

A Comparative Study on Low-Cost Multiuser Detectors

Lihai Liu

Communication & Signaling Research & Design
Department, The Fourth Survey and Design Institute of
China Railway, Wuhan, China

Li Ping

Department of Electronic Engineering,
City University of Hong Kong
eelping@cityu.edu.hk

Abstract—In this paper, we compare different low-cost detection techniques for multiple access systems. The per-user complexities of these techniques are independent of the number of users. We demonstrate that the chip extrinsic information (CEI) method achieves the best performance, and the *a posteriori* probability information (APPI) and bit extrinsic information (BEI) methods provide different tradeoffs between performance, complexity and storage requirements. These low-cost multiuser detectors are applicable to a variety of systems, including interleave-division multiple-access (IDMA) and code-division multiple-access (CDMA) systems. Both analysis and simulation demonstrate that interleaving can greatly improve the effectiveness of the low-cost algorithms. In the presence of chip interleavers, the CEI method slightly outperforms the more complex minimum mean square error (MMSE) method.

Keywords—multi-user detection (MUD), iterative detection, factor graph, interleave-division multiple-access (IDMA), code-division multiple-access (CDMA).

I. INTRODUCTION

Iterative multiuser detection (MUD) is an effective technique for alleviating the interference problem in multiple access systems [1]–[7]. Complexity is, however, always a serious concern for MUD. The maximum *a posteriori* probability (MAP) based multiuser detector [2][3] has an exponential complexity of $O(2^K)$ (excluding the decoding for forward error control (FEC) codes), where K is the number of users. The minimum mean square error (MMSE) based detector [4] and the probability data association (PDA) detector [5] both have a polynomial complexity $O(K^3)$ due to the matrix operations involved. These complexities increase rapidly with K and are high for practical systems. Low-cost detectors with linear complexity ($O(K)$) have been studied recently [6]–[9], indicating that MUD has finally become an attractive option for practical systems.

This paper presents a comparative study on several low-cost iterative MUD methods. We adopt the treatment introduced in [11] where MUD is realized as a message passing process over factor graphs. This approach provides a useful framework for various implementation methods, including some well-known iterative code-division multiple-access (CDMA) multiuser detector structures [4]–[7][12] as special cases.

Our focus will be on message passing techniques that can avoid matrix operations. We compare several strategies that differ in the generation of feedback messages. We will refer to these methods according to the nature of the feedback information, i.e., chip extrinsic information (CEI), *a posteriori* probability information (APPI) and bit extrinsic information (BEI). The complexity involved is $O(K)$ for all of these options. We demonstrate that CEI achieves the best performance and the other two options, namely APPI and BEI, provide different tradeoffs between performance, complexity and storage requirements.

Low cost and high performance are two desired features for a detection scheme. In addition, it is also highly desirable that the performance of a detection scheme can be predicted

quickly and accurately. We show that CEI together with user-specific interleaving can meet this additional requirement very well, making it an efficient option for search-based (involving repeated analysis) system optimization. Performance close to the theoretical limit is demonstrated by numerical results.

II. SYSTEM MODEL AND DETECTION PRINCIPLE

A. System Model

We consider a general multiple access system with K users as shown in Fig. 1. At the transmitter side (Fig. 1), the information bits d_k for user- k are first coded by an FEC encoder (ENC_k), interleaved by a bit-level interleaver Π_k , spread using a length- S spreading sequence in SP_k and then interleaved by a chip-level interleaver π_k before transmission. The resulting signal streams are respectively denoted by $\{u_k(i)\}$, $\{v_k(i)\}$, $\{w_k(j)\}$ and $\{x_k(j)\}$ in Fig. 1. For convenience, we call $\{u_k(i)\}$ and $\{v_k(i)\}$ “bits”, and $\{w_k(j)\}$ and $\{x_k(i)\}$ “chips”, following the CDMA convention. We assume that these sequences are all in BPSK format, taking values of $\{+1$ or $-1\}$. Denoting by $\{s_k(j), j = 1, \dots, S\}$ the spreading sequence for $v_k(1)$, we have

$$w_k(j) = s_k(j)v_k(1), \quad j = 1, \dots, S. \quad (1)$$

Similar relationships apply to other $v_k(i)$ and $w_k(j)$.

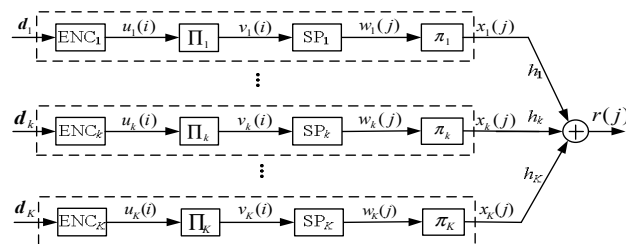


Fig. 1. The transmitter model.

