스마트 반지름 제어를 사용한 효율적인 구복호 알고리즘

한희구, 오성근
아주대학교 정보통신대학 전자공학부

A Computationally Efficient Sphere Decoding Algorithm with Smart Radius Control

Hee Goo Han, Seong Keun Oh
School of Electrical and Computer Engineering
Ajou University
E-mail : hanhg@ajou.ac.kr, oskn@ajou.ac.kr

Abstract

We propose a computationally efficient sphere decoding (SD) algorithm with smart radius control (SRC). As a baseline algorithm for SD, we consider the modified Schnorr-Euchner (SE) algorithm [1] (hereafter, called as the MSE algorithm). In principle, the radius after zero-forcing decision feedback equalization (ZF-DFE) estimation can be reduced further if we select a new lattice vector closer to the received signal vector than the lattice vector corresponding to the ZF-DFE estimate does. In our case, we obtain such a better lattice vector by performing a sequence of alternating one-dimensional searches, starting from the ZF-DFE estimate. We then develop a novel SRC algorithm that adopts adaptively the additional radius reduction process according to the estimated signal-to-noise-power ratio (SNR) after ZF-DFE estimation. In addition, we analyze the effect of detection ordering on the complexity for SD. Column-norm ordering of the channel matrix and optimal ordering [1] are considered here. From our analyses, we see that SRC can reduce greatly the complexity for SD and the degree of complexity reduction gets significant as the SNR decreases, irrespective of detection ordering schemes used.

I. Introduction

SD [1]-[4] can achieve an exact maximum likelihood (ML) performance with a moderate complexity by confining the search space inside a hypersphere with a given radius, centered at the received signal vector. The ML performance is guaranteed if the radius is chosen so that at least one valid lattice vector could be included inside the hypersphere. The complexity for SD is still heavily dependent on the choice of the initial radius since it determines the number of lattice vectors inside the hypersphere to be searched over. Moreover, it is also dependent on how to search lattice vectors inside the hypersphere. Tree-search processing [1]-[4] has been widely employed to search the lattice vectors inside the hypersphere. Hence, the complexity for SD is approximately proportional to the number of visited lattice points during the tree-search process. Two tree-search strategies have been widely used to avoid the full search over all the lattice points inside the hypersphere, thus reducing further the complexity for SD [1]-[4]. One is to shrink continuously the current hypersphere through continuous radius updating. The other is to search more intelligently the lattice vectors inside the hypersphere in order to find quickly the ML estimate. However, they should be combined together in order to get a synergistic effect to reduce effectively the computational complexity.

The MSE algorithm (i.e., the algorithm II in [1]) in which the SE strategy has been modified to take into account the finite signal set boundary merges the above two strategies by using the SE enumeration rather than the traditional Pohst enumeration [1] for tree search. The algorithm has very low sensitivity to the initial radius since it can always obtain a ZF-DFE estimate as the first valid estimate irrespective of the initial radius [1]. Especially, it has been well known that the MSE algorithm can offer a very attractive implementation of SD with finite-alphabet constellations. Its complexity is mainly determined by the radius after ZF-DFE estimation (hereafter, simply called as the DFE-radius), that is, the Euclidean distance between the lattice vector corresponding to the ZF-DFE estimate and the received vector. However, the DFE-radius might be still extremely large as compared to the ML distance corresponding to the ML estimate, thus including an extremely large number of lattice vectors inside the hypersphere [1],[4]. Hence, if a valid radius smaller than the DFE-radius is found, the complexity could be further reduced.

