasing) in the ith dimension and then s_i can be fixed as -1 (or 1). The bounds m_i and M_i that force variables to be fixed can be improved by exploiting the constant modulus property of set {-1,1}. Note that the objective function in the original optimization problem (4) is equivalent to

$$\min \tilde{f}(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T\tilde{\mathbf{A}}\mathbf{s} + \mathbf{B}^T\mathbf{s} \tag{7}$$

where $\tilde{a}_{ij} = a_{ij}$ for $i \ne j$ and $\tilde{a}_{ii}$, $i = 1, 2, \ldots, M$ can be any real number. The partial derivative of $\tilde{f}(\mathbf{s})$ with respect to variable s_i is

$$\begin{aligned} \frac{\partial \tilde{f}(\mathbf{s})}{\partial s_i} &= \tilde{\mathbf{a}}_i^T\mathbf{s} + b_i \\ &= \sum_{j=1, j\ne i}^{M} a_{ij}s_j + \tilde{a}_{ii}s_i + b_i \end{aligned} \tag{8}$$

where $\tilde{\mathbf{a}}_i$ is the ith row of $\mathbf{A}$. Since $s_j \in \{-1, 1\}$, $j = 1, 2, \ldots, M$, we have

$$-\sum_{j=1, j\ne i}^{M} |a_{ij}| - |\tilde{a}_{ii}| + b_i \le \frac{\partial \tilde{f}(\mathbf{s})}{\partial s_i} \le \sum_{j=1, j\ne i}^{M} |a_{ij}| + |\tilde{a}_{ii}| + b_i. \tag{9}$$

Hence the range of the partial derivative is $2\sum_{j=1, j\ne i}^{M} |a_{ij}| + 2|\tilde{a}_{ii}|$. To improve the bounds m_i and M_i, a heuristic consideration is to choose $\tilde{a}_{ii}$ that minimize the range of the partial derivative. The range of the partial derivative is minimized for $\tilde{a}_{ii} = 0$, $i = 1, 2, \ldots, M$. Therefore, the minimization problem (4) can by replaced by the equivalent optimization problem

$$\min_{\mathbf{s} \in \{-1, 1\}} f(s) = \frac{1}{2}\mathbf{s}^T\mathbf{A}_0\mathbf{s} + \mathbf{B}^T\mathbf{s} \tag{10}$$

where $\mathbf{A}_0$ is the matrix $\mathbf{A}$ whose diagonal entries are replaced by zero. Hence

$$m_i = -\sum_{j=1, j\ne i}^{M} |a_{ij}| + b_i, \quad M_i = \sum_{j=1, j\ne i}^{M} |a_{ij}| + b_i \tag{11}$$

Note that the bounds m_i and M_i can be improved further by using the predetermined variables prior to the test of s_i. If P_i denotes the index set whose components are the indexes corresponding to the predetermined components of $\mathbf{s}$ prior to the test of the ith components, from (8) we have

$$\begin{aligned} -\sum_{j=1, j\bar{\in} P_i}^{M} |a_{ij}| + \sum_{j\in P_i} a_{ij}s_j^* + b_i &\le \frac{\partial \tilde{f}(\mathbf{s})}{\partial s_i} \\ &\le \sum_{j=1, j\bar{\in} P_i}^{M} |a_{ij}| + \sum_{j\in P_i} a_{ij}s_j^* + b_i. \end{aligned} \tag{12}$$

The range of partial derivative reduces to $2\sum_{j=1, j\bar{\in} P_i}^{M} |a_{ij}|$. Since the latter tested variables can use the prior predetermined results, after testing all the components of s for one time all the variables are tested again and again until no further components of s can be predetermined. In this way, all the components of s can use other components' predetermination information. We call this Predetermination Algorithm 1 or shortly **Predetermination 1**.

However, if the sign of the partial derivative is not constant, the predetermination fails. To the best of the authors' knowledge, little attention has been paid to the use of the absolute value of the partial derivative for predetermination in the literature. In the following, we give another predetermination theorem due to the special structure of the quadratic form.

Theorem 2 Suppose that

$$m_i^{-1} \leq \left| \frac{\partial f(\mathbf{s})}{\partial s_i}\bigg|_{s_i=-1} \right| \leq M_i^{-1}, \quad m_i^{1} \leq \left| \frac{\partial f(\mathbf{s})}{\partial s_i}\bigg|_{s_i=1} \right| \leq M_i^{1} \tag{13}$$

for all $\mathbf{s} \in \{-1, 1\}^M$ and $i = 1, 2, \ldots, M$. If $M_i^{-1} < m_i^1$, then $s_i^* = -1$. If $M_i^1 < m_i^{-1}$, then $s_i^* = 1$.

