# Towards the Performance of ML and the Complexity of MMSE - A Hybrid Approach for Multiuser Detection

Byonghyo Shim, *Senior Member, IEEE,*, Jun Won Choi, *Member, IEEE,*, and Insung Kang, *Member, IEEE,*

***Abstract*—In this paper, we consider a low-complexity multiuser detection for downlink of high speed packet access (HSPA) of universal mobile telecommunications system (UMTS). Instead of attempting to perform maximum likelihood (ML) detection of all users in multiple cells, which is impractical for battery powered mobile receiver, we utilize interference cancelled chips obtained from iterative linear minimum mean square error (LMMSE) estimation to perform a near ML detection in a reduced dimensional system. As a result, the worst case complexity of the detection process, achieved by the closest lattice point search (CLPS), is bounded to a controllable level irrespective of multipath spans. Furthermore, by exploiting the LMMSE estimate in tightening the hypersphere condition of the CLPS algorithm so called sphere decoding, we achieve significant improvement in search complexity. From simulations on realistic downlink communication scenario in HSPA systems, we show that the proposed method offers substantial performance gain over conventional receiver algorithms with reasonable complexity.**

***Index Terms*—Closest lattice point search, sphere decoding, multiuser detection, Babai point, noncausal path metric.**

## I. Introduction

Recently, increasing demand for high quality data services is fueling the deployment of high speed packet access (HSPA) mobile systems [1]. In order to support the promised data rate (42 Mbps) in multiuser and multipath fading conditions, a strong receiver algorithm combatting the intercell interference (ICI) and intersymbol interference (ISI) is critical. Fundamental studies on a joint detection strategy dealing with all interferences from multiple cells, so called multiuser detection (MUD), have demonstrated substantial gain in capacity and performance [2], [3]. In fact, ever since the Verdu's seminal work [2], there have been large amount of researches on the MUD for code division multiplexing access (CDMA) systems [3]–[5]. However, the MUD has not been considered as a suitable option for battery-powered mobile receiver due to its enormous computational overhead, and therefore, the MUD technique for downlink applications fell into oblivion over the years. As a cost effective alternative, single-user detection (SUD) methods such as RAKE and linear minimum mean square error (MMSE) equalizer have long been employed, and still popularly being used, mainly due to their low complexity and implementation simplicity [2], [3], [6]. While these schemes have been effective in mitigating interferences by averaging out chips with long spreading sequence, in the HSPA system employing the reduced spreading factor (SF) for achieving higher data rate[1], such philosophy is no more effective and thus improved receiver algorithm dealing with interferences is required.

When applying MUD in HSPA system, interferences from adjacent symbols, so called *inter-symbol interferences* (ISI) are no longer negligible so that proper control of the ISI along with multi-user interferences is critical. This makes design of the MUD even more challenging since a problem size handled by the joint detector becomes considerable (usually order of $10 \sim 100$). Indeed, this range of dimension is burdensome even for state of the art closest lattice point search (CLPS) technique so called *sphere decoding* (SD) algorithm. Note that while the SD algorithm has received much attention especially in the context of flat-fading multi-input multi-output (MIMO) system (e.g., $4 \times 4$ or $8 \times 8$ MIMO-OFDM system) [7], [8], application of the SD algorithm in high dimensional system is still demanding job [9]. On the contrary, symbol estimator with linear structure can achieve significant reduction in complexity over the CLPS. A linear MMSE (LMMSE) estimator [10] and a decision feedback LMMSE (DF-LMMSE) detector [11] are examples of such detectors. Although these schemes can be implemented in a reasonable complexity, performance gap from the ML detection is substantial.

In this paper, we propose a low-complexity joint detection technique, referred to as a closest lattice point search with noncausal path metric (CLPS-NP), for the downlink of the HSPA-based universal mobile telecommunications system (UMTS). Exploiting the fact that the space spanned by an observation vector in a downlink is well modeled by skewed lattice points corrupted by the noise, the proposed CLPS-NP searches for a near closest lattice point of the multiuser systems. In searching this point, instead of performing the strict ML detection for all multiuser symbols in multiple cells, which obviously is computationally prohibitive, we utilize the interference cancelled chips obtained from the low complexity LMMSE estimation. As a result, dimension of the lattice search operation is limited to the current symbols (at most product of active cells and number of users per cell), and thus

B. Shim is with School of Information and Communication, Korea University, Seoul, Korea (email: bshim@korea.ac.kr).

J. Choi and I. Kang is with Qualcomm Inc., CA 92121 USA (email: {junwonc,insungk}@qualcomm.com).

This work is supported by Basic Science Research Program through National Research Foundation (NRF) funded by the Ministry of Education, Science and Technology (MEST) (No. 2011-0012525) and second Brain Korea 21 project.

[1]While the SF of the data channel in WCDMA is varying from 4 to 256, SF of data channel (HS-PDSCH) in HSPA is fixed to 16 [1].

the search complexity of the CLPS-NP is reduced significantly compared to the ML detection using all users. In addition to the search dimension reduction, the CLPS-NP exploits the information obtained from the LMMSE estimation to improve the computational complexity of the lattice search operation.

A key ingredient of the proposed CLPS-NP is the two-stage symbol detections consisting of 1) the first stage *iterative LMMSE estimation* and 2) the second stage *lattice search with noncausal path metric* exploiting the LMMSE estimate. While the purpose of the first stage LMMSE estimation is to maximize the signal-to-interference-plus-noise ratio (SINR) seen by users being detected by cancelling out the intersymbol interference, the role of the second stage is to incorporate interference estimates obtained from the first stage into the path metric for speeding up the CLPS operation. There have been some works that take into account noncausal part of the path metric to improve the CLPS techniques [12], [13]. Also, approaches approximating a noncausal part in an optimistic scenario have been suggested in [14], [15]. While these approaches require considerable amount of run-time computations by performing direct estimation of noncausal symbols [13] or their lower bound calculation [12], our technique recycles part of symbols from the iterative LMMSE estimation stage, and hence, virtually no additional cost is required in the CLPS-NP technique. Due to the tightening of the hypersphere condition via the noncausal path metric, unpromising paths are promptly pruned from the search tree and thus we can achieve substantial reduction in the search effort. In fact, from the complexity analysis and simulations on realistic HSPA scenarios, we demonstrate that the proposed CLPS-NP achieves excellent performance gain over the conventional MUD techniques with surprisingly small computational complexity.

The outline of the paper is as follows. In Section II, basic model for the HSPA downlink and various CLPS options for the MUD receiver are introduced. The first part (iterative LMMSE estimation) and the second part (CLPS-NP) of the proposed method are presented in Section III and Section IV, respectively. The simulation results are presented in Section VI and the conclusion is provided in Section VII.

## II. Multiuser Detection in HSPA Downlink

### A. Downlink System Model

Let $N_{SF}$ be the spreading factor and $N_c$ be the number of cells in the system, then the chip-level received signal vector $\mathbf{r} \in \mathcal{C}^{3N_{SF}}$ of a downlink in the HSPA system is given by

$$\mathbf{r} = \sum_{i=1}^{N_c} \mathbf{T}_i \mathbf{s}_i + \mathbf{v} = \sum_{i} \mathbf{H}_i \mathbf{C}_i \mathbf{W} \mathbf{G}_i \mathbf{s}_i + \mathbf{v} \qquad (1)$$

where $\mathbf{H}_i \in \mathcal{C}^{(N_{SF}+N_h-1)\times(N_{SF}+2N_h-2)}$, $\mathbf{C}_i \in \mathcal{C}^{(N_{SF}+2N_h-2)\times(N_{SF}+2N_h-2)}$, and $\mathbf{G}_i \in \mathcal{C}^{3N_{SF}\times 3N_{SF}}$ are the channel, scrambling code, and user gain matrices for a cell $i$, $\mathbf{W} \in \mathcal{R}^{(N_{SF}+2N_h-2)\times 3N_{SF}}$ is the Hadamard matrix (so called channelization code matrix) and $\mathbf{T}_i = \mathbf{H}_i \mathbf{C}_i \mathbf{W} \mathbf{G}_i$ is the composite system matrix of the cell $i$ (see Fig. 1). In addition, $\mathbf{s}_i$ is $3N_{SF}$-dimensional symbol vector of users in a cell $i$ and $\mathbf{v} \sim \mathcal{CN}(\mathbf{0}, \sigma_v^2 \mathbf{I})$ is the background Gaussian noise vector, respectively.

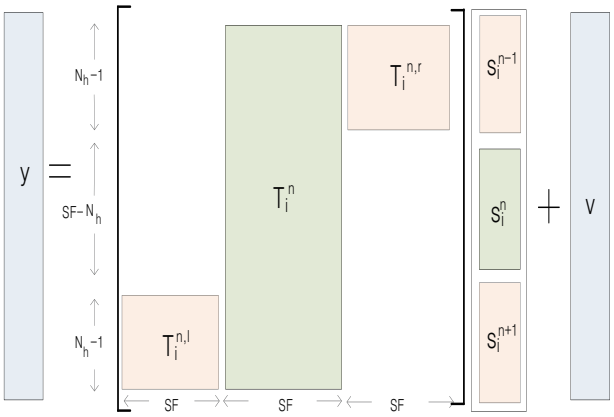



Fig. 1. Received signal structure for the HSPA downlink with a single cell transmission when $N_h \leq N_{SF}$.