In this paper, we propose a computationally efficient SD algorithm with SRC. As a baseline algorithm for SD, we consider the MSE algorithm since it has a radius-insensitive complexity. In principle, the DFE-radius can be reduced further by selecting a new lattice vector closer to the received signal vector in terms of Euclidean distance than the ZF-DFE lattice vector does. In our case, we obtain such a better lattice vector by performing a sequence of alternating one-dimensional searches, starting from the ZF-DFE estimate. Then, we develop the SRC algorithm for SD in which after ZF-DFE estimation, the additional radius reduction process is adaptively adopted according to the estimated SNR. In addition, we analyze the effect of detection ordering on the complexity since the complexity of the MSE algorithm is also highly dependent on the detection ordering strategy used. Column-norm ordering of the channel matrix and optimal ordering [1],[5] are considered here. For complexity analyses, we evaluate computations for both pre-processing and tree-search processing. Pre-processing can be divided into two parts: detection ordering and QR decomposition of the channel matrix. Computations for tree-search processing are obtained by first evaluating the number of operations to get a lattice point at each dimension, and then counting the number of visited lattice points for respective dimensions through computer simulation. We use the average number of operations as a performance measure.

II. System Description

We consider a symbol-synchronized multiple-input multiple-output (MIMO) system having N transmit antennas and M receive antennas ($N \le M$) and also using Q-QAM constellation. Then, the $M \times 1$ complex-valued received signal vector can be expressed as,

$$\tilde{\mathbf{r}} = \tilde{\mathbf{H}} \cdot \tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \qquad (1)$$

where $\tilde{\mathbf{H}}$ denotes the $M \times N$ channel matrix whose channel coefficients are independent and identically distributed (i.i.d.) zero-mean, unit-variance complex Gaussian random variables; $\tilde{\mathbf{x}}$ is the $N \times 1$ transmit symbol vector with each element having Q-QAM format; $\tilde{\mathbf{n}}$ is the $M \times 1$ noise vector of which elements are i.i.d. zero-mean complex Gaussian random variables with equal variance of σ_n^2.

For a lattice representation, the complex-valued received signal vector is transformed into the equivalent real-valued received signal vector by stacking the real part and the imaginary part of $\tilde{\mathbf{r}}$ as $\mathbf{r} = \mathbf{H} \cdot \mathbf{x} + \mathbf{n}$ [2]-[4], i.e.,

$$\begin{bmatrix} \Re(\tilde{\mathbf{r}}) \\ \Im(\tilde{\mathbf{r}}) \end{bmatrix} = \begin{bmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\tilde{\mathbf{x}}) \\ \Im(\tilde{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \Re(\tilde{\mathbf{n}}) \\ \Im(\tilde{\mathbf{n}}) \end{bmatrix}, \qquad (2)$$

where $\mathbf{r} = [\Re(\tilde{\mathbf{r}}), \Im(\tilde{\mathbf{r}})]^T$ is the $m \times 1$ ($m = 2M$) real-valued received signal vector with the superscript T denoting the transpose; $\tilde{\mathbf{x}} = [\Re(\tilde{\mathbf{x}}), \Im(\tilde{\mathbf{x}})]^T$ is the $n \times 1$ ($n = 2N$) real-valued transmit symbol vector; $\tilde{\mathbf{n}} = [\Re(\tilde{\mathbf{n}}), \Im(\tilde{\mathbf{n}})]^T$ is the $m \times 1$ real-valued noise vector; and

$$\mathbf{H} = \begin{bmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{bmatrix}.$$

In the above lattice representation, $\mathbf{x}$ can be interpreted as an n-dimensional vector defined on n-dimensional integer lattice $\mathbb{Z}_q^n$ with $q = \sqrt{Q}$. For example, $\mathbb{Z}_4 = \{-3, -1, 1, 3\}$ when using 16-QAM modulation with the average power of 10.

III. MSE Algorithm

Generic ML detection (MLD) solves the following problem by exhaustive search,

$$\hat{\mathbf{x}}_{ML} = \arg\min_{\mathbf{x}} \left\| \mathbf{r} - \mathbf{H} \cdot \mathbf{x} \right\|^2. \qquad (3)$$

Hence, MLD requires the exponential complexity according to the number of modulation levels and/or the dimension of symbol vector $\mathbf{x}$, that is, the number of symbols to be jointly detected [6]. SD has been considered as a promising alternative that can reduce the computational complexity while achieving the exact ML performance [1],[2],[4].