Proof For feasible points $\mathbf{s}_1$ and $\mathbf{s}_2$, all the components of $\mathbf{s}_1$ and $\mathbf{s}_2$ are equal except the ith component. That is $s_{1j} = s_{2j}$, $j \neq i$ and $s_{1i} = s_{2i}$. The optimization problem (4) is equivalent to

$$\min g(s_i) = \frac{a_{ii}}{2} s_i^2 + (\sum_{j=1, j\neq i}^{M} a_{ij} s_j + b_i) s_i \tag{14}$$

Note that since $\mathbf{A} = 2\mathbf{H}^T\mathbf{H}$, $\mathbf{A} = \mathbf{A}^T$, $a_{ij} = a_{ji}$ and $a_{ii} > 0$. From (8), the partial derivative of $f(\mathbf{s})$ with respect to variable s_i is

$$\frac{\partial f(\mathbf{s})}{\partial s_i} = a_{ii} s_i + \sum_{j=1, j\neq i}^{M} a_{ij} s_j + b_i = \frac{\mathrm{d}g(s_i)}{\mathrm{d}s_i} \tag{15}$$

Thus we have

$$\left| \frac{\mathrm{d}g(s_i)}{\mathrm{d}s_i} \right|^2 = 2a_{ii} g(s_i) + (\sum_{j=1, j\neq i}^{M} a_{ij} s_j + b_i)^2 \tag{16}$$

Since $a_{ii} > 0$, minimizing $g(s_i)$ is equivalent to minimize $|\mathrm{d}g(s_i)/\mathrm{d}s_i|$. For all $\mathbf{s} \in \{-1, 1\}^M$, if $M_i^{-1} < m_i^1$, $g(-1) < g(1)$ and $f(\mathbf{s}_1) < f(\mathbf{s}_2)$ where $s_{1j} = -1$, $s_{2j} = 1$. Therefore, s_i is fixed to -1 and vice versa.

The theorem is valid due to the special structure of the quadratic form. We follow the iterative testing procedure in the discussion of Theorem 1. We call this Predetermination Algorithm 2 or shortly **Predetermination 2**.

Theorem 1 and 2 can be readily generalized to the case that the number of elements in $\mathcal{A}$ or the cardinal number of $\mathcal{A}$ is larger than 2.

Remarks:

- If P variables can be predetermined, the search space will reduce to $\mathcal{A}^{M-P}$, which means the complexity of the search can be significantly reduced. From simulation, we find that when N is small, most of the variables can be predetermined.
- The predetermination technique can be combined with the lower bound preprocessing to achieve lower complexity.
- If the channel is constant over several blocks, $\sum |a_{ij}|$ does not need to be calculated again. The predetermination stage computational cost is negligible.

III. Sphere Decoding with Predetermination

We next incorporate predetermination into the FP decoder. The FP decoder [8] solves the equivalent problem

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \mathcal{A}^M} \|\mathbf{y}' - \mathbf{R}\mathbf{s}\|^2 \tag{17}$$

where $\mathbf{y}' = \mathbf{Q}_1^H \mathbf{y}$, $\mathbf{R}$ an $M \times M$ upper triangular matrix and $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2]$ an $N \times N$ orthogonal matrix are QR factorization of the matrix $\mathbf{H}$. The matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ represent the first M and last $N - M$ orthonormal columns of $\mathbf{Q}$.

Hybrid Algorithm 1: When the search proceeds to s_i and the size of candidate set for s_i is larger than one, we check the bounds for $\partial f(\mathbf{s})/\partial s_i$, $m_i = -\sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{M} a_{ij} s_j^* + b_i$, $M_i = \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{M} a_{ij} s_j^* + b_i$. s_j^* ($j = i+1, \ldots, M$) have already been assigned values. Using Theorem 1, if $m_i > 0$, $s_i^* = -1$ and if $M_i < 0$, $s_i^* = 1$. Since $\sum_{j=1}^{i-1} |a_{ij}|$ can be pre-computed before search, it needs $2(M-i)$ flops to obtain the bounds. Theorem 2 can be applied in the same way.