From the viewpoint of detecting a symbol in desired cell $i$, there are two important sources of hindrance: First, the signal vector coming from other cells (i.e., $\sum_{j\neq i} \mathbf{T}_j \mathbf{s}_j$) is unwanted interference and commonly referred to as an *inter-cell interference*. Second, for most physical channels in the HSPA system, orthogonal spreading codes are assigned, thereby creating mutually orthogonal downlink signals. However, orthogonality of the channels can no longer be maintained at the receiver for dispersive multipath channels, inducing an *intracell multiuser interference*. In fact, when the multipath spans $N_h$ chips, the length of $\mathbf{r}$ becomes $N_{SF} + N_h - 1$ and thus $N_h - 1$ chips of previous and next symbols in a desired cell smear into the current symbol, giving rise to the intracell interference called ISI.[2] In this situation where the delay spread causes ISI, it would be convenient to divide the composite system matrix $\mathbf{T}_i$ into three submatrices: $\mathbf{T}_i^{n,l}$, $\mathbf{T}_i^n$, and $\mathbf{T}_i^{n,r}$, each of which represents the composite system matrix for previous, current, and next symbols ($\mathbf{s}_i^{n-1}$, $\mathbf{s}_i^n$, and $\mathbf{s}_i^{n+1}$) in a cell $i$, respectively (see Fig. 1). In this setup, (1) can be rewritten as

$$\begin{aligned} \mathbf{r} &= \sum_{i=1}^{N_c} \mathbf{T}_i \mathbf{s}_i + \mathbf{v} \\ &= \sum_{i=1}^{N_c} \left( \mathbf{T}_i^{n,l} \mathbf{s}_i^{n-1} + \mathbf{T}_i^n \mathbf{s}_i^n + \mathbf{T}_i^{n,r} \mathbf{s}_i^{n+1} \right) + \mathbf{v} \end{aligned} \qquad (2)$$

where $\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_i^{n,l} & \mathbf{T}_i^n & \mathbf{T}_i^{n,r} \end{bmatrix}$ and $\mathbf{s}_i = \begin{bmatrix} \mathbf{s}_i^{n-1} \\ \mathbf{s}_i^n \\ \mathbf{s}_i^{n+1} \end{bmatrix}$.

### B. Multiuser Detection using Closest Lattice Point Search (CLPS)

An ideal detection strategy under the existence of the interferences would be joint detection of all symbols contributing to the observation vector $\mathbf{r}$. In other words, symbols of all previous and next symbols of the desired cell and interfering cells (collectively called multiuser information) are desired

[2]We henceforth use the term ISI exclusively for denoting previous and next symbols of every cells in the system.

for the best detection of users of interest. Depending on the level of accuracy of a system model being employed, detection strategies can be classified as follows:

1) (*ML multiuser detection*) The maximum-likelihood multiuser detection (ML-MUD) of (2) is described as

$$\max_{\mathbf{s}_1\in\mathcal{X}_1^{3N_{SF}},\cdots,\mathbf{s}_{N_c}\in\mathcal{X}_{N_c}^{3N_{SF}}} P(\mathbf{r}\,|\,\mathbf{T}_1,\cdots,\mathbf{s}_1,\cdots,\mathbf{s}_{N_c})$$
$$= \min_{\mathbf{s}_1\in\mathcal{X}_1^{3N_{SF}},\cdots,\mathbf{s}_{N_c}\in\mathcal{X}_{N_c}^{3N_{SF}}} \parallel \mathbf{r}-\sum_{i=1}^{N_c}\mathbf{T}_i\,\mathbf{s}_i\parallel^2 \qquad (3)$$

where $\mathcal{X}_i^n$ is the constellation set of size $n$ for a cell $i$. For the HSPA system ($N_{SF}=16$) having 16-QAM modulation and three cells ($N_c=3$), the number of lattice points becomes $16^{144}\approx 2.5\times10^{173}$. Although the solution of this closest lattice point search (CLPS) equals the true maximum likelihood (ML) solution [16], needless to say, the complexity associated with this problem is astronomical and hence computationally infeasible.

2) (*Single-cell approximation*) We regard the signals of all interfering cells plus noise as an effective noise in the system. In this case, the system model becomes

$$\mathbf{r}=\mathbf{T}_i\mathbf{s}_i+\mathbf{w} \qquad (4)$$

where $\mathbf{w}=(\sum_{j\neq i}\mathbf{T}_j\,\mathbf{s}_j+\mathbf{v})$ and the CLPS is simplified to

$$\min_{\mathbf{s}_i\in\mathcal{X}_i^{3N_{SF}}} \parallel \mathbf{r}-\mathbf{T}_i\,\mathbf{s}_i\parallel^2. \qquad (5)$$

The rationale behind this model is that the sum of interferences coming from multiple cell locations can be roughly assumed to be white. Although the worst complexity associated with this problem is much lower than that in (3), the CLPS output is by no means an ML solution since the effective noise $\mathbf{w}$ is colored. Further, the signal-to-interference-noise ratio (SINR) in this problem is given by $\frac{E[|\mathbf{T}_i\mathbf{s}_i|^2]}{\sum_{j\neq i}E[|\mathbf{T}_j\mathbf{s}_j|^2]+\sigma_v^2}$, which is clearly much smaller than $\frac{\sum_i E[|\mathbf{T}_i\mathbf{s}_i|^2]}{\sigma_v^2}$ in (3). Recalling that the search complexity of the CLPS depends strongly on the SINR of the systems, the option is still burdensome for practical purpose.

3) (*Approximation with only current symbols*) One can ignore the ISI and focus only on the current-time symbols. The CLPS for this scenario becomes

$$\min_{\mathbf{s}_1\in\mathcal{X}_1^{N_{SF}},\cdots,\mathbf{s}_{N_c}\in\mathcal{X}_{N_c}^{N_{SF}}} \parallel \mathbf{r}-\sum_{i=1}^{N_c}\mathbf{T}_i^n\mathbf{s}_i^n\parallel^2. \qquad (6)$$

Using this approximation, the number of lattice points is reduced to $16^{48}\approx 6.3\times10^{57}$, achieving $\frac{1}{10^{116}}$ of the original number of lattice points. Although this reduction is phenomenal, this approach is undesirable since the ISI is not properly controlled in this problem. Furthermore, the complexity of this model is still overwhelming.

From the discussion thus far, we infer that two key requirements for algorithm achieving performance close to the ML-MUD with implementable complexity are the reduction of the CLPS dimension and the mitigation of interferences not included in the CLPS. The proposed method addresses these issues by controlling ISI using the cost effective LMMSE estimation iteratively. To be specific, we employ an iterative LMMSE estimation in the first stage to obtain the estimate of multiuser information. By cancelling out the estimated ISI, in the second stage, we can employ the CLPS of (6) in an improved SINR condition. Furthermore, we attempt to use by-products of the first stage to tighten the hypersphere condition of the CLPS. While the conventional SD algorithm employs a path metric generated from causal symbols only, the proposed method accounts for the impact of noncausal symbols from the LMMSE estimates in the path metric generation of the CLPS. In the next two sections, we describe the LMMSE based iterative interference cancellation and the CLPS-NP exploiting these LMMSE estimates.

## III. Iterative LMMSE Estimation

The goal of the iterative LMMSE estimation is to generate multiuser information being used for the second stage. Part of multiuser information (reconstructed ISI) is used for interference cancellation and the rest is used for the CLPS-NP. Our iterative LMMSE estimation, on the contrary to the "one time" LMMSE operation, combines the LMMSE estimation with the ICI cancellation. That is, starting from the strongest cell, the contribution for each cell is being removed successively and this process is performed iteratively (see Table I). In doing so, residual interference level gets smaller, the symbol estimate becomes more accurate, and hence better performance can be achieved [2], [17]. We note that this step not only helps improving the detection quality of the second stage CLPS, but also bring forth significant reduction in search complexity.

### A. Transmit Signal Estimation

Although (1) is a form adequate for the symbol detection, it is not appropriate to express the LMMSE estimation since the construction of $\mathbf{T}_i$ requires great deal of computations in connecting channel estimate, scrambling code, Hadamard matrix, as well as the gain of physical channels. Hence, instead of using (1) to obtain $\mathbf{s}_i$, we estimate the transmitted chip signal $\mathbf{x}_i$ in the following signal model

$$\mathbf{r}=\sum_{i=1}^{N_c}\mathbf{r}_i+\mathbf{v}=\sum_{i=1}^{N_c}\mathbf{H}_i\mathbf{x}_i+\mathbf{v} \qquad (7)$$

where $\mathbf{x}_i=\mathbf{C}_i\mathbf{W}\mathbf{G}_i\mathbf{s}_i=\mathbf{C}_i\mathbf{W}\mathbf{u}_i$. In essence, the received signal is a linear convolution between the transmit chip vector $\mathbf{x}_i$ and channel impulse response (a row of toeplitz matrix $\mathbf{H}_i$).

The LMMSE estimate $\hat{\mathbf{x}}_i$ of $\mathbf{x}_i$ becomes [18]

$$\hat{\mathbf{x}}_i=R_{\mathbf{x}_i\mathbf{r}}R_{\mathbf{r}\mathbf{r}}^{-1}(\mathbf{r}-E[\mathbf{r}]). \qquad (8)$$

Using the following property (see Appendix A)

$$E[\mathbf{x}_j\mathbf{x}_j^H]=E[\mathbf{C}_j\mathbf{W}\mathbf{G}_j\mathbf{s}_j\mathbf{s}_j^H\mathbf{G}_j\mathbf{W}^T\mathbf{C}_j^H]=tr(\mathbf{G}_j^2)\mathbf{I} \qquad (9)$$

and $E[|\mathbf{v}|^2] = \sigma_v^2\mathbf{I}$, the crosscorrelation between $\mathbf{r}$ and $\mathbf{x}_i$ becomes

$$R_{\mathbf{x}_i\mathbf{r}} = E\left[\mathbf{x}_i\left(\sum_{j=1}^{N_c}\mathbf{H}_j\mathbf{x}_j + \mathbf{v}\right)^H\right] = tr(\mathbf{G}_i^2)\mathbf{H}_i^H.$$

Similarly, the autocorrelation of $\mathbf{r}$ is

$$\begin{aligned} R_{\mathbf{rr}} &= E\left[\left(\sum_{j=1}^{N_c}\mathbf{H}_j\mathbf{x}_j + \mathbf{v}\right)\left(\sum_{j=1}^{N_c}\mathbf{H}_j\mathbf{x}_j + \mathbf{v}\right)^H\right] \\ &= \left(\sum_{j=1}^{N_c} tr(\mathbf{G}_j^2)\mathbf{H}_j\mathbf{H}_j^H + \sigma_v^2\mathbf{I}\right). \end{aligned}$$

These together with the fact that $E[\mathbf{r}] = 0$, $\hat{\mathbf{x}}_i$ becomes

$$\hat{\mathbf{x}}_i = R_{\mathbf{x}_i\mathbf{r}}R_{\mathbf{rr}}^{-1}\mathbf{r} = tr(\mathbf{G}_i^2)\mathbf{H}_i^H R_{\mathbf{rr}}^{-1}\mathbf{r} \tag{10}$$

Noting that $\mathbf{C}_i$ and $\mathbf{W}$ are unitary and orthogonal matrices, respectively, the estimate of $\mathbf{u}_i = \mathbf{G}_i\mathbf{s}_i$ becomes

$$\hat{\mathbf{u}}_i = \mathbf{W}\mathbf{C}_i^H\hat{\mathbf{x}}_i. \tag{11}$$

Once scrambling code $\mathbf{C}_i$ and orthogonal spreading code $\mathbf{W}$ are removed, we can model (gain included) symbol vector as $\hat{\mathbf{u}}_i = \mathbf{G}_i\mathbf{s}_i + \eta$ where $\eta$ is the composite of the interference and noise of the code channel. For each element of $\hat{\mathbf{u}}_i$ which corresponds to physical channel estimate in the HSPA system, the slicing is done per symbol basis. After the slicing, the sliced symbol vector $\tilde{\mathbf{u}}_i$ is retransmitted for the cancellation. As shown in Fig. 2, reconstruction follows the footstep of the basestation transmission and consists of scrambling, inverse Hadamard transform, and the convolution with the channel impulse response.