In principle, SD should search all lattice vectors inside the hypersphere with the initial radius r. Therefore, it is required that $\mathbf{x}$ must satisfy the following inequality as

$$r^2 \ge \left\| \mathbf{Q}^T \cdot \mathbf{r} - \mathbf{R} \cdot \mathbf{x} \right\|^2 = \left\| \mathbf{Q}_1^T \cdot \mathbf{r} - \mathbf{R} \cdot \mathbf{x} \right\|^2 + \left\| \mathbf{Q}_2^T \cdot \mathbf{r} \right\|. \qquad (4)$$

Here, $\mathbf{R}$ and $\mathbf{Q}$ are the $n \times n$ upper triangular matrix and the $m \times m$ unitary matrix, respectively, obtained by the QR decomposition of the real-valued channel matrix $\mathbf{H}$. The matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$, respectively, represent the first n and last $m - n$ orthogonal columns of $\mathbf{Q}$. Using $\mathbf{y} = \mathbf{Q}_1^T \cdot \mathbf{r}$, $\mathbf{z} = \mathbf{Q}_2^T \cdot \mathbf{r}$ and $r'^2 = r^2 - \left\| \mathbf{z} \right\|^2$, (4) can be rewritten as

$$\begin{aligned} r'^2 &\ge \sum_{i=1}^{n} \left| y_i - \sum_{j=i}^{n} r_{i,j} x_j \right|^2 \\ &= \left| y_n - r_{n,n} x_n \right|^2 + \left| y_{n-1} - r_{n-1,n} x_n - r_{n-1,n-1} x_{n-1} \right|^2 + \cdots \end{aligned} \qquad (5)$$

Now, we review briefly the MSE algorithm [1] with a slight modification to consider regular QAM constellations as follows.

Input : r', $\mathbf{y}$, $\mathbf{R}$ Output : $\hat{\mathbf{x}}$

Step 1. (Initialization)

Set $i := n, T_n := 0, \xi_n := 0, r := r'$

Step 2. (DFE on x_i)

Set $x_i := \mathcal{Q}[(y_i - \xi_i) / r_{i,i}]$

$\Delta_i := \mathrm{sign}(y_i - \xi_i - r_{i,i} x_i)$.

Step 3. (Main step)

If $r < T_i + \left| y_i - \xi_i - r_{i,i} x_i \right|^2$, then go to Step 4.

Else if $x_i \notin \mathbb{Z}_q$, then go to Step 6.

Else if $i > 1$, then

$$\xi_{i-1} := \sum_{j=i}^{n} r_{i-1,j} x_j, \; T_{i-1} := T_i + \left| y_i - \xi_i - r_{i,i} x_i \right|^2$$

$i = i - 1$, and go to Step 2.

Else go to Step 5.

Step 4. If $i = n$, terminate, else set $i := i + 1$ and go to Step 6.

Step 5. (Valid lattice vector)

Set $r := T_1 + \left| y_1 - \xi_1 - r_{1,1} x_1 \right|^2$,

Save $\hat{\mathbf{x}} = \mathbf{x}$, $i := i + 1$ and go to Step 6.

Step 6. (SE enumeration)

$x_i := x_i + \Delta_i$,

$\Delta_i = -\Delta_i - \mathrm{sign}(\Delta_i)$ and go to Step 3.

In the above, $\mathcal{Q}[\cdot]$ denotes quantization to the closest lattice point.

IV. Radius Reduction Control (RRC)

Rewriting the DFE-radius as

$$r = \sqrt{\left\| \mathbf{y} - \mathbf{R} \cdot \hat{\mathbf{x}} \right\|^2 + \left\| \mathbf{z} \right\|^2}, \qquad (6)$$

we introduce an effective RRC scheme that can reduce further the DFE-radius after ZF-DFE estimation in the MSE algorithm. It consists of a sequence of alternating one-dimensional minimization processes, starting from (r, $\hat{\mathbf{x}}$) as follows:

Initialization (r, $\hat{\mathbf{x}}$)
For $i = 1$ to n

$$\left(\hat{x}_i^v, r^v \right) = \begin{pmatrix} \arg\min_{x_i^v \in \mathbb{Z}_q^{(i)}} \sqrt{\left\| \mathbf{y} - \mathbf{R} \cdot \hat{\mathbf{x}}^{(i)} \right\|^2 + \left\| \mathbf{z} \right\|^2}, \\ \sqrt{\left\| \mathbf{y} - \mathbf{R} \cdot \hat{\mathbf{x}}^v \right\|^2 + \left\| \mathbf{z} \right\|^2} \end{pmatrix}; \qquad (7)$$

If $r^v < r$, then $(r, \hat{\mathbf{x}}) = (r^v, \hat{\mathbf{x}}^v)$,
End;

Here, $\hat{\mathbf{x}}^{(i)} = [\hat{x}_1 \cdots \hat{x}_{i-1} \, x_i^v \, \hat{x}_{i+1} \cdots \hat{x}_n]^T$ can be obtained from $\hat{\mathbf{x}}$ by replacing $\hat{x}_i$ by $x_i^v \in \mathbb{Z}_q^{(i)}$; $\mathbb{Z}_q^{(i)}$ is a set deleted the element "$\hat{x}_i$" in $\mathbb{Z}_q$ (e.g., $\mathbb{Z}_4^{(i)} = \{-3, -1, 3\}$ with 16-QAM modulation and $\hat{x}_i = 1$); and $\hat{\mathbf{x}}^v = [\hat{x}_1 \cdots \hat{x}_{i-1} \, \hat{x}_i^v \, \hat{x}_{i+1} \cdots \hat{x}_n]^T$.

For the above RRC process, we use the ZF-DFE estimate and the corresponding radius (i.e., the DFE-radius) as starting parameters. We first choose the uppermost dimension at the pre-determined order. Kept fixed the estimates at all other dimensions except for the chosen dimension, we find a new estimate at the chosen dimension by searching over all the possible points on $\mathbb{Z}_q^{(i)}$, that is, $(r^v, \hat{x}_i^v)$, which the resulting new lattice vector minimizes the Euclidean distance from the received vector. If $r^v < r$, the current radius and estimate are updated by the new radius and estimate, respectively. This alternating one-dimensional search process will be continued until all n dimensions are exhausted [4].

In the above alternating procedure, note that some pre-computed values can be reused through element-wise recursion. Hence, we can further simplify the above procedure as follows:

Initialization $(r, \hat{\mathbf{x}})$, $\mathbf{R} = [\mathbf{r}_1 \cdots \mathbf{r}_n]$, $\mathbf{e}_0^{(0)} = \begin{bmatrix} \mathbf{y} - \mathbf{R}\cdot\hat{\mathbf{x}} \\ \mathbf{z} \end{bmatrix}$

For $i = 1$ to n
$\mathbf{e}_f^{(i)} = \mathbf{e}_0^{(i-1)} + \mathbf{r}_i \hat{x}_i$; : Fixed for i th dimension
$\left(\hat{x}_i^v, r^v\right) = \left(\arg\min_{x_i^v \in \mathbb{Z}_q^{(i)}} \left\|\mathbf{e}_r^{(i)}\right\|, \left\|\mathbf{e}_r^{(i)}\right\|_{x_i^v = \hat{x}_i^v} \right)$
with $\mathbf{e}_r^{(i)} = \mathbf{e}_f^{(i)} - \mathbf{r}_i x_i^v = \mathbf{e}_0^{(i-1)} + \mathbf{r}_i(\hat{x}_i - x_i^v)$; (8)
If $r^v < r$,
then $(r, \hat{\mathbf{x}}) = (r^v, \hat{\mathbf{x}}^v)$, $\mathbf{e}_0^{(i)} = \mathbf{e}_0^{(i-1)} + \mathbf{r}_i(\hat{x}_i - \hat{x}_i^v)$;
Else $\mathbf{e}_0^{(i)} = \mathbf{e}_0^{(i-1)}$;
End;