At the receiver side, the signal from all of the K users (after matched filtering and sampling) can be written as

$$r(j) = \sum_{k=1}^K h_k x_k(j) + n(j), \quad j = 1, 2, \dots, J \quad (2)$$

where h_k represents the joint effect of power control at the transmitter side and channel gain for user- k , J is the frame length, and $\{n(j)\}$ are samples of a zero-mean additive white Gaussian noise (AWGN) with variance $\sigma^2 = N_0/2$. For simplicity, we assume that $\{h_k\}$ are real but the results can be easily extended to complex channels [9].

The system in Fig. 1 represents a conventional CDMA system [4][6] when $\{\pi_k\}$ are trivial, i.e., $w_k(j) = x_k(j)$, $\forall k, j$. It represents a chip interleaved CDMA system [8][9] when $\{\pi_k\}$ are non-trivial. In particular, it represents the so-called interleave-division multiple-access (IDMA) system [9] when (i) $\{\Pi_k\}$ are trivial, (ii) $\{\text{SP}_k\}$ are realized by repetition codes and (iii) $\{\pi_k\}$ are user-specific, hence users are separated solely by the interleavers. In this paper, we assume that the interleavers for IDMA are independently and randomly generated.

This work was fully supported by a grant from the Research Grant Council of the Hong Kong SAR, China [Project No. CityU 1164/03E].

B. Factor-Graph Representations

Fig. 2 shows a factor graph [1] for the system in Fig. 1 with the number of users $K = 2$ and spreading length $S = 3$. Chip-level interleaving is assumed in Fig. 2. IDMA is its special case when spreading constraints are realized by repetition codes for both users. (Then the two users are only distinguished by the edge connections between spreading constraint nodes and chip nodes.) Fig. 2 is similar to the factor graph for a low-density parity-check (LDPC) code, except that more types of constraints are involved here: the coding constraints are defined by the FEC codes used and the spreading and channel constraints by (1) and (2) respectively.

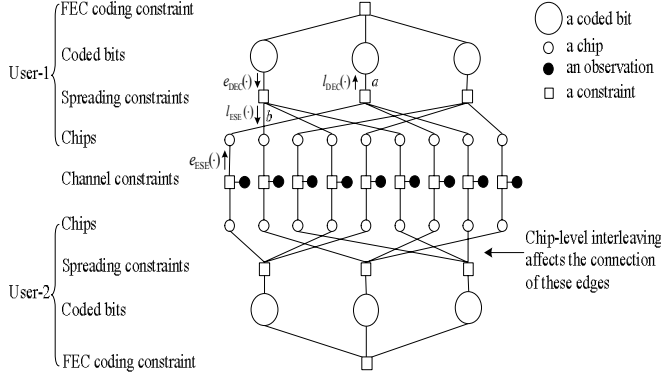


Fig. 2. Factor-graph representation for the system in Fig. 1. $K = 2$ and $S = 3$.

C. The Message Passing Detection Process

The factor graph in Fig. 2 can be solved using a message passing process. The operations at the variable nodes are trivial since the degrees of the variable nodes are all two. The messages go straight through these nodes. The operations at the FEC coding and spreading constraint nodes can be implemented by APP decoding in a straightforward way. (Besides the sign change specified by $\{(s_k(j))\}$, the spreading operation defined in (1) can be seen as a special case of repetition coding.)

We now focus on the channel constraint nodes. The key problem here is to resolve the linear superposition relationship (2) related to these nodes. For this purpose, we rewrite (2) as

$$r(j) = h_k x_k(j) + \zeta_k(j) \quad (3)$$

where $\zeta_k(j) = \sum_{k' \neq k} h_{k'} x_{k'}(j) + n(j)$. We approximate $\zeta_k(j)$ by a Gaussian random variable with mean $E(\zeta_k(j))$ and variance $\text{Var}(\zeta_k(j))$ respectively. Using this assumption, the message from a channel constraint node towards a chip node is the extrinsic log-likelihood ratio (LLR) regarding $x_k(j)$ computed based on (3) as follows,

$$e_{\text{ESE}}(x_k(j)) = \frac{2h_k}{\text{Var}(\zeta_k(j))} (r(j) - E(\zeta_k(j))). \quad (4)$$

The mean and variance involved in (4) can be estimated as follows:

$$E(x_k(j)) = \tanh(I_{\text{ESE}}(x_k(j))/2), \quad \forall k, j \quad (5a)$$

$$\text{Var}(x_k(j)) = 1 - (E(x_k(j)))^2, \quad \forall k, j \quad (5b)$$

$$E(\zeta_k(j)) = \sum_{k'=1}^K h_{k'} E(x_{k'}(j)) - h_k E(x_k(j)), \quad \forall k, j \quad (5c)$$

$$\text{Var}(\zeta_k(j)) = \sum_{k'=1}^K |h_{k'}|^2 \text{Var}(x_{k'}(j)) + \sigma^2 - |h_k|^2 \text{Var}(x_k(j)), \quad \forall k, j. \quad (5d)$$

Here $I_{\text{ESE}}(x_k(j))$, $\forall k, j$, are initialized to zeros and they are updated using the feedback LLRs from the spreading constraint nodes. (See Fig. 2 and (iv) below.)