### *B. Soft Symbol Slicing*

The estimate of a symbol for each physical channel at the symbol time $n$ becomes

$$\hat{u}_i[n] = u_i[n] + \eta[n] = g_i s_i[n] + \eta[n] \tag{12}$$

Since $s_i[n]$ is constant for the pilot channel ($s_i[n] = \frac{1+j}{\sqrt{2}}$) [1], $\frac{1}{2}E[|\,\hat{u}_i[n] - \hat{u}_i[n-1]\,|^2] = \sigma_\eta^2$ where $\sigma_\eta^2 = E[|\eta|^2]$. Also, noting that $E[|\,\hat{u}_i\,|^2] = g_i^2 + \sigma_\eta^2$, the gain estimate for each physical channel is obtained as

$$\hat{g}_i = \sqrt{E[|\,\hat{u}_i\,|^2] - \sigma_\eta^2}. \tag{13}$$

A straightforward way to slice the symbol is a hard decision given by

$$\tilde{u}_i[n] = \hat{g}_i\left(\arg\min_s |\,\hat{u}_i[n] - \hat{g}_i s\,|\right). \tag{14}$$

A drawback of the hard decision is the error propagation caused by incorrectly detected symbols [19]. As an alternative way, one can consider the nonlinear MMSE-based soft slicing

$$\begin{aligned} \tilde{u}_i[n] &= E[u_i\,|\,\hat{u}_i] \\ &= \frac{\sum_{u_i} u_i P(\hat{u}_i|u_i)P(u_i)}{\sum_{u_j} P(\hat{u}_j|u_j)P(u_j)}. \end{aligned} \tag{15}$$

Using an equal prior assumption on $P(u_i)$ and the Gaussian model of $\eta$, one can easily show that (15) becomes

$$\tilde{u}_i[n] = \frac{\sum_s \hat{g}_i s\exp\left(-\frac{|\hat{u}_i[n]-\hat{g}_i s|^2}{2\sigma_\eta^2}\right)}{\sum_s \exp\left(-\frac{|\hat{u}_i[n]-\hat{g}_i s|^2}{2\sigma_\eta^2}\right)}. \tag{16}$$

For a high SNR physical channel ($\frac{\hat{g}_i}{\sigma_\eta^2} \gg 1$), the MMSE estimate approximates to the hard decision since a single exponential term gets close to 1 while all others are negligible. On the contrary, since no dominant term exists for a low SNR physical channel, $\tilde{u}_i[n]$ is expressed as the sum of several terms and thus the magnitude of $\tilde{u}_i[n]$ will be moderately small. Due to the suppression of unreliable symbol, the MMSE based soft slicing is in general more robust to the error propagation than the hard decision slicing.

### *C. Iterative Estimation Process*

By combining the reconstruction of the received vector with an iterative successive cancellation, we can improve the quality of estimated symbols. Note that the basic concept behind the successive interference cancellation is well known in information theory [20, Ch. 15.1]. While the theory assumes an ideal scenario of perfect interference cancellation, an iteration loop on top of the successive cancellation is required in practical communication system due to the imperfection of cancellation. To be specific, starting from the strongest cell, we perform the symbol generation and retransmission followed by the cancellation. As is clear from the previous works [3], [21], canceling the interference in a descending order minimizes the error propagation. This retransmission followed by the cancellation is repeated until the final (weakest) cell, completing one iteration of successive cancellation. For the re-detection of a cell $i$ at $\ell$-th iteration, we need to add back the retransmit signals of the cell $i$. Thus, an observation vector $\mathbf{r}_i$ used for the re-detection of $i$-th cell becomes

$$\begin{aligned} \mathbf{r}_i &= \mathbf{r} - \left(\sum_{j<i}\mathbf{H}_j\hat{\mathbf{x}}_j^{(\ell)} + \sum_{j>i}\mathbf{H}_j\hat{\mathbf{x}}_j^{(\ell-1)}\right) \\ &= \mathbf{H}_i\mathbf{x}_i + \sum_{j<i}\mathbf{H}_j\epsilon_j^{(\ell)} + \sum_{j>i}\mathbf{H}_j\epsilon_j^{(\ell-1)} \end{aligned} \tag{17}$$

where $\hat{\mathbf{x}}_j^{(\ell)}$ denotes the chip estimate generated from $\ell$-th iteration (assuming an initial condition $\hat{\mathbf{x}}_j^{(0)} = 0$) and $\epsilon_i^{(\ell)} = \mathbf{x}_i - \hat{\mathbf{x}}_i^{(\ell)}$. The reason of articulating an iteration index $\ell$ is to emphasize that all estimates are the most recent ones. In fact, depending on the order of cell, some estimates are from $\ell$-th detection and others are from $(\ell-1)$-th detection. We illustrate the cancellation process in Table I.

## IV. CLOSEST LATTICE POINT SEARCH WITH NONCAUSAL PATH METRIC (CLPS-NP)

Once all multiuser information is estimated using the LMMSE equalizer, the information from the LMMSE equalizer is exploited in the subsequent CLPS step to improve the detection performance and complexity. Two key features of the

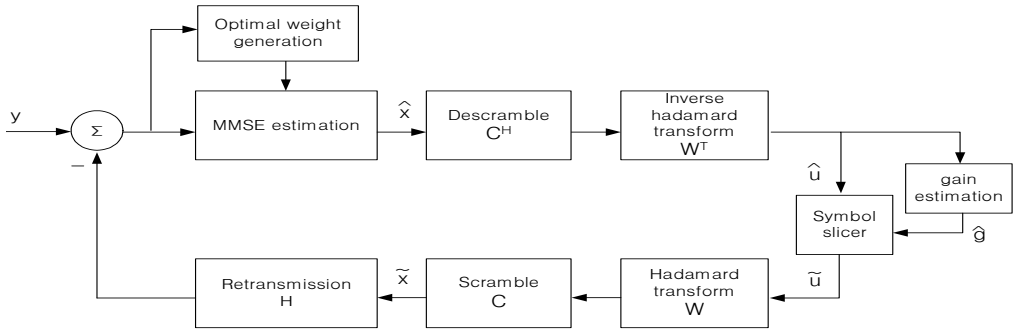



Fig. 2. Structure of the iterative MMSE estimation stage.

TABLE I
ITERATIVE LMMSE ESTIMATION PROCESS FOR THREE CELLS AND TWO ITERATIONS

| cell index | observation | estimate |
|---|---|---|
| 1 | $\mathbf{r}_1 = \mathbf{r}$ | $\hat{\mathbf{x}}_1^{(1)}$ |
| 2 | $\mathbf{r}_2 = \mathbf{r}_1 - \mathbf{H}_1\hat{\mathbf{x}}_1^{(1)}$ | $\hat{\mathbf{x}}_2^{(1)}$ |
| 3 | $\mathbf{r}_3 = \mathbf{r}_2 - \mathbf{H}_2\hat{\mathbf{x}}_2^{(1)}$ | $\hat{\mathbf{x}}_3^{(1)}$ |
| 1 | $\mathbf{r}_1 = \mathbf{r}_3 - \mathbf{H}_3\hat{\mathbf{x}}_3^{(1)} + \mathbf{H}_1\hat{\mathbf{x}}_1^{(1)}$ | $\hat{\mathbf{x}}_1^{(2)}$ |
| 2 | $\mathbf{r}_2 = \mathbf{r}_1 - \mathbf{H}_1\hat{\mathbf{x}}_1^{(2)} + \mathbf{H}_2\hat{\mathbf{x}}_2^{(1)}$ | $\hat{\mathbf{x}}_2^{(2)}$ |
| 3 | $\mathbf{r}_3 = \mathbf{r}_2 - \mathbf{H}_2\hat{\mathbf{x}}_2^{(2)} + \mathbf{H}_3\hat{\mathbf{x}}_3^{(1)}$ | $\hat{\mathbf{x}}_3^{(2)}$ |

proposed CLPS-NP are as follows. First, by substituting the strong colored interference for the small residual error (mostly an MMSE estimation error), the interference approximates to white signal and thus the CLPS solution gets close to the ML solution. The signal model after the ISI cancellation is

$$\mathbf{r}' = \mathbf{r} - \sum_{i=1}^{N_c}\left(\mathbf{T}_i^{n,l}\hat{\mathbf{s}}_i^{n-1} + \mathbf{T}_i^{n,r}\hat{\mathbf{s}}_i^{n+1}\right) \approx \sum_{i=1}^{N_c}\mathbf{T}_i^n\mathbf{s}_i^n + \mathbf{v}' \quad (18)$$

where $\mathbf{v}'$ contains the residual interferences as well as the noise ($\mathbf{v}' = \sum_{i=1}^{N_c}\mathbf{T}_i^{n,l}(\mathbf{s}_i^{n-1} - \hat{\mathbf{s}}_i^{n-1}) + \mathbf{T}_i^{n,r}(\mathbf{s}_i^{n+1} - \hat{\mathbf{s}}_i^{n+1})$) and the corresponding CLPS problem becomes

$$\min_{\mathbf{s}_1^n \in \mathcal{X}_1^{N_{SF}}, \cdots, \mathbf{s}_{N_c}^n \in \mathcal{X}_{N_c}^{N_{SF}}} \left\| \mathbf{r}' - \sum_{i=1}^{N_c}\mathbf{T}_i^n\mathbf{s}_i^n \right\|^2. \quad (19)$$

Although it seems that (19) is equivalent to (6), it is clearly distinct from (6) due to the use of interference cancelled observation $\mathbf{r}'$. In fact, since (19) is performed in an improved SINR condition, outputs are more reliable and also generated much faster than those of (6). Second, the LMMSE estimate is exploited to speed up the CLPS. At the heart of the proposed approach lies deliberately designed path metric jointly employing causal path metric and noncausal path metric generated from estimated symbols. Previous attempts to exploit noncausal path metric incur additional cost from the on-line computation of the LMMSE estimate [12], [13] or bring mild pruning effect due to the benign assumption that noncausal path metric contains only Gaussian noise [14], [15]. Whereas, the noncausal path metric of the proposed approach jointly uses the estimated symbols in the previous stage and detected symbols during the search. Hence, fairly accurate noncausal path metric can be obtained with small computational overhead.