In the above procedure, the Euclidean distance $\left\|\mathbf{e}_r^{(i)}\right\|^2$ can be again computed by using the pre-computed values as follows:

$$\begin{aligned}\left\|\mathbf{e}_r^{(i)}\right\|^2 &= \mathbf{e}_r^{(i)T}\cdot\mathbf{e}_r^{(i)} = \left\|\mathbf{e}_0^{(i-1)}\right\|^2 + 2\mathbf{e}_0^{(i-1)T}\cdot\mathbf{r}_i(\hat{x}_i - x_i^v) + \left\|\mathbf{r}_i\right\|^2(\hat{x}_i - x_i^v)^2 \\ &= \left\|\mathbf{e}_0^{(0)}\right\|^2 + \sum_{j=0}^{i-2}(\hat{x}_{j+1} - \hat{x}_{j+1}^v)\left\{2\mathbf{e}_0^{(j)T}\cdot\mathbf{r}_{j+1} + \left\|\mathbf{r}_{j+1}\right\|^2(\hat{x}_{j+1} - \hat{x}_{j+1}^v)\right\} \\ &\quad + (\hat{x}_i - x_i^v)\left\{2\mathbf{e}_0^{(i-1)T}\cdot\mathbf{r}_i + \left\|\mathbf{r}_i\right\|^2(\hat{x}_i - x_i^v)\right\}\end{aligned} \quad (9)$$

Here, the squared Euclidean distance at the i th dimension can be simply computed by using the pre-computed values up to the ($i-1$)th dimension. Hence, the procedure in (8) requires only $2N^2 + 4Nq + 3N - 2$ multiplications [4].

The above RRC process can reduce further the radius after ZF-DFE estimation with high probability if the ZF-DFE estimate is still different from the ML estimate. However, it requires extra computations for the process itself. At low SNR values, such extra computations are negligible as compared to the complexity reduction in the tree-search process due to RRC. As the SNR goes higher, however, it may not be true since the DFE-radius itself can be already extremely small so that the ZF-DEF estimate could be equal to the ML estimate. In the above extreme case, the RRC process is not beneficial any more. Hence, we would adopt the radius reduction process adaptively according to the estimated SNR after ZF-DFE estimation.

For adaptivity, we modify Step 5 of the MSE algorithm as follows.

Step 5.

Set $r := T_1 + \left|y_1 - \xi_1 - r_{1,1}x_1\right|^2$, $\hat{\mathbf{x}} = \mathbf{x}$, $\gamma = \dfrac{\sum_{i=1}^{n}\left\|\mathbf{h}_i\right\|^2}{r}$

If γ < a threshold value, then perform the radius control process in (8); $i := i + 1$;
End
Go to Step 6.

In the above procedure, the estimated SNR after ZF-DFE estimation may be the joint contribution of the noise and the estimation error. Hence, only if the estimated SNR is lower than the threshold value, we perform the RRC process.

V. Computational Complexities

We evaluate computations for pre-processing which is required prior to tree-search processing and computations to get a lattice point at each dimension during tree-search processing.

A. QR decomposition of channel matrix

The modified Gram-Schmidt procedure [7] is used for QR decomposition of the real-valued channel matrix in (4). Table I shows the number of operations for QR decomposition.

Table I. Number of operations for QR decomposition

Computation	Multiplications	Divisions	Additions	Square Root
$\mathbf{H} = \mathbf{Q}\cdot\mathbf{R}$	$8MN^2$	$4MN$	$8MN^2 - 2N^2 - N$	$2N$

B. Column ordering of channel matrix

We evaluate the complexities for column-norm ordering and optimal ordering as described above, used to reduce further the overall complexity for SD by improving the accuracy of the initial ZF-DFE estimate.

1) Column-norm ordering

First, the norm of each column of the real channel matrix $\mathbf{H}$ is computed. Then, the columns of $\mathbf{H}$ and the elements of $\mathbf{x}$ are permutated in the ascending order of column norms. In addition, since the column norms of $\mathbf{H}$ are equal to those of $\mathbf{R}$, they can be reused for SNR calculation after ZF-DFE estimation and for the RRC process in (8). The operations for column-norm ordering are shown in the Table II.