The overall message passing process consists of the following four classes of operations over three types of constraint nodes. (**Note:** A spread constraint node involves two different classes of operations generating messages towards, respectively, a coded bit node and chip nodes.)

- (i) **Elementary signal estimation (ESE) at channel constraint nodes:** Generate $\{e_{\text{ESE}}(x_k(j))\}$ using (4) based on the feedback messages $\{I_{\text{ESE}}(x_k(j))\}$. (See Fig. 2.)
- (ii) **APP de-spreading at spreading constraint nodes:** Compute the bit-level estimate as

$$I_{\text{DEC}}(v_k(i)) = \sum_{j=(i-1)S+1}^{iS} s_k(j) e_{\text{ESE}}(x_k(\pi_k(j))) \quad (6)$$

where $\pi_k(\cdot)$ denotes the interleaving operation for user- k , i.e., $x(\pi_k(j)) = w_k(j)$. In Fig. 2, the arrow on edge a indicates the direction of a message generated in (6).

- (iii) **APP decoding at coding constraint nodes:** Based on the inbound messages $\{I_{\text{DEC}}(v_k(i))\}$, perform the APP decoding for the FEC codes to generate the outbound messages $\{e_{\text{DEC}}(v_k(i))\}$.
- (iv) **Feedback updating at channel constraint nodes:** Compute the chip-level soft information $\{I_{\text{ESE}}(x_k(j))\}$ to be fed back to the channel constraint nodes.

In the above, the ESE operation has been discussed earlier and the APP decoding and de-spreading are standard. The feedback updating in (iv) will be discussed in detail in Section III, where we show that different implementation strategies result in different trade-offs between performance and cost.

It can be verified that the operations in (i) – (iv) involve only single user operations (i.e., the operations at a node involve only the information for one user), except for the two summations defined in (5c) and (5d). Although (5c) and (5d) involve information from all users, their costs are also shared by these users. For this reason, it is easy to verify that the overall cost of the above detection procedure is $O(K)$.

D. CDMA and IDMA

As mentioned before, CDMA is a special case of the system in Fig. 1. This can be seen by the factor graph in Fig. 3 for a two-user CDMA setting where a frame of the chips related to a common coded bit are transmitted consecutively. We assume that the frames of the two users are fully synchronized, so that the graph consists of regular patterns.

The main difference between Figs. 2 and 3 lies in the edge connections between the chip nodes and channel constraint nodes. In Fig. 2, the edge connection is randomized by chip-level interleaving. Without such randomization, short cycles are formed in Fig. 3 (involving the chips connected to common coded bits). From our experience gained with LDPC codes [14], we know that short cycles are detrimental to system performance when iterative detection is involved. This is indeed true, as will be demonstrated later.

Due to the presence of short cycles, the messages from the chips to a common coded bit in Fig.3 are correlated. The simple summation defined in (6) is not an efficient way to handle this situation. Instead, the MMSE approach defined in

[4] (or the PDA approach in [5]) offers a better solution to the problem, since it takes into consideration the correlation effect. In this case, the MMSE algorithm can be represented using a bit-level factor graph (as shown in [6]). However, the MMSE approach involves a $K \times K$ correlation matrix and the related complexity is quite high ($O(K^3)$).

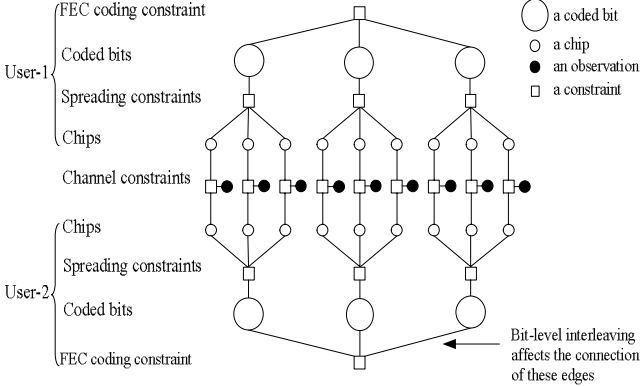


Fig. 3. The factor-graph representation for a synchronous CDMA system with $K = 2$ and $S = 3$.

The discussion below shows that, by introducing user-specific interleavers, low cost techniques with complexity $O(K)$ can perform as well as the MMSE one.

III. DIFFERENT INFORMATION COMBINING TECHNIQUES

We now return to the feedback updating (class (iv) operations) discussed in Section II.C. The issue involved is how to combine the messages at a spreading constraint node. We will refer to the methods considered according to the nature of the feedback information, i.e., chip extrinsic information (CEI), *a posteriori* probability information (APPI) and bit extrinsic information (BEI). We then relate and compare them with the existing detectors. CEI is closely related to the chip-by-chip detection algorithm discussed in [9] and APPI and BEI are related to the detection methods considered in [7][9].