In what follows, we use the notation $\tilde{\mathbf{s}}$ and $\hat{\mathbf{s}}$ to distinguish the symbols found by the sphere search and estimated in the previous stage, respectively. For the mixture of the detected and estimated symbols, we use the notation $\bar{\mathbf{s}}_1^m$. Also, we denote $\mathbf{s} = \left[\begin{array}{ccc}(\mathbf{s}_1^n)^H & \cdots & (\mathbf{s}_{N_c}^n)^H\end{array}\right]^H$ and $\mathbf{T} = \left[\begin{array}{ccc}\mathbf{T}_1^n & \cdots & \mathbf{T}_{N_c}^n\end{array}\right]$ for notational simplicity.

### A. SD algorithm and its Drawback

In this subsection, we describe a brief summary of the SD algorithm for motivating our work (see [16], [22] for details). For simplicity, we assume a real system model in the sequel. The SD algorithm performs a lattice point search within a hypersphere and returns $\mathbf{s}^* = \min_{\mathbf{s}} \|\mathbf{r}' - \mathbf{T}\mathbf{s}\|^2$ which corresponds to the integer least squares solution. In particular, when a system is expressed as $\mathbf{r}' = \mathbf{T}\mathbf{s} + \mathbf{v}$ where $\mathbf{T}$ and $\mathbf{s}$ are $n \times m$ system matrix and $m \times 1$ finite alphabet symbol vector, respectively, and $\mathbf{v} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, then $\mathbf{s}^*$ becomes the ML solution.

In finding the closest lattice point $\mathbf{T}\mathbf{s}^*$ to an observation vector $\mathbf{r}'$, the SD algorithim exploits $n$-dimensional hypersphere $S(\mathbf{r}', \sqrt{d})$ with radius $\sqrt{d}$ centered at $\mathbf{r}'$. A necessary condition for searching the solution is

$$\|\mathbf{r}' - \mathbf{T}\mathbf{s}\|^2 \le d. \quad (20)$$

In order for the systematic search, a linear system is converted into the real-valued one and then QR-decomposition of real-valued system matrix $\tilde{\mathbf{T}} = \begin{bmatrix}\Re(\mathbf{T}) & -\Im(\mathbf{T}) \\ \Im(\mathbf{T}) & \Re(\mathbf{T})\end{bmatrix}$ is applied. That is, $\tilde{\mathbf{T}} = [\mathbf{Q}\ \ \mathbf{U}]\begin{bmatrix}\mathbf{R} \\ \mathbf{0}\end{bmatrix}$ where $\mathbf{R}$ is an $m \times m$ upper triangular matrix with positive diagonal elements, $\mathbf{0}$ is an $(n-m) \times m$ zero matrix, and $\mathbf{Q}$ and $\mathbf{U}$ are $n \times m$ and $n \times (n-m)$ unitary matrices. Then (20) becomes $\|\mathbf{y} - \mathbf{R}\mathbf{s}\|^2 \le d_0$ where $\mathbf{y} = \mathbf{Q}^T\mathbf{r}'$ and $d_0 = d - \|\mathbf{U}^T\mathbf{r}'\|^2$. Further, by denoting a branch metric at layer $l - j + 1$ as

$$B_j(\mathbf{s}_j^l) = \left(y_k - \sum_{t=j}^{l} r_{k,t}s_t\right)^2, \quad (21)$$

the sum of path metric, often referred to as the path metric, becomes

$$J(\mathbf{s}) = \| \mathbf{y} - \mathbf{R}\mathbf{s} \|^2 = \sum_{j=1}^{l} B_j(\mathbf{s}_j^l), \tag{22}$$

where $r_{k,t}$ is the $(k,t)$th entry of $\mathbf{R}$.

The SD algorithm can be commonly interpreted as the branch and bound (BB) based tree search algorithm [23]. In the first layer, i.e., the bottom row of $\mathbf{s}$, candidates satisfying $B_l(s_l) \leq d_0$ are found (bounding). Once this step is finished, moving to the next layer of the best candidate (branching), $s_{l-1}$ satisfying $B_{l-1}(\mathbf{s}_{l-1}^l) + B_l(s_l) \leq d_0$ is searched. By repeating this step and updating the radius whenever a new lattice point $\mathbf{R}\mathbf{s}$ is found, the SD algorithm outputs the solution $\mathbf{s}^*$ for which the path metric $J(\mathbf{s}) = \| \mathbf{y} - \mathbf{R}\mathbf{s} \|^2 = \sum_{j=1}^{l} B_j(\mathbf{s}_j^l)$ is minimized.

It is worth comparing actually used (causal) hypersphere condition at $(m-k+1)$-th layer

$$J(\mathbf{s}_k^m) = B_k + B_{k+1} + \cdots + B_m \leq d_0 \tag{23}$$

and the original hypersphere condition

$$J(\mathbf{s}_1^m) = B_1 + \cdots + B_{k-1} + B_k + \cdots + B_m \leq d_0. \tag{24}$$

Since the branch metrics of unvisited nodes $k-1, \cdots, 1$ are unavailable, the causal hypersphere condition in (23), which is only necessary condition of (24), is used as a pruning criterion in the lattice search. In particular, this necessary condition is quite loose for initial layers. Therefore, even though the path metric $J(\mathbf{s}_k^m)$ is fairly large and thus we are sure that the path is unlikely to survive, we cannot remove this path simply because it is smaller than $d_0$. Indeed, loose pruning conditions lead to many back-tracking operations during the search, thereby increasing search complexity substantially. Shortcomings of the SD algorithm, in view of the search efficiency, can be summarized as follows:

- **Loose comparison**: Since the contribution of the partial layers $(k, \cdots, m)$ is employed, pruning operations in early layers are quite inefficient (e.g., $J(s_m) \leq d_0$ at the root node). As the search moves on to the final layers, the hypersphere condition gets tighter.
- **Inefficient use of information**: Due to the causality of the search, observations of unvisited layers $(y_1, \cdots, y_{k-1})$ are not considered in the path metric computation, which inhibits pruning of unpromising paths.

Although the SD algorithm provides an exact solution of the CLPS problem, due to aforementioned reasons, considerable amount of computations is wasted. In the next subsection, we describe an approach tightening the hypersphere condition by exploiting the estimate $\hat{\mathbf{s}}$ obtained from the previous stage.

### *B. Tightening of Hypersphere Condition with CLPS-NP*

With a reference to the current node $k$ being searched, the path metric is expressed as

$$J(\mathbf{s}) = \|\mathbf{y} - \mathbf{R}\mathbf{s}\|^2 = \|\mathbf{y} - (\mathbf{R}_l \mathbf{s}_1^{k-1} + \mathbf{R}_r \mathbf{s}_k^m)\|^2 \tag{25}$$

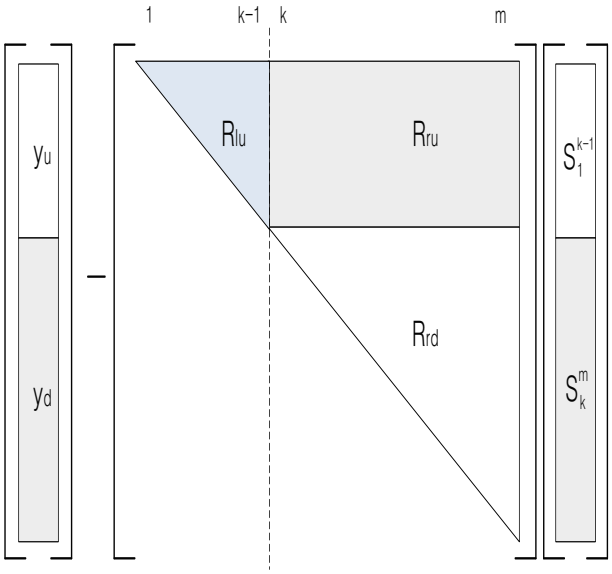



Fig. 3. The structure of $\mathbf{y} - \mathbf{R}\mathbf{s}$. $\mathbf{s}_1^{k-1}$ and $\mathbf{s}_k^m$ correspond to causal and noncausal symbol vector, respectively.

where $\mathbf{R} = [\mathbf{R}_l \ \mathbf{R}_r]$, $\mathbf{s}_k^m = \begin{bmatrix} s_k \\ \vdots \\ s_m \end{bmatrix}$, $\mathbf{s}_1^{k-1} = \begin{bmatrix} s_1 \\ \vdots \\ s_{k-1} \end{bmatrix}$, and $\mathbf{s} = \mathbf{s}_1^m = \begin{bmatrix} \mathbf{s}_1^{k-1} \\ \mathbf{s}_k^m \end{bmatrix}$. Furthermore, with $\mathbf{y} = \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_d \end{bmatrix}$, $\mathbf{R}_l = \begin{bmatrix} \mathbf{R}_{lu} \\ \mathbf{0} \end{bmatrix}$, and $\mathbf{R}_r = \begin{bmatrix} \mathbf{R}_{ru} \\ \mathbf{R}_{rd} \end{bmatrix}$, we have

$$\begin{aligned} J(\mathbf{s}) &= \left\| \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_d \end{bmatrix} - \begin{bmatrix} \mathbf{R}_{lu} & \mathbf{R}_{ru} \\ \mathbf{0} & \mathbf{R}_{rd} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^{k-1} \\ \mathbf{s}_k^m \end{bmatrix} \right\|^2 \\ &= \left\| \mathbf{y}_u - \begin{bmatrix} \mathbf{R}_{lu} & \mathbf{R}_{ru} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^{k-1} \\ \mathbf{s}_k^m \end{bmatrix} \right\|^2 + \|\mathbf{y}_d - \mathbf{R}_{rd}\mathbf{s}_k^m\|^2. \end{aligned} \tag{26}$$