Table II. Number of operations for column-norm ordering

Computation	Multiplications	Additions
$\left\|\mathbf{h}_k\right\|^2_{k=1,\ldots,n}$	$4MN$	$4MN - 2N$

2) Optimal ordering

In the original MSE algorithm [1], an optimal detection ordering scheme has been used. It finds a permutation matrix according to the following criterion as

$$\pi(k) = \arg\max_{j\in\Lambda_k}\left\{\mathbf{h}_j^T\left[\mathbf{I} - \mathbf{H}_{k,j}\left(\mathbf{H}_{k,j}^T\mathbf{H}_{k,j}\right)^{-1}\mathbf{H}_{k,j}^T\right]\mathbf{h}_j\right\}. \quad (10)$$

where Λ_k denotes the set of column indexes for the remaining columns not selected yet and $\mathbf{H}_{k,j}$ is $m\times(k-1)$ matrix formed by the columns $\mathbf{h}_i$ with $i \in \Lambda_k - \{j\}$; $k = n, n-1, \ldots, 1$. Table III shows the operations for optimal ordering.

Table III. Number of operations for optimal ordering

Computation	$\pi(k)_{k=n,\ldots,1}$
Multiplications	$\sum_{p=1}^{2N-1}\frac{5}{6}p^4 + 4Mp^3 - \frac{5}{6}p^2 + 4M^2p^2 + 4M^2p + 2Mp$
Additions	$\sum_{p=1}^{2N-1}\frac{5}{6}p^4 + 4Mp^3 - 2p^3 + 4M^2p^2 - 2Mp^2 + \frac{p^2}{6} - p$
Divisions	$\sum_{p=1}^{2N-1}p^3$

C. Tree-search process

Computations for tree-search processing are obtained by evaluating the number of operations to get a valid lattice point at each dimension, and then by counting the number of visited lattice points for respective dimensions by computer simulation.

VI. Simulation Results

We performed computer simulations to evaluate the average number of operations to get a lattice point at each dimension. In all simulations, we assumed that the channel matrix is kept fixed during a block of ten spatial snapshots. In what follows, we deal with only using 4 transmit and 4 receive antennas and consider 64-QAM modulations. For averaging the fading effect, we used 10,000 independent fading cases. We use the average number of multiplications as a performance measure and only count operations for tree-search processing without including operations for pre-processing such as QR decomposition, ordering, and so on, for Figs. 1 and 2. In Table IV, the effect of pre-processing on the overall complexity for SD is considered.

Fig. 1 shows the average number of multiplications for tree-search processing according to ordering schemes, in the case of 64-QAM modulation. In this figure, the number of multiplications is evaluated by first multiplying the average number of visited lattice points with the number of multiplications to get each point at each dimension and then summing up the number of total multiplications for each dimension over all the dimensions considered for tree search. If the RRC process is adopted, the number of multiplications for the RRC process is added to the complexity for SD. From Fig. 1, we see that using the proposed RRC scheme can reduce effectively the number of multiplications at low SNR values, but at high SNR values, the extra computations for RRC dominate the complexity for SD as expected. In addition, optimal ordering outperforms column-norm ordering irrespective of the adoption of the RRC process.

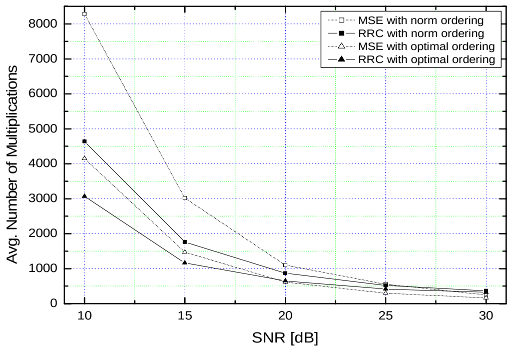

Fig. 1. Average number of multiplications for tree-search processing of the MSE and RRC algorithms according to ordering schemes. ($N=4, M=4$, 64-QAM).