A. Different Methods to Compute the LLR Values in (iv)

For simplicity, let us focus on the chips related to a particular coded bit $v_k(1)$ through a spreading constraint node. These chips are generated using the spreading operation defined in (1), which is essentially repetition coding plus sign change (by $\{s_k(j)\}$). Then, the feedback updating at a spreading constraint node in Fig. 2 can be implemented using an APP de-spreader, taking into consideration the feedback message from the FEC decoder. It generates messages $\{l_{ESE}(x_k(j))\}$ from $v_k(1)$ towards the channel constraint node. This is shown graphically in Fig. 2, where the arrow on edge b indicates the direction of $l_{ESE}(x_k(j))$. We now consider different ways to compute this message.

CEI: We compute $l_{ESE}(x_k(j))$ on edge b as the sum of all the inbound messages at the check node connected to b , except that on edge b itself.

$$l_{ESE}(x_k(\pi_k(j))) \leftarrow s_k(j)(e_{DEC}(v_k(1)) + l_{DEC}(v_k(1))) - e_{ESE}(x_k(\pi_k(j))) \quad (7)$$

where $\pi_k(\cdot)$ denotes the interleaving operation mapping $w_k(j)$ to

$$x(\pi_k(j)), \text{ and } l_{DEC}(v_k(1)) = \sum_{j=1}^S s_k(j) e_{ESE}(x_k(\pi_k(j))) \text{ (see (6)).}$$

Note that $l_{ESE}(x_k(j))$ and $e_{ESE}(x_k(j))$ are the messages on the same edge but in opposite directions. The subtraction in (7) ensures that $l_{ESE}(x_k(j))$ is “extrinsic”, since $e_{ESE}(x_k(j))$ is excluded from $l_{ESE}(x_k(j))$.

APPI: We can approximate (7) by

$$l_{ESE}(x_k(\pi_k(j))) \leftarrow s_k(j)(e_{DEC}(v_k(1)) + l_{DEC}(v_k(1))). \quad (8)$$

Without the subtraction in (7), $l_{ESE}(x_k(j))$ now contains $e_{ESE}(x_k(j))$ and is no longer “extrinsic”. Consequently, this method may result in performance loss.

BEI: We can also approximate (7) by

$$l_{ESE}(x_k(\pi_k(j))) \leftarrow s_k(j) e_{DEC}(v_k(1)). \quad (9)$$

Here $l_{ESE}(x_k(j))$ consists of the message from $v_k(1)$ only. This method is suboptimal, since it does not exploit the information from other chip nodes.

B. Comparison on Storage Requirement

CEI requires relatively high memory usage to store the chip-level values. In particular, the messages $\{e_{ESE}(x_k(j))\}$ from chip nodes towards a coded bit node must be stored after soft de-spreading (see Section II.C) since they are used in feedback updating defined in (7) later. For both APPI and BEI methods, these messages can be discarded after soft de-spreading since they are not used again. (Note that (8) and (9) do not involve chip-level messages.) This greatly reduces the storage usage.

C. Comparison on Computational Cost

Relatively speaking, APPI and BEI have lower computational cost. This can be seen from (8) and (9) where all messages $\{l_{ESE}(x_k(\pi_k(j))), j = 1, \dots, S\}$ have the same value. (Note: These messages correspond to the chips generated by a common bit $v_k(1)$.) Consequently, many related operations in (5) are repeated and so can be saved. However, when the overall complexity is considered, the APP decoding cost at FEC coding constraint nodes is usually a substantial part, in which case the complexity difference among the three methods is marginal since they all have the same DEC cost. Overall, as discussed in Section II.C, the complexities of these three methods are all $O(K)$ [9].

D. Connections to Existing CDMA MUD

Most existing MUD algorithms are based on the CDMA framework and operate at bit level. They were usually presented (e.g., in [6] and [7]) differently from the chip-level detection principle discussed in Section II.B. However, we can relate them to the chip-level detection as follows.

Consider a conventional CDMA system where all $\{\pi_k\}$ are trivial, i.e., $x_k(j) = w_k(j) = s_k(j)v_k(1)$, $j = 1, \dots, S$. Substituting (5e) into (6) yields

$$\begin{aligned} l_{DEC}(v_k(1)) &= \sum_{j=1}^S s_k(j) e_{ESE}(x_k(\pi_k(j))) = \sum_{j=1}^S s_k(j) e_{ESE}(x_k(j)) \\ &= \sum_{j=1}^S \frac{2h_k s_k(j)(r(j) - E(\zeta_k(j)))}{\text{Var}(\zeta_k(j))}. \end{aligned} \quad (10)$$

When APPI or BEI is employed, the denominators in (10) for $j = 1, \dots, S$ are all equal. Consequently, (10) can be rewritten as

$$l_{DEC}(v_k(1)) = \frac{2h_k}{\text{Var}(\zeta_k(1))} \sum_{j=1}^S s_k(j)(r(j) - E(\zeta_k(j))). \quad (11)$$

The subtraction $r(j) - E(\zeta_k(j))$ in (11) can be interpreted as soft cancellation. The summation weighted by $s_k(j)$ in (11) represents de-spreading (or matched filtering). Eq. (11) is now in a form that has been used in the derivation of many CDMA MUD algorithms, e.g., [6][7]. For example, it can be used to derive the single-user matched filter with interference cancellation (SUMF-IC) method [6] where the feedback information is handled in a BEI manner. It can also be used to

derive the detector proposed in [7] where the feedback information is handled in an APPI manner.