Hybrid Algorithm 2: In (17), when s_{i+1} has been assigned a value, $\mathbf{r}_{i+1} s_{i+1}$ can be subtracted from $\mathbf{y}'$ and it forms a new equivalent problem

$$\mathbf{s}^i = \arg \min_{\mathbf{s}^i \in \mathcal{A}^i} \|\mathbf{y}'^i - \mathbf{R}^i \mathbf{s}^i\|^2 \tag{18}$$

where $\mathbf{s}^i = [s(1), s(2), \ldots, s(i)]^T$, $\mathbf{y}'^i = \mathbf{y}'^{i+1} - \mathbf{r}_{i+1} s_{i+1}$, $\mathbf{R}^i$ is the $i \times i$ submatrix of $\mathbf{R}$ and $\mathbf{r}_{i+1}$ is the $(i+1)$-th column of $\mathbf{R}$. Problem (18) is equivalent to

$$\min f(\mathbf{s}^i) = \frac{1}{2} (\mathbf{s}^i)^T \mathbf{A}^i \mathbf{s}^i + (\mathbf{B}^i)^T \mathbf{s}^i \tag{19}$$

where $\mathbf{A}^i = (\mathbf{R}^i)^T \mathbf{R}^i$ and $\mathbf{B}^i = -\mathbf{R}^i \mathbf{y}'^i$. The bounds for the partial derivative of $f(\mathbf{s}^i)$ with respect to s_i are $m_i = -\sum_{j=1}^{i-1} |a_{ij}^i| + b_i$, $M_i = \sum_{j=1}^{i-1} |a_{ij}^i| + b_i$. Since $a_{ij}^i = a_{ij}$ where a_{ij} are elements of $\mathbf{A} = \mathbf{R}^T\mathbf{R}$, $\sum_{j=1}^{i-1} |a_{ij}^i|$ can be pre-computed. Calculating b_i needs $2i$ flops.

IV. Simulation Results and Conclusion

We now compare the proposed algorithms with the FP decoder. We consider an M transmit and N receive antennas, BPSK modulated, uncoded MIMO system over a multiple antenna flat Rayleigh fading channel. We simulate this system using MATLAB V5.3 on a PC with an Intel Pentium-4 processor at 1.8GHz. The matlab command "flops" is used to count the number of flops. Only the flops of the search algorithm are counted without accounting for the prepossessing stage. The initial radius is chosen according to the noise variance as in [18]. Fig. 1 compares the average complexity of the two predetermination techniques with that of the FP decoder for SNR=15dB. Predetermination is especially effective when

$m \leq 12$. Surprisingly, $\mathbf{s}$ can sometimes be totally predetermined without performing any search when $m \leq 12$, which is often the case in practical systems. When m increases, the bounds for partial derivative becomes loose. Fig. 2 show the average complexity of different algorithms: Hybrid 1, Hybrid 2 and FP, when SNR=15dB. The two hybrid algorithms are effective when m is small. If $m > 12$, the forcing rules can be applied in the last 12 dimensions.

In this paper, we developed reduced complexity variants of the SD algorithm by using forcing rules that utilizes bounds on the partial derivatives. Simulation results demonstrate that such hybrid algorithms are effective when the dimension of the lattice is less than 12.

References

[1] J. H. Conway and N. J. Sloane, *Sphere packing, Lattices and Groups*, 3rd ed. New York: Springer-Verlag, 1998.

[2] P. van Emde Boas, “Another NP-complete problem and the complexity of computing short vectors in a lattice,” 1981, Tech. Report 81-04, Mathematische Instituut, Universiry of Amsterdam.

[3] D. Micciancio, “The hardness of the closest vector problem with preprocessing,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 1584–1589, July 2000.

[4] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk, “The hardness of approximate optima in lattices,codes,and systems of linear equations,” *J. Comput. System Sci.*, vol. 54, pp. 317–331, 1997.

[5] I. Dinur, G. Kindler, and S. Safra, “Approximating CVP to within almost-polynomial factors is np-hard,” in *Proceedings of 39th Annual IEEE Symposium on Foundations of Computer Science*, Palo Alto, CA, 1998.

[6] M. Ajtai, “The shortest vector problem in L2 is NP-hard for randomized reductions,” in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, Dallas, TX, 1998, pp. 10–19.

[7] D. Micciancio, “The shortest vector in a lattice is hard to approximate to within some constant,” *SIAM J. Comput.*, vol. 30, no. 6, pp. 2008–2035, 2001.