Fig. 2 shows the average number of multiplications for tree-search processing according to ordering schemes when the SRC scheme and 64-QAM modulation are used. In this figure, the SRC algorithm can reduce effectively the complexity as compared to that of the MSE algorithm even at the high SNR values since the RRC process is not used any more when the estimated SNR values are greater than the specific threshold. The complexity of the SRC algorithm does not crossover those of the MSE algorithm any more irrespective of ordering schemes used. As threshold values for the SRC algorithm, 13 dB and 11 dB for norm ordering and optimal ordering, respectively, were used.

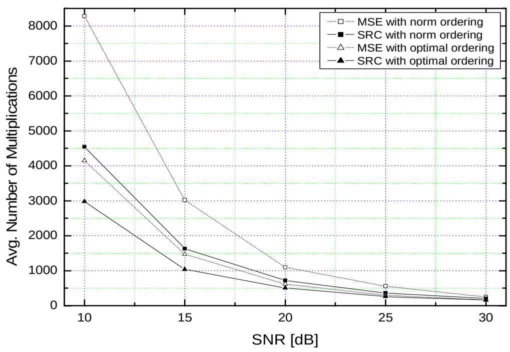

Fig. 2. Average number of multiplications for tree-search processing of the MSE and SRC algorithms according to ordering schemes. ($N=4, M=4$, 64-QAM).

In Table IV, we show the effect of the channel variability on the overall computations including pre-processing. In the table, we see that the number of operations for optimal ordering is quite larger than those of column-norm ordering. Especially, if the channel characteristics change independently at each time slot, then that for optimal ordering has a considerably high impact on overall computations. The impact gets more significant as SNR increases. But, if the channel characteristics are kept for a long time over 100 time slots, the complexity of optimal ordering gets smaller so that at low SNR values, the overall complexity with optimal ordering could be even smaller as compared to that with norm-ordering.

Table IV. Total Complexity of the SRC Algorithm according to Channel Variation. ($N=4, M=4$, 64QAM)

Computation		QR + norm ordering			QR + optimal ordering		
Timeslot		1	10	100	1	10	100
Preprocessing		576	57.6	5.76	27300	2730	273
Total complexity	10dB	5121	4603	4551	30278	5708	3251
	15dB	2209	1691	1639	28340	3769	1312
	20dB	1299	781	729	27807	3237	780
	25dB	937	419	367	27557	2987	531
	30dB	778	260	208	27459	2889	432

VII. Conclusions

In this paper, we proposed a computationally efficient SD algorithm with SRC. For RRC, after ZF-DFE estimation, we found a better lattice vector by performing a sequence of alternating one-dimensional searches, starting from the ZF-DFE estimate. Then, we developed a novel SRC algorithm in which the RRC process is adaptively adopted according to the estimated SNR after ZF-DFE estimation. In addition, we analyzed the effect of detection ordering strategies on the complexity for SD.

From analysis results, we conclude that SRC can reduce greatly the overall complexity of the MSE algorithm for SD over all SNR values since it reduce effectively only huge radii after ZF-DFE estimation through adaptive RRC, and the degree of complexity reduction gets significant as the SNR decreases, irrespective of detection ordering schemes used.

References

[1] M. O. Damen, H. E. Gamal and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389-2402, Oct. 2003.

[2] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. Expected complexity,” *IEEE Trans. Sig. Proc.*, vol. 53, pp. 2806-2818, Aug., 2005.

[3] H. Vikalo and B. Hassibi, “On the sphere-decoding algorithm II. Generalization, second-order statstics, and applications to communications ,” *IEEE Trans. Sig. Proc*., vol. 53, pp. 2819-2834, Aug., 2005.

[4] H. G. Han and S. K. Oh “The computational complexity of sphere decoding according to initial radius selection schemes an an efficient initial radius reduction scheme,” *accepted by IEEE GLOBECOM’05* .

[5] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel,” *in Proc. ISSSE’98*, pp. 295-300, Sept. 1998.

[6] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bölcdkei, “An overview of MIMO communications - a key to gigabit wireless,” *in Proc. IEEE*, vol. 92, pp. 198-218, Feb. 2004.

[7] S. J. Leon, *Linear algebra with applications*, Prentice-Hall, 6th ed. 2002.