E. Performance Comparison

We now compare the bit-error-rate (BER) performance of the above methods by simulation. For convenience of comparison with the existing multiuser detectors, we consider a long-code (the spreading sequence varies from bit to bit) CDMA system with bit-level user-specific interleavers (no chip-level user-specific interleaver). Each user employs a rate-1/2 convolutional code with generators $(23, 35)_8$ followed by a length-16 ($S = 16$) spreading code. All interleavers and spreading sequences are independent and randomly generated.

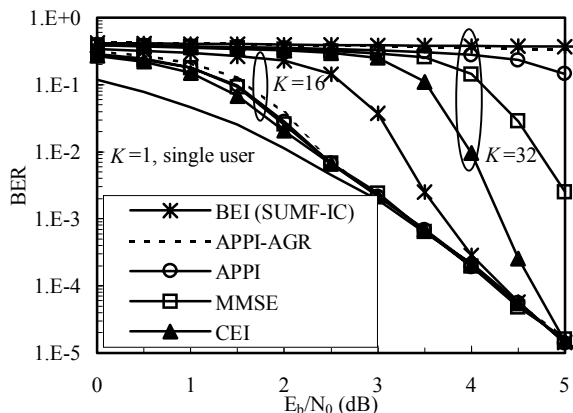


Fig. 4. Performance comparison of different detection methods for convolutionally coded systems over AWGN channels. $S = 16$, $K = 16$ and 32 , the information block length ($N_{\text{info}} = 128$) and the number of iterations ($It = 10$).

The simulated performance is shown in Fig. 4, where various detection strategies are compared, including CEI, APPI, BEI, MMSE [4], as well as the low-cost detectors in [7] (denoted by APPI-AGR)¹. BEI is equivalent to the SUMF-IC detector [6] in this case. These different iterative MUD methods have similar performance for $K \leq 16$, but the performance difference becomes significant at $K = 32$. It is seen that CEI achieves the best performance, since it best follows the “extrinsic information” principle as explained above. Interestingly, CEI outperforms MMSE, even though the latter has much higher complexity. This indicates that message passing detection may be a promising option in MUD.

In what follows, we compare other aspects of different MUD strategies in system analysis and design.

IV. COMPARISON BASED ON SNIR EVOLUTION

A. SNIR Analysis

It is usually time-consuming to assess the performance of an iterative receiver by simulation. The signal-to-noise-plus-interference ratio (SNIR) evolution technique [7][9][10][15][16] is an efficient alternative. This technique is closely related to methods such as density evolution and extrinsic-information-transfer (EXIT) chart analysis [6][14][17][18], as well as the variance transfer and multiuser efficiency evolution techniques [7][16][19]. We now apply the SNIR technique to analyze the performance of the three low-cost methods discussed in Section III.A and compare the achievable BER performance and spectral efficiency.

¹ This detector was proposed in [7] by Alexander, Grant and Reed, which is similar to the APPI method except that an approximate method is employed to calculate the interference variance.

Return to the detection algorithm in Section II. For the q th iteration, denote

- by $\gamma_k^{(q)}$ the SNIR of the messages from channel constraints towards FEC coding and spreading constraints for user- k ,
- and by $v_k^{(q)}$ the variance of the messages from FEC coding and spreading constraints for user- k towards channel constraints.

This is illustrated in Fig. 5 graphically. As shown in [9][10], we can use the following recursion to estimate $\gamma_k^{(q)}$.

$$\begin{cases} \gamma_k^{(q)} = \frac{|h_k|^2}{\sum_{k' \neq k} |h_{k'}|^2 v_{k'}^{(q-1)} + \sigma^2} \\ v_{k'}^{(q-1)} = f(\gamma_{k'}^{(q-1)}) \end{cases} \quad (12)$$

Intuitively, we can interpret (12) as follows.

- In the numerator, $|h_k|^2$ is the received signal power.
- In the denominator, $v_{k'}^{(q-1)}$ is the average variance of the feedback messages. It represents the remaining average uncertainty about $\{x_{k'}(j)\}$ after the $(q-1)$ th iteration, which determines the residual interference to other users.
- $\sum_{k' \neq k} |h_{k'}|^2 v_{k'}^{(q-1)} + \sigma^2$ is the total residual interference plus noise seen by user- k after the $(q-1)$ th iteration.
- The feedback messages refine the results, so that the operations at the channel constraint nodes make better estimates during the next iteration, as the de-spreading-decoding operations can be regarded as a filtering process. The SNIR-variance transfer function $v_{k'}^{(q-1)} = f(\gamma_{k'}^{(q-1)})$ measures the effectiveness in suppressing interference and noise for this filter.