[8] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, inlcuding a complexity analysis,” *Math. Computation*, vol. 44, pp. 463–471, Apr. 1985.

[9] O. Damen, A. Chkeif, and J.-C. Belfiore, “Lattice code decoder for space-time codes,” *IEEE Communications Letters*, vol. 4, no. 5, pp. 161 – 163, May 2000.

[10] H. Vikalo and B. Hassibi, “On maximum-likelihood sequence detection for multiple antenna systems over dispersive channels,” *EURASIP Journal on Applied Signal Processing*, vol. Special Issue on Space-Time Coding, pp. 525–531, May 2002.

[11] E. Viterbo and E. Biglieri, “A universal lattice decoder,” Juan-les-Pins, France, Sept. 1993, pp. 611–614.

[12] L. Brunel and J. Boutros, “Euclidean space lattice decoding for joint detection in CDMA systems,” in *Proceedings of IEEE Information Theory and Communications Workshop*, June 1999, p. 129.

[13] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, “Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture,” *Electronics Letters*, vol. 35, no. 1, pp. 14–15, Jan. 1999.

[14] P. Hansen, “Methods of nonlinear 0-1 programming,” *Annals of Discrete Mathematics*, vol. 5, pp. 53–70, 1979.

[15] P. M. Pardolos and G. P. Rodgers, “Parallel branch and bound algorithms for unconstrained zero-one programming,” in *Impacts of Recent Computer Advances on Operations Research*, R. Sharda, Ed. New York: North-Holland, 1989, pp. 131–143.

[16] P. M. Pardalos and G. P. Rodgers, “Computational aspects of a branch and bound algorithm for quadratic zero-one programming,” *Computing*, vol. 45, pp. 131–144, 1990.

[17] P. M. Pardalos, “Construction of test problems in quadratic bivalent programming,” *ACM Transactions on Mathematical Software*, vol. 17, pp. 74–87, 1991.

[18] B. Hassibi and H. Vikalo, “On the sphere decoding algorithm: Part I, the expected complexity,” *IEEE Transactions on Signal Processing*, submitted.

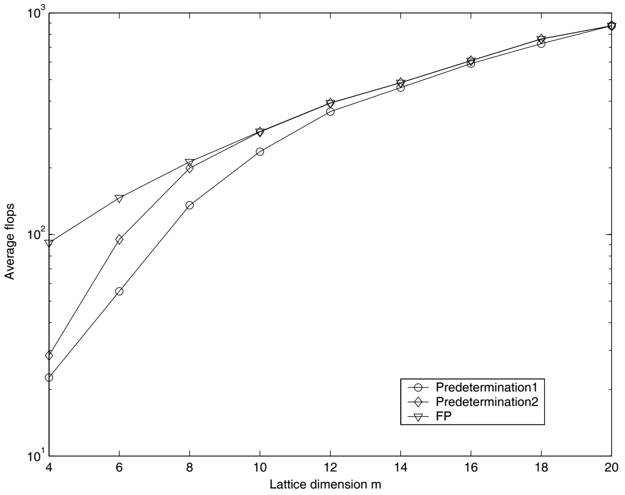

Fig. 1. The effect of preprocessing when SNR=15dB

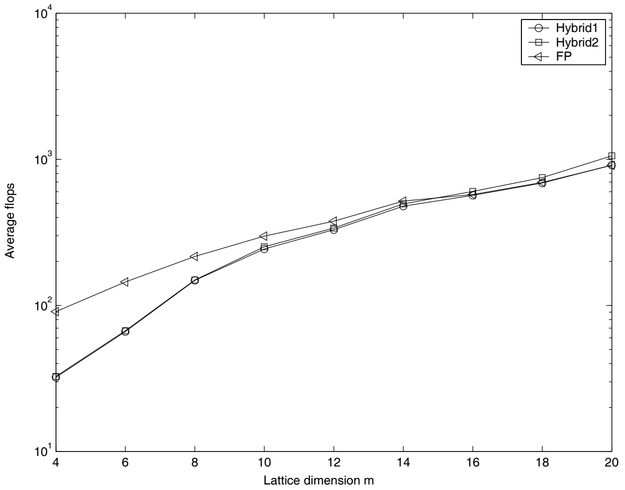

Fig. 2. The complexity of different sphere decoding algorithms when SNR=15dB