Refer to the Fig. 3 for details. Let $J_u(\mathbf{s}_1^{k-1}, \mathbf{s}_k^m) = \left\| \mathbf{y}_u - \begin{bmatrix} \mathbf{R}_{lu} & \mathbf{R}_{ru} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^{k-1} \\ \mathbf{s}_k^m \end{bmatrix} \right\|^2$ and $J_d(\mathbf{s}_k^m) = \|\mathbf{y}_d - \mathbf{R}_{rd}\mathbf{s}_k^m\|^2$, then

$$J(\mathbf{s}) = J_u(\mathbf{s}_1^{k-1}, \mathbf{s}_k^m) + J_d(\mathbf{s}_k^m) = J_u(\mathbf{s}_1^m) + J_d(\mathbf{s}_k^m). \tag{27}$$

In a nutshell, our approach tightens the necessary condition of the sphere search by using the LMMSE estimate for the noncausal symbols $\mathbf{s}_1^{k-1}$ in (27). Consequently, the hypersphere condition of the original SD, $J_d(\tilde{\mathbf{s}}_k^m) \leq d_0$ is modified to $J_u(\bar{\mathbf{s}}_1^m) + J_d(\tilde{\mathbf{s}}_k^m) \leq d_0$. To be specific, since the node $k$ is currently visited, we already have symbols $\tilde{\mathbf{s}}_k^m$ obtained by the sphere search. Clearly, $J_d(\tilde{\mathbf{s}}_k^m)$ is

$$J_d(\tilde{\mathbf{s}}_k^m) = \| \mathbf{y}_d - \mathbf{R}_{rd}\tilde{\mathbf{s}}_k^m \|^2. \tag{28}$$

Moreover, noting that $\hat{\mathbf{s}}_1^{k-1}$ is available, $J_u(\bar{\mathbf{s}}_1^m)$ becomes

$$J_u(\bar{\mathbf{s}}_1^m) = \|\mathbf{y}_u - \mathbf{R}_{lu}\hat{\mathbf{s}}_1^{k-1} - \mathbf{R}_{ru}\tilde{\mathbf{s}}_k^m\|^2. \tag{29}$$

Combining $J_d(\tilde{\mathbf{s}}_k^m)$ and $J_u(\bar{\mathbf{s}}_1^m)$, we obtain the modified hypersphere condition

$$J(\bar{\mathbf{s}}_1^m) = J_u(\bar{\mathbf{s}}_1^m) + J_d(\tilde{\mathbf{s}}_k^m) \leq d_0. \tag{30}$$

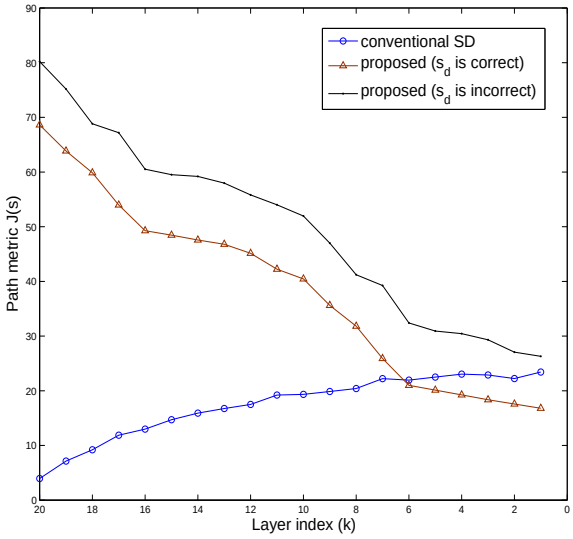



Fig. 4. Average of path metric $J(\bar{\mathbf{s}}_1^m)$ for the conventional SD and the proposed method.

An intuitive justification of the additional term $J_u(\bar{\mathbf{s}}_1^m)$ is as follows. When a path visited by the CLPS-NP is correct ($\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$) and $\hat{\mathbf{s}}_1^{k-1} \approx \mathbf{s}_1^{k-1}$ then $J_u(\bar{\mathbf{s}}_1^m) = \|\mathbf{R}_{lu}(\mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1}) + \mathbf{n}_u\|^2$ is not much different from $\|\mathbf{n}_u\|^2$. Whereas if $\hat{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$ then $J_u(\bar{\mathbf{s}}_1^m) = \|\mathbf{R}_{lu}(\mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1}) + \mathbf{R}_{ru}(\mathbf{s}_k^m - \tilde{\mathbf{s}}_k^m) + \mathbf{n}_u\|^2$ so that $J_u(\bar{\mathbf{s}}_1^m)$ is much larger than $\|\mathbf{n}_u\|^2$. In this scenario, $J_u(\bar{\mathbf{s}}_1^m) + J_d(\tilde{\mathbf{s}}_k^m)$ is highly likely to be larger than the hypersphere radius $d_0$ so that all children of this incorrect path might be pruned from the search tree. Our claim for competitive optimality of the proposed scheme is validated in the following theorem.

*Theorem 1:* Let $EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ be the expected value of $J(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ given $\tilde{\mathbf{s}}_k^m$ for the CLPS-NP. Then $EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ for $\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$ is smaller than $EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ for $\tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$. Specifically, the expected path metric $EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ for two scenarios are

$$EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) = \begin{cases} E(\epsilon^T \mathbf{T}_{1:k-1}^T \mathbf{T}_{1:k-1}\epsilon) + m\sigma^2 & \\ \qquad\qquad \text{if } \tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m & \\ E(\epsilon^T \mathbf{T}_{1:k-1}^T \mathbf{T}_{1:k-1}\epsilon) + E(\delta^T \mathbf{T}_{k:m}^T \mathbf{T}_{k:m}\delta) & \\ +m\sigma^2 \qquad\qquad \text{if } \tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m & \end{cases}$$

where $\epsilon = \mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1}$, $\delta = \mathbf{s}_k^m - \tilde{\mathbf{s}}_k^m$, and $\mathbf{T}_{a:b}$ is the submatrix generated from $a$-th to $b$-th columns of the channel matrix $\mathbf{T}$.

*Proof:* See Appendix B. ∎

*Corollary 2:* Let $EJ(\tilde{\mathbf{s}}_k^m)$ be the expected value of $J(\tilde{\mathbf{s}}_k^m)$ for the conventional CLPS. Then $EJ(\tilde{\mathbf{s}}_k^m)$ for $\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$ is smaller than $EJ(\tilde{\mathbf{s}}_k^m)$ for $\tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$. To be specific, we have

$$EJ(\mathbf{s}_k^m, \tilde{\mathbf{s}}_k^m) = \begin{cases} (m-k+1)\sigma^2 & \text{if } \tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m \\ E(\delta^T \mathbf{R}_{rd}^T \mathbf{R}_{rd}\delta) + (m-k+1)\sigma^2 & \text{else} \end{cases}$$

Theorem 1 and Corollary 2 provide the following interesting observations.

- While the path metric of the CLPS-NP for the first case ($\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$) contains the estimation error and noise, that for the seconde case ($\tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$) includes additional positive term $E(\delta^T \mathbf{T}_{k:m}^T \mathbf{T}_{k:m}\delta)$ caused by detection error. To empirically verify the difference between path metrics for two scenarios, we evaluate $J(\bar{\mathbf{s}}_1^m)$ from the simulation (simulation setup is described in Section VI-A). In our simulation, we consider three cases including the path metric of the conventional SD ($J_d(\tilde{\mathbf{s}}_k^m)$ only) and $J(\bar{\mathbf{s}}_1^m) = J_u(\bar{\mathbf{s}}_1^m) + J_d(\tilde{\mathbf{s}}_k^m)$ of the proposed method for $\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$ and $\tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$, respectively. We see from Fig. 4 that the difference between path metrics of two scenarios caused by the detection error $E(\delta^T \mathbf{T}_{k:m}^T \mathbf{T}_{k:m}\delta)$ is considerable, which implies that most of incorrect path will be removed from the search.
- In contrast to the conventional SD algorithm, the path metric of the CLPS-NP is higher in early layers of the tree due to the inclusion of (estimation/detection) error terms. In fact, as shown in Fig. 4, $J(\bar{\mathbf{s}}_1^m)$ tends to decrease as the search proceeds to the bottom of the tree.[3] Note that since the conventional path metric does not consider the noncausal path metric, the path metric of early layer nodes is in general much smaller than that of bottom layer nodes (see Fig. 4). Since the path metric $J(\tilde{\mathbf{s}}_k^m)$ of early layers is typically smaller than the hypersphere radius ($d_0 = J(\tilde{\mathbf{s}}_1^m)$), regardless of whether the current path is true or not, we expect that lots of back-tracking occurs during the search. Whereas, noncausal path metric of the CLPS-NP compensates the gap between path metrics at top and bottom layers so that chances of back-tracking operations are much smaller than the conventional CLPS. We will say more on this in the next subsection.

We summarize the proposed CLPS-NP algorithm in Table II. Note that, when compared to the SD algorithm, the extra operation of the proposed method is only on $J(\bar{\mathbf{s}}_1^m)$ computation in step 5.

### C. Search Complexity of CLPS-NP

The fact that the path metric decreases as the search proceeds to the bottom of the tree is very helpful in improving the pruning capability of the CLPS-NP. Following theorem formalizes this argument.

*Theorem 3:* Suppose the path metric $J(\bar{\mathbf{s}}_1^m) = J(\tilde{\mathbf{s}}_k^m, \hat{\mathbf{s}}_1^{k-1})$ for each layer is ordered in descending order. That is, $J(\tilde{\mathbf{s}}_m^m, \hat{\mathbf{s}}_1^{m-1}) \geq J(\tilde{\mathbf{s}}_{m-1}^m, \hat{\mathbf{s}}_1^{m-2}) \geq \cdots \geq J(\tilde{\mathbf{s}}_1^m)$. Then the first lattice point found (so called *Babai point* [24]) becomes the output of the SD algorithm.