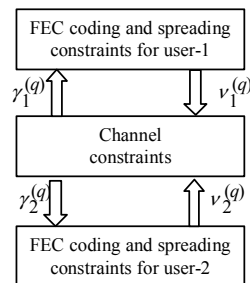


Fig. 5. Illustration of SNIR evolution for the detection algorithm for Fig. 2.

We use, respectively, $f_{\text{CEI}}(\gamma_k)$, $f_{\text{APPI}}(\gamma_k)$ and $f_{\text{BEI}}(\gamma_k)$ to denote the function f for the three methods in (7), (8) and (9). These functions can be obtained by the Monte-Carlo method as illustrated in Fig. 6 for an example system with a rate-1/3 convolutional code followed by a length-8 spreading code.

Some comments on the above SNIR evolution are in order:

- The basic assumption of SNIR evolution, similar to that of density evolution for an LDPC code [14], is that the inbound messages at each node are mutually independent. This will be referred to as the “independence” assumption. The assumption is ensured by two measures. One is using “extrinsic” information, i.e., the messages in the two opposite directions on an edge should be approximately independent. The other is using interleavers to randomize information flow and reduce the correlation among the messages arriving at a common node.

- Intuitively, chip-level interleaving should be better than bit-level interleaving in randomizing the edge connections. (See Fig. 2 and Fig. 3.) We will discuss this in Section V.
- With user-specific chip-level interleavers, CEI and BEI meet the above “independence” assumption. Thus SINR evolution for these two methods is quite accurate.
- For APPI, the feedback information is not extrinsic (see Section III.A), so the SNIR evolution technique is inaccurate for APPI (see Fig. 7 below).

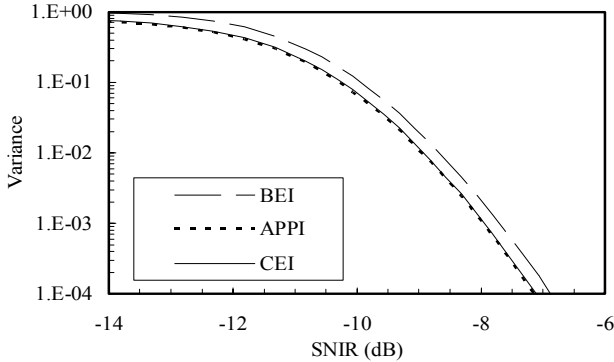


Fig. 6. Functions $f_{CEI}(\cdot)$, $f_{APPI}(\cdot)$ and $f_{BEI}(\cdot)$ for a rate-1/3 convolutional code with generators $(133, 145, 175)_8$ followed by a length-8 spreading code. The APPI and CEI curves almost overlap.

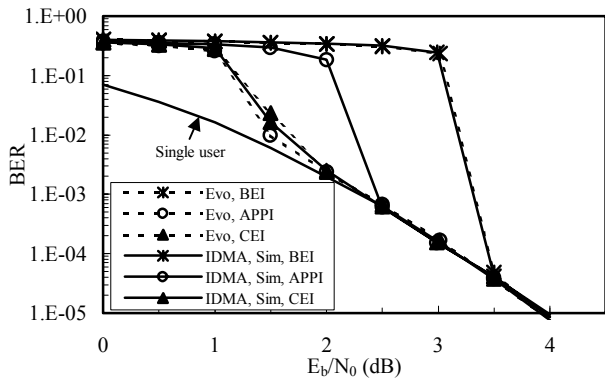


Fig. 7. The performance of three detectors obtained by SNIR evolution and simulation for IDMA systems (with user-specific chip-level interleavers). $K = 16$, $S = 8$, $N_{\text{info}} = 2048$ and $It = 20$.

As an example, we show in Fig. 7 the BERs obtained by SNIR evolution (dashed curves) for a convolutionally coded system with generators $(133, 145, 175)_8$, $K = 16$, $S = 8$, and the number of iterations (It) is 20. For comparison, we also show the simulated performance (solid curves) in an IDMA setting. As expected, for both CEI and BEI, the simulation and evolution curves agree very well; for APPI, there is a significant difference between simulation and evolution results. This is due to the problem mentioned earlier, i.e., the feedback messages at channel constraint nodes are not extrinsic.

B. Spectral Efficiency

We now compare the achievable spectral efficiency [20] based on different detection strategies. It has been demonstrated that power allocation can significantly enhance the system throughput for both IDMA and CDMA [10][20]-[22]. When the number of users (K) is large, we can approximately express (12) as

$$\gamma_k^{(q)} \approx |h_k|^2 / \left(\sum_{k'} |h_{k'}|^2 f(\gamma_{k'}^{(q-1)}) + \sigma^2 \right) \quad (13)$$

for both CEI and BEI. Based on (13), we can optimize the received power $\{|h_k|^2\}$ by linear programming as in [10].

As mentioned above, the SNIR evolution technique can not accurately predict APPI performance. Consequently, the optimization results based on (13) are not reliable for APPI (i.e., simulation results usually deviate significantly from the evolution technique results). For this reason, we will only consider CEI and BEI here.

We show in Fig. 8 the achievable sum-rates using $f_{CEI}(\cdot)$ and $f_{BEI}(\cdot)$ in Fig. 6 with power allocation. For comparison, the MMSE method² in [19] for conventional CDMA systems is also included. The same rate-1/3 convolutional code as in Fig. 7 is used. The spreading length is 8. The target BER is 10^{-5} or less. We have also carried out simulation to confirm the prediction in Fig. 8. Some results can be found in [9][10].

In Fig. 8, we can see that CEI achieves similar performance as MMSE, which agrees with our previous observation in Fig. 4. However, CEI offers a more competitive solution, since its complexity ($O(K)$ per iteration) is much lower than the MMSE complexity ($O(K^2)$ per iteration).

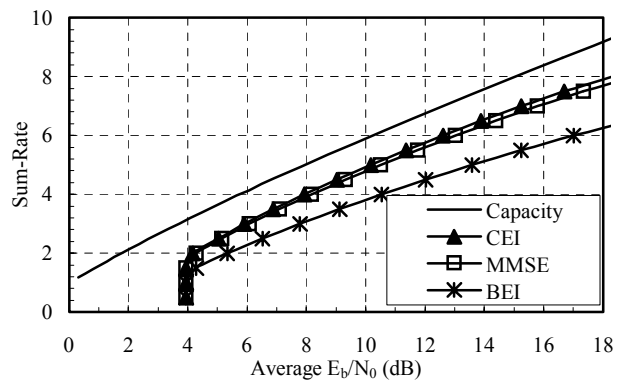


Fig. 8. Achievable sum-rates versus average E_b/N_0 for different systems with rate-1/3 convolutional codes (generators $(133, 145, 175)_8$). The target BER is 10^{-5} or less.

V. THE IMPACT OF INTERLEAVERS

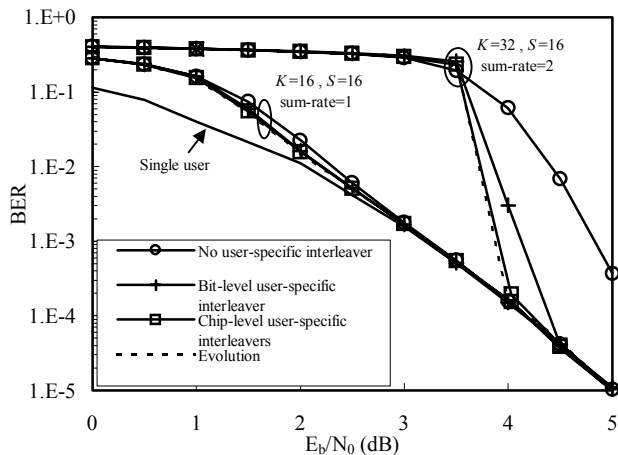
In this section, we examine the impact of different interleaving strategies on system performance. We consider in Fig. 9 (for equal and unequal power cases) three systems, respectively, with no user-specific interleaver, bit-level user-specific interleavers and chip-level user-specific interleavers. Each user employs the same convolutional code as in Fig. 4 followed by a length-16 (Fig. 9(a)) or a length-8 (Fig. 9(b)) spreading code. CEI is used at the receiver side in all cases. The power profiles listed in Table I for Fig. 9(b) are obtained by the linear programming technique in [10]. QPSK signaling is adopted. The sum-rates in Fig. 9(a) are 1 and 2 bits/chip, and in Fig. 9(b) 4 and 6 bits/chips. As we can see, when the sum-rate is relatively low, e.g., for $K = 16$, the difference between different methods is marginal. However, when the sum-rate is high, the difference becomes noticeable. Especially in unequal power cases, the system with no user-specific interleaver performs poorly.

It is interesting to observe that CEI works well even without user-specific interleavers, but its performance improves with bit-level user-specific interleavers, and further improves with chip-level user-specific interleavers. The difference between

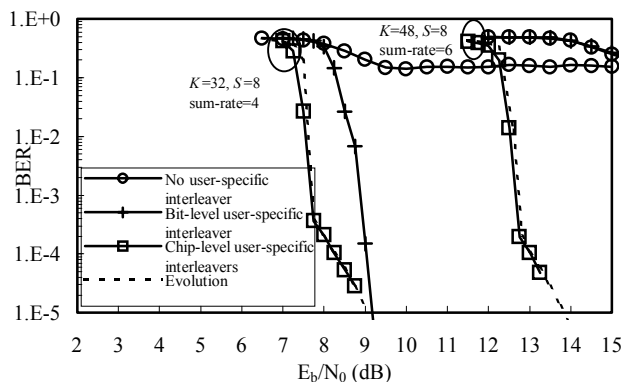
² To reduce the implementation complexity, we have adopted the unconditional MMSE method [3] in producing the results in Fig. 8.

different methods becomes more noticeable for higher sum rate. For sum rate ≥ 4 (see Fig. 9(b)), only the system with chip-level user-specific interleaver shows good agreement between simulated and predicted performance. This is because only chip-level user-specific interleaving best satisfies the “independence” assumption mentioned in Section IV.A.