*Proof:* Once the Babai point $\tilde{\mathbf{s}}_1^m(b)$ is found in the sphere search, then the hypersphere radius is updated to $d_0 = J(\tilde{\mathbf{s}}_1^m(b))$. Since the path metric in the search tree is sequenced in descending order, when the search is moved to the upper layer of the tree, $J(\tilde{\mathbf{s}}_2^m, \hat{s}_1) \geq d_0 = J(\tilde{\mathbf{s}}_1^m(b))$ and hence all subtrees of the node $\tilde{\mathbf{s}}_2^m$ are pruned from the tree (see Fig. 5). Since the condition $J(\tilde{\mathbf{s}}_k^m, \hat{\mathbf{s}}_1^{k-1}) \geq d_0$ holds true for the rest of layers ($1 < k \leq m$), no node satisfies

[3]This is in particular true for $\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$.

TABLE II
CLPS-NP ALGORITHM

| | |
|---|---|
| Input: | $d_0$, $\mathbf{y}$, $\mathbf{R}$, and $\hat{\mathbf{s}}$ |
| Output: | $\tilde{\mathbf{s}}$ |
| Variable: | $k$ denotes the $(m-k+1)$-th layer being examined |
| | $i_k$ denotes the lattice point index sorted by the SE enumeration in the $(m-k+1)$-th layer. |
| step **1** : | Set $k=m$. |
| step **2** : | Find the sorted list $[s_{k,\min}, \cdots, s_{k,\max}]$ satisfying $J_d(\tilde{\mathbf{s}}_k^m) \le d_0$. |
| | Set $N_s = s_{k,\max} - s_{k,\min} + 1$ and $i_k = 0$. |
| step **3** : | $i_k = i_k + 1$. |
| | If $i_k > N_s$, go to step **4**. |
| | Else, go to step **5**. |
| step **4** : | $k = k+1$. |
| | If $k = m+1$, output the latest s and terminate. |
| | Else, go to step **3**. |
| step **5** : | Compute $J_u(\hat{s}_1^{k-1}, \tilde{s}_k^m)$ and then update the path metric $J(\bar{\mathbf{s}}_1^m) = J_d(\tilde{s}_k^m) + J_u(\hat{s}_1^{k-1}, \tilde{s}_k^m)$. |
| | If $k=1$, go to step **6**. |
| | Else, $k = k-1$, go to step **2**. |
| step **6** : | If $J(\bar{\mathbf{s}}_1^m) < d_0$, save $\mathbf{s} = \bar{\mathbf{s}}_1^m$ and update $d_0 = J(\bar{\mathbf{s}}_1^m)$. |
| | Go to step **3**. |

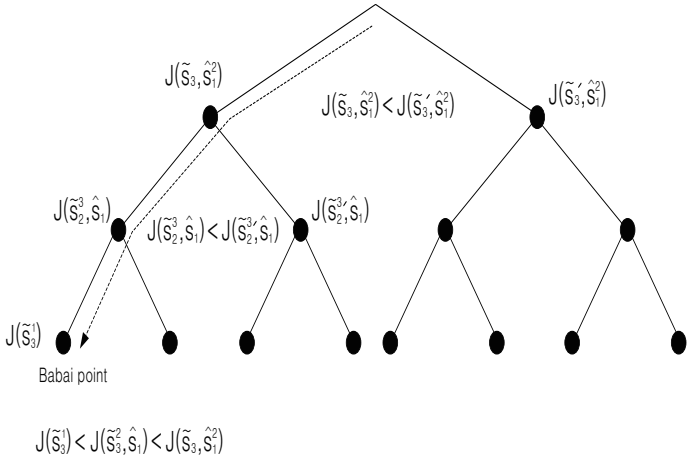



Fig. 5. Illustration of the CLPS-NP operation. Since the path metric tends to decrease as the search proceeds to the bottom of the tree and the path with a small path metric is searched first in each layer, it is highly likely that the search is finished at the Babai point.

the hypersphere condition until the search proceeds to the top layer of the search tree. Hence, the Babai point $\tilde{\mathbf{s}}_1^m(b)$ becomes the output of the SD algorithm. ■

*Corollary 4:* Suppose the path metric $J(\bar{\mathbf{s}}_1^m) = J(\tilde{\mathbf{s}}_k^m, \hat{\mathbf{s}}_1^{k-1})$ for each layer is ordered in descending order. Then the search complexity (in terms of number of nodes visited) becomes $2m-1$ ($m$ steps to arrive at the Babai point and additional $m-1$ steps for the termination).

Since the path metric of the CLPS-NP is not strictly ordered and ordered only on the expected sense, Corollary 4 cannot be directly applied to the CLPS-NP. Nevertheless, the property that the path metric in the lower layer of the tree tends to be smaller than that at the upper layer improves the pruning capability substantially. Indeed, as will be supported in our simulations in Section V, most of the outputs of the CLPS-NP equals to the Babai point, which demonstrates the effectiveness of the proposed noncausal path metric.

## V. COMPLEXITY ANALYSIS

In this section, we discuss the complexity of the proposed algorithm including the iterative LMMSE estimation and the CLPS-NP. In our analysis, we measure the complexity of the proposed algorithm by the number of floating point operations (flops). While the major operation of the CLPS in (6) and (19) is the sphere decoding and that of the LMMSE receiver is FIR filtering for equalization, operations of the proposed method include both. Overall, they consist of 1) equalization and symbol generation, 2) retransmission for interference cancellation, and 3) CLPS-NP.

In the first step, received signal vector is fed to the equalizer. If we assume the number of taps of the equalizer filter to be $N_{eq}$[4], then $N_{eq}N_{SF}$ flops are required. Equalized chips are converted to the symbol estimate via descrambling and inverse Hadamard transformation. Operation of the descrambling is trivial since the multiplication with $\pm 1$ can be implemented with simple sign change and $N_{SF}^2$ flops are required for the Hadamard transformation. The output of the Hadamard transform is converted into the soft symbol through the soft slicing. Since $3|S|+5$ flops are required for each code channel, assuming $N_u$ to be the number of active code channels, $(3|S|+5)N_u$ flops are required for the soft slicing. Retransmission for the interference cancellation is performed in the second step. Operations in this step consist of the scrambling, inverse Hadamard transform ($N_{SF}^2$ flops), convolution with the channel impulse response ($N_h N_{SF}$ flops), and the interference cancellation ($N_{SF}+N_h-1$ flops). Finally, operations in the CLPS-NP consist of the QR-decomposition of $\tilde{\mathbf{T}}$-matrix, $\mathbf{r}$-vector generation, precomputation of the path metric using the LMMSE estimate, and the modified sphere decoding. Since the dimension of $\tilde{\mathbf{T}}$-matrix is $2(N_{SF}+N_h-1) \times 2N_cN_u$, $12N_c^2N_u^2(2N_{SF}+2N_h-2-\frac{2}{3}N_cN_u)$ flops are required for the QR-decomposition.[5] The matrix-vector multiplication for

[4]In order to ensure reliable performance, $N_{eq}$ should be sufficiently larger than the length of the channel impulse response [25].

[5]For the QR-decomposition of the $m \times n$ matrix, $3n^2(m-n/3)$ flops are required [26].

TABLE III
COMPLEXITY OF THE PROPOSED ALGORITHM

| Operations | Detailed operations (flops) |
|---|---|
| 1) Equalization and symbol generation (per iteration) | - Equalization filtering: $2N_{eq}N_{SF}$<br>- Inverse Hadamard transform: $N_{SF}^2$<br>- Gain estimation: $5N_{SF}$<br>- Soft symbol processing: $(3\|S\|+5)N_u$ |
| 2) Retransmission for interference cancellation (per iteration) | - Inverse Hadamard transform: $N_{SF}^2$<br>- Channel convolution filtering: $N_hN_{SF}$<br>- Interference cancellation : $N_{SF}+N_h-1$ |
| 3) CLPS-NP | - QR-decomposition: $12N_c^2N_u^2(2N_{SF}+2N_h-2-\frac{2}{3}N_cN_u)$<br>- $\mathbf{y}$ vector gen.: $8(N_{SF}+N_h-1)^2-2(N_{SF}+N_h-1)$<br>- Precomputation of the path metric using the LMMSE estimate: $4N_c^2N_u^2+2N_cN_u$<br>- Modified sphere decoding : 4K at layer $2N_cN_u-K+1$ |

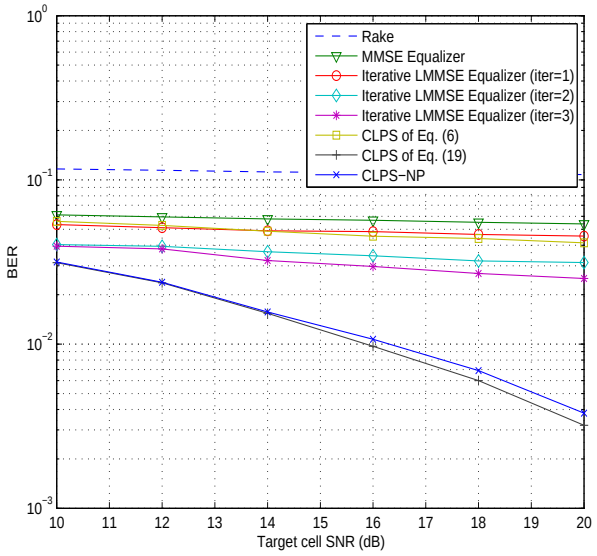
generating $\mathbf{y}$ vector ($\mathbf{y}=\mathbf{Q}^T\mathbf{r}'$) requires $4N_cN_u(N_{SF}+N_h-1)$ flops. Preprocessing for computing $J(\hat{\mathbf{s}}_1^m)$ requires $4N_c^2N_u^2+2N_cN_u$ flops. Finally, in the modified sphere decoding of the CLPS, replacement of the estimated component by the detected component should be done. Number of operations is largest in the bottom layer and it gradually decreases as the search goes on. Specifically, $4K$ flops are required in the layer $2N_cN_u-K+1$.

In contrast to the CLPS-NP, the sphere decoding is a major task for the CLPS algorithm in (6) and (19). Number of computations required for the QR-decomposition and $\mathbf{y}$-vector generation is the same as that of the CLPS-NP. In the sphere decoding stage, $2(2N_cN_u-K+1)-1$ operations are required in the layer $2N_cN_u-K+1$. It is interesting to note that while the number of operations for the sphere decoding increases as the search proceeds into the upper layer (i.e., bottom of the search tree), that for the CLPS-NP decreases since the number of elements to be replaced decreases gradually. Finally, number of the operations for the iterative LMMSE equalizer becomes simply the sum of the first and second steps of the proposed method (see Table II).