The examples in Fig. 9 clearly indicate that a chip-level interleaved multiuser system is easier to analyze and optimize, and generally offers better performance.



(a)



(b)

Fig. 9. Performance comparison of multiple access systems with no user-specific interleaver, bit or chip-level user-specific interleavers over AWGN channels. $N_{\text{info}} = 2048$. (a) $S = 16$ with equal power allocation and $l_t = 10$ (b) $S = 8$ with unequal power allocation and $l_t = 50$.

TABLE I. POWER PROFILES FOR $K = 32$ AND 48 IN FIG. 9(b). THE TARGET BER IS 10^{-4} OR LESS.

	(Power level $ h_k ^2$) \times (the number of users)
$K = 32$	$0 \times 15, 3.1673 \times 1, 3.9591 \times 7, 7.1263 \times 8, 7.9181 \times 1$
$K = 48$	$0 \times 15, 3.1671 \times 4, 3.9589 \times 1, 4.7508 \times 3, 6.3344 \times 3, 7.1261 \times 3, 8.7098 \times 4, 9.5016 \times 1, 10.2934 \times 1, 11.0852 \times 5, 13.4607 \times 5, 14.2525 \times 3$

VI. CONCLUSIONS

We have examined several low-cost MUD techniques and compared their computational complexities, storage requirements and performance. We have shown that CEI, APPI and BEI provide different tradeoffs between performance and complexity. In many cases (especially when loading is low), APPI can achieve similar performance to CEI at lower detection cost. However, it is difficult to analyze the performance of APPI, since it involves “non-extrinsic”

message passing. On the other hand, BEI is an “extrinsic” message passing method and can be easily analyzed, but performance loss may occur since BEI has not utilized all the information available. CEI offers the best performance when used with chip level interleaves. CEI can also be accurately predicted using a simple SNIR evolution technique, which facilitates search-based system optimization.

REFERENCES

- [1] S. Verdú, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
- [2] M. Moher, “An iterative multiuser decoder for near-capacity communications,” *IEEE Trans. Commun.*, vol. 46, pp. 870–880, July 1998.
- [3] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, “Iterative multi-user detection for CDMA with FEC: Near-single-user performance,” *IEEE Trans. Commun.*, vol. 46, pp. 1693–1699, Dec. 1998.
- [4] X. Wang and H. V. Poor, “Iterative (turbo) soft interference cancellation and decoding for coded CDMA,” *IEEE Trans. Commun.*, vol. 47, pp. 1046–1061, July 1999.
- [5] J. Luo, K. R. Pattipati, P. K. Willett and F. Hasegawa, “Near-optimal multiuser detection in synchronous CDMA using probabilistic data association,” *IEEE Commun. Lett.*, vol. 5, pp. 361–363 Sept. 2001.
- [6] J. Boutros and G. Caire, “Iterative multiuser joint decoding: Unified framework and asymptotic analysis,” *IEEE Tans. Inform. Theory*, vol. 48, pp. 1772–1793, July 2002.
- [7] P. D. Alexander, A. J. Grant, and M. C. Reed, “Iterative detection in code-division multiple-access with error control coding,” *Eur. Trans. Telecommun.*, vol. 9, no. 5, pp. 419–426, Sep-Oct. 1998.
- [8] R. H. Mahadevappa and J. G. Proakis, “Mitigating multiple access interference and intersymbol interference in uncoded CDMA systems with chip-level interleaving,” *IEEE Trans. Wireless Commun.*, vol. 1, pp. 781–792, Oct. 2002.
- [9] Li Ping, L. Liu, K. Y. Wu, and W. K. Leung, “Interleave-division multiple-access,” to appear *IEEE Trans. Wireless Commun.*, Available at: <http://www.ee.cityu.edu.hk/~liping>.
- [10] L. Liu, J. Tong, and Li Ping “Analysis and optimization of CDMA systems with chip-level interleavers,” *IEEE J. Select. Areas Commun.*, vol. 24, pp. 141–150, Jan. 2006
- [11] R. McEliece, “Are turbo-like codes effective on nonstandard channels?” *IEEE Inform. Theory Society Newsletter*, vol. 51, no. 4, pp. 1–8, Dec. 2001.
- [12] B. Lu, G. Yue, X. Wang, and M. Madhian, “Factor-graph-based soft self-iterative equalizer for multipath channels,” *EURASIP J. Wireless Commun. Network.*, 2005(2), pp. 187–196, April 2005.
- [13] F. E. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [14] S. Chung, T. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density-parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.
- [15] D. Tse and S. Hanly, “Linear multiuser receivers: Effective interference, effective bandwidth and capacity,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 641–675, Mar. 1999.
- [16] Z. Shi and C. Schlegel, “Joint iterative decoding of serially concatenated error control coded CDMA,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1646–1653, Aug. 2001.
- [17] H. Gamal and A. Hammons