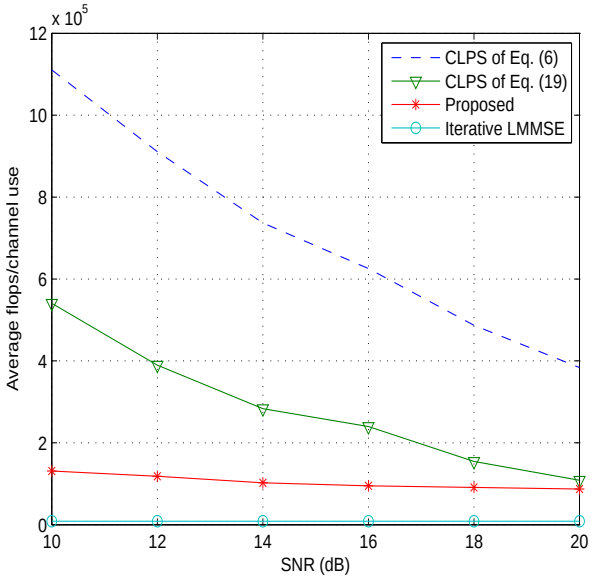
## VI. Simulations Results and Discussion

### A. Simulations setup

In our simulation, we check the performance of the proposed method in the downlink of HSPA system [1] where the received signal of the mobile is coming from two asynchronously operated cells (one chip delay between target and interfering cells) in a multipath environment. The simulated propagation channel is generated based on the ITU model [27]. Both desired users in a target cell as well as interferers in a secondary cell employ five physical data channels (HS-PDSCH) with QPSK modulation. Two typical scenarios of cellular system, i.e., cell-edge and cell-site scenarios, are simulated. To quantify the power ratio between the target cell $P_t$ and the interfering cell $P_i$ for each scenario, we define the signal to interference ratio (SIR$=\frac{P_t}{P_i}$). As a measure for performance and complexity, we employ the bit error rate (BER) of data channels in a target cell and the average number of nodes visited of the CLPS algorithms. For each point in the BER curve, we simulate at least 100 transmission time intervals (TTI).

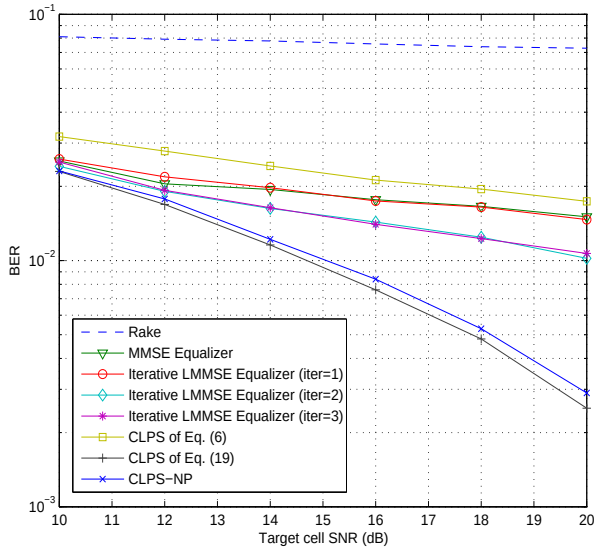



(a)

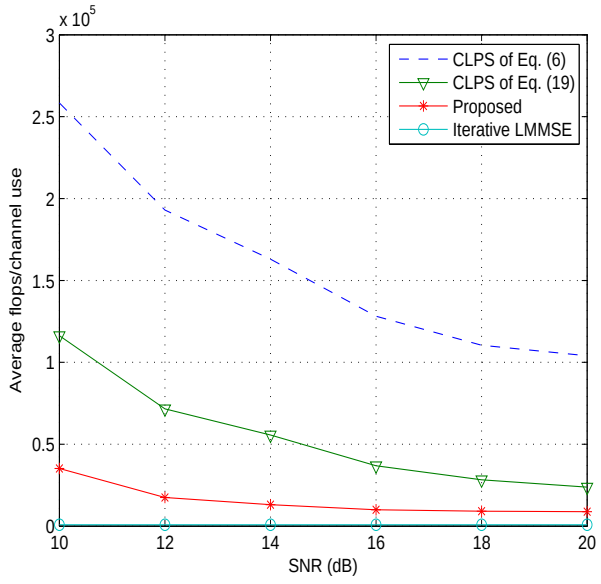



(b)

Fig. 6. Performance of proposed approaches at $SIR=0$ dB: (a) BER and (b) required flops.




(a)




(b)

Fig. 7. Performance of proposed approaches at $SIR = 6$ dB: (a) BER and (b) required flops.

### B. Simulations results

We first consider the scenario with 0 dB SIR ($P_t = P_i$). This scenario represents the situation where a mobile is located close to the cell edge so that the mobile receives same amount of power from the target and interfering cells. In order to obtain a comprehensive view, we test conventional methods including the RAKE and LMMSE equalizer as well as the approaches presented in the previous sections (the iterative LMMSE equalizer, the CLPS of (6) and (19), and the CLPS-NP). Note that we do not consider the CLPS of (3) and (5) simply because it takes extremely long simulation time. Note also that the CLPS of (19) provides the performance bound that can be achieved by the CLPS-NP. The initial radius of the CLPS methods is set to $J(\hat{\mathbf{s}})$.

Due to the existence of strong interference, we expect that the proposed CLPS algorithms detecting users of target and interfering cells outperform the conventional approaches designed to detect/estimate users of target cell only. In fact, as shown in Fig. 6(a), performance of the iterative LMMSE equalizer improves as the number of iteration increases. While the gain of the iterative LMMSE equalizer is limited due to the considerable amount of residual interference, the gain of the CLPS approach of (19) and the proposed CLPS-NP increases with SNR due to the control of multiuser interference. For example, proposed method provides more than 8 dB gain over the iterative LMMSE equalizer at 5% BER and the gain increases with target cell SNR. On the contrary, we observe that the CLPS of (6) does not perform well since they are operated in a poor SINR condition.

Fig. 6(b) shows the average flops for CLPS approaches and the iterative LMMSE estimation. Since the CLPS of (6) is operated in a poor SINR condition, it is no wonder that this scheme shows extremely large complexity. Although the CLPS of (19) is better than the CLPS of (6), computational cost of this approach is still quite expensive, in particular for low and mid SNR regime, so that the implementation of this option seems to be unrealistic. Complexity reduction of the proposed CLPS-NP, when compared to these approaches, is remarkable due to the fact most of runs are finished at Babai point. Clearly, we can conclude that strengthening of the hypersphere condition via the noncausal path metric is very effective in pruning of unpromising path.

We next consider the scenario with 6 dB SIR ($P_i = 0.25P_t$). This case represents a situation where a mobile is located near the basestation so that the target cell power is dominant over the interfering cell power. As shown in Fig. 7(a), due to the reduction of the interference power, we see that there is a slight performance improvement in receiver algorithms focusing only on the target cell. We also observe that performance gain of the CLPS of (19) and the proposed CLPS-NP is still maintained. Although there is a slight performance gap between two methods (around 0.3 dB at BER $= 10^{-2}$), as shown in Fig. 7(b), complexity benefit of the CLPS-NP over the CLPS of (19) is significant.

## VII. Concluding Remarks

In this work, we investigated a low-complexity multiuser detection algorithm for the downlink of HSPA system. A key feature of our method lies in the combination of the estimation technique (iterative LMMSE estimation) and the nonlinear detection technique (CLPS-NP). By deliberately exploiting estimated symbols in the ISI cancellation, we could limit the search dimension of the CLPS. Further, we could tighten the hypersphere condition of the sphere decoding by incorporating the noncausal path metric from the iterative LMMSE estimation. Indeed, due to the strengthening of the hypersphere condition, most of solutions in the CLPS-NP are found in the first candidate often referred to as Babai point. In doing so, we could achieve an excellent tradeoff between performance

and computational complexity. Although our discussions in this paper focus on the multiuser detection in HSPA based UMTS system, our method can be readily extended to various applications such as single user detection in channels under frequency selective fading and symbol detection of the MIMO-OFDM systems.

## APPENDIX A
## PROOF OF (9)

By definition,

$$E[|\mathbf{x}_j|^2] = E[\mathbf{x}_j\mathbf{x}_j^H] = E[\mathbf{C}_j\mathbf{W}\mathbf{G}_j\mathbf{s}_j\mathbf{s}_j^H\mathbf{G}_j\mathbf{W}^T\mathbf{C}_j^H] \quad (31)$$

Let $\mathbf{A} = \mathbf{W}\mathbf{G}_j\mathbf{s}_j\mathbf{s}_j^H\mathbf{G}_j\mathbf{W}^T$ and $\mathbf{B} = \mathbf{G}_j\mathbf{s}_j\mathbf{s}_j^H\mathbf{G}_j$. Since $\mathbf{C}_j$ is diagonal matrix, we have

$$\begin{aligned}\mathbf{x}_j\mathbf{x}_j^H &= \mathbf{C}_j\mathbf{A}\mathbf{C}_j^H \\ &= \begin{bmatrix} a_{1,1} & c_1c_2^*a_{1,2} & \cdots & c_1c_n^*a_{2,n} \\ c_2c_1^*a_{2,1} & a_{2,2} & \cdots & c_1c_n^*a_{2,n} \\ & & \ddots & \\ c_nc_1^*a_{n,1} & c_nc_2^*a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \quad (32)\end{aligned}$$

Noting that $\mathbf{A} = \mathbf{W}\mathbf{B}\mathbf{W}^T$ and $\mathbf{W}$ is the Hadamard matrix, one can easily show that $a_{m,n} = \sum_u\sum_v b_{u,v}w_{m,u}w_{v,n}$. In addition, from the definition of $\mathbf{B}$, $b_{u,v} = g_ug_vs_us_v^*$ and thus

$$a_{m,n} = \sum_u\sum_v g_ug_vs_us_v^*w_{m,u}w_{v,n}. \quad (33)$$

Using (32) and (33), we have

$$\begin{aligned}(\mathbf{x}_j\mathbf{x}_j^H)_{m,n} &= c_mc_n^*\sum_u g_ug_vs_us_v^*w_{m,u}w_{v,n}c_mc_n^* \\ &= c_mc_n^*\sum_u g_u^2|s_u|^2w_{m,u}w_{u,n} \\ &\quad + c_mc_n^*\sum_{u\neq v} g_ug_vs_us_v^*w_{m,u}w_{v,n}. \quad (34)\end{aligned}$$

After taking expectation,

$$\begin{aligned}(E[\mathbf{x}_j\mathbf{x}_j^H])_{m,n} &= E[c_mc_n^*]\sum_u g_u^2E|s_u|^2w_{m,u}w_{u,n} \\ &= \begin{cases} \sum_u g_u^2 = tr(\mathbf{G}^2) & m = n \\ 0 & m \neq n \end{cases} \quad (35)\end{aligned}$$

where $E[|s_u|^2] = 1$, $E[s_us_v^*] = 0$ for $u \neq v$, and $E[c_mc_n^*] = 0$ for $m \neq n$.

## APPENDIX B
## PROOF OF THEOREM 1

The expected path metric $EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m)$ becomes

$$\begin{aligned}EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) &= EJ_u(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) + EJ_d(\tilde{\mathbf{s}}_k^m) \\ &= E\|\mathbf{y}_u - \mathbf{R}_{lu}\hat{\mathbf{s}}_1^{k-1} - \mathbf{R}_{ru}\tilde{\mathbf{s}}_k^m\|^2 \\ &\quad + E\|\mathbf{y}_d - \mathbf{R}_{rd}\tilde{\mathbf{s}}_k^m\|^2. \quad (36)\end{aligned}$$

First, when $\tilde{\mathbf{s}}_k^m = \mathbf{s}_k^m$, it is clear that

$$\begin{aligned}EJ_d(\tilde{\mathbf{s}}_k^m) &= E\|\mathbf{y}_d - \mathbf{R}_{rd}\tilde{\mathbf{s}}_k^m\|^2 \\ &= E\|\mathbf{n}_d\|^2 = (m-k+1)\sigma^2. \quad (37)\end{aligned}$$

In addition, we have

$$\begin{aligned}EJ_u(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) &= E\|\mathbf{y}_u - \mathbf{R}_{lu}\hat{\mathbf{s}}_1^{k-1} - \mathbf{R}_{ru}\tilde{\mathbf{s}}_k^m\|^2 \\ &= E\|\mathbf{R}_{lu}(\mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1}) + \mathbf{n}_u\|^2 \\ &= E(\|\mathbf{R}_{lu}(\mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1})\|^2 \\ &\quad + 2(\mathbf{R}_{lu}(\mathbf{s}_1^{k-1} - \hat{\mathbf{s}}_1^{k-1}))^T\mathbf{n}_u + \|\mathbf{n}_u\|^2). \quad (38)\end{aligned}$$

Denoting the MMSE estimation error as $\epsilon = \mathbf{s}_1^{k-1} - \tilde{\mathbf{s}}_u$, $EJ_u(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) = E(\|\mathbf{R}_{lu}\epsilon\|^2 + \|\mathbf{n}_u\|^2) = E\|\mathbf{R}_{lu}\epsilon\|^2 + (k-1)\sigma^2$. From (36) and (38), we have

$$\begin{aligned}EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) &= E\|\mathbf{R}_{lu}\epsilon\|^2 + m\sigma^2 \\ &= E\epsilon^T\mathbf{R}_{lu}^T\mathbf{R}_{lu}\epsilon + m\sigma^2 \\ &= E\epsilon^T(\mathbf{Q}\mathbf{R}_{lu})^T(\mathbf{Q}\mathbf{R}_{lu})\epsilon + m\sigma^2 \\ &= E(\epsilon^T\mathbf{T}_{1:k-1}^T\mathbf{T}_{1:k-1}\epsilon) + m\sigma^2. \quad (39)\end{aligned}$$

Next, consider the case $\tilde{\mathbf{s}}_k^m \neq \mathbf{s}_k^m$. First, we have

$$\begin{aligned}EJ_d(\tilde{\mathbf{s}}_k^m) &= E\|\mathbf{y} - \mathbf{R}_{rd}\tilde{\mathbf{s}}_k^m\|^2 \\ &= E\|\mathbf{R}_{rd}(\mathbf{s}_k^m - \tilde{\mathbf{s}}_k^m) + \mathbf{n}_d\|^2. \quad (40)\end{aligned}$$

Denoting an error of the sphere detection as $\delta = \mathbf{s}_k^m - \tilde{\mathbf{s}}_k^m$, (40) becomes

$$\begin{aligned}EJ_d(\tilde{\mathbf{s}}_k^m) &= E\|\mathbf{R}_{rd}\delta + \mathbf{n}_d\|^2 \\ &= E\|\mathbf{R}_{rd}\delta\|^2 + (m-k+1)\sigma^2. \quad (41)\end{aligned}$$

In addition, the first term in the right-hand of (36) is

$$\begin{aligned}EJ_u(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) &= E\|\mathbf{y}_u - \mathbf{R}_{lu}\hat{\mathbf{s}}_1^{k-1} - \mathbf{R}_{ru}\tilde{\mathbf{s}}_k^m\|^2 \\ &= E\|\mathbf{R}_{lu}\epsilon + \mathbf{R}_{ru}\delta + \mathbf{n}_u\|^2 \\ &= E\|\mathbf{R}_{lu}\epsilon\|^2 + E\|\mathbf{R}_{ru}\delta\|^2 + (k-1)\sigma^2. \quad (42)\end{aligned}$$

Combining (41) and (42), we have

$$\begin{aligned}EJ(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) &= EJ_u(\hat{\mathbf{s}}_1^{k-1}, \tilde{\mathbf{s}}_k^m) + EJ_d(\tilde{\mathbf{s}}_k^m) \\ &= E\|\mathbf{R}_{lu}\epsilon\|^2 + E\|\mathbf{R}_{ru}\delta\|^2 + E\|\mathbf{R}_{rd}\delta\|^2 + \frac{m}{2} \\ &= E\|\mathbf{R}_{lu}\epsilon\|^2 + E\|\mathbf{R}_r\delta\|^2 + m\sigma^2 \\ &= E(\epsilon^T\mathbf{T}_{1:k-1}^T\mathbf{T}_{1:k-1}\epsilon) \\ &\quad + E(\delta^T\mathbf{T}_{k:m}^T\mathbf{T}_{k:m}\delta) + m\sigma^2. \quad (43)\end{aligned}$$

## REFERENCES


[1] "3rd Generation Partnership Project: Technical specification group radio access network; physical layer (Release 7)," Tech. Spec. 25.201 V7.2.0, 2007. (see http://www.3gpp.org/ftp/Specs/archive/25_series/25.201/25201-720.zip).

[2] S. Verdu, *Multiuser detection*, Cambridge Univ. Press, 1998.

[3] S. Moshavi, "Multi-user detection for DS-CDMA communications," *IEEE Commun. Magazine*, pp. 124-136, Oct. 1996.

[4] A. Dual-Hallen, J. Holtzman and Z. Zvonar, "Multiuser detection for CDMA systems," *IEEE Personal Commun. Magazine*, pp. 46-58, April 1995.

[5] R. Lupas and S. Verdu, "Linear multiuser detectors for synchronous code-division multiple-access channels", *IEEE Trans. Inform. Theory*, vol. 35, pp. 123-136, 1989.

[6] T. S. Rappaport, *Wireless Communications*, Prentice Hall, 1996.

[7] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463-471, 1985.

[8] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, pp. 2806-2818, Aug. 2005.

[9] J. Jalden and B. Ottersten,"On the complexity of sphere decoding in digital communication," *IEEE Trans. Signal Process.*, vol. 53, pp. 1474-1484, April 2005.

[10] D. Tse and O. Zeitouni, "Linear multiuser receiver in random environments," *IEEE Trans. Inf. Theory*, vol. 46, pp. 171-188, Jan. 2000.

[11] N. Prasad and M. K. Varanasi, "Analysis of decision feedback detection for MIMO rayleigh-fading channels and the optimization of power and rate allocations," *IEEE Trans. Inf. Theory* vol. 50, pp. 1009-1025, June 2004.

[12] M. Stojnic, H. Vikalo, and B. Hassibi, "Speeding up the sphere decoder with $H^{\infty}$ and SDP inspired lower bounds," *IEEE Trans. Signal Process.*, vol. 56, pp. 712-726, Feb. 2008.

[13] B. Shim, J. Choi, and I. Kang, "Towards the performance of ML and the complexity of MMSE - A hybrid approach", Proc. of *Globecom conf.*, Dec. 2008, pp. 1-5.

[14] B. Shim and I. Kang, "Sphere decoding with a probabilistic tree pruning," *IEEE Trans. Signal Process.*, vol. 56, pp. 4867-4878, Oct. 2008.

[15] J. W. Choi, B. Shim, N. I. Cho, and A. C. Singer, "Low-complexity decoding via reduced dimension maximum-likelihood search," *IEEE Trans. Signal Process.*, vol. 58, pp 1780-1793, March 2010.

[16] E. Viterbo, E. Biglieri, "A universal lattice decoder," in *GRETSI 14-eme Colloque*, Jun-les-pins France, Sept. 1993.

[17] M. F. Madkour, S. C. Gupta, and Y. E. Wang, "Successive interference cancellation algorithms for downlink W-CDMA communications," *IEEE Trans. Commun.*, vol. 1, pp. 169-177, Jan. 2002.

[18] S. M. Kay, *Fundamentals of statistical signal processing*, Prentice Hall, 1999.

[19] J. G. Proakis, *Digital Communications*, 3rd ed. McGraw-Hill, 1995.

[20] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. Wiley, 2006.

[21] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, pp. 41-59, Oct. 1996.

[22] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," *ACM SIGSAM*, vol. 15, pp. 37-44, 1981.

[23] J. Clausen,"Branch and bound algorithms; principles and examples," Dept. Comput. Sci., Univ. Copenhagen, [Online]. Available:http://www.imm.dtu.dk/jha/.

[24] L. Babai, "On Lovasz' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, pp. 1-13, 1986.

[25] B. Shim and N. Shanbhag, "Complexity analysis of multicarrier and single-carrier systems for very high-speed digital subscriber line", *IEEE Trans. Signal Process.*, vol. 51, pp. 282-292, Jan. 2003.

[26] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1989

[27] "Guidelines for the evaluation of radio transmission technologies for IMT-2000," Recommendation ITU-R M.1225, 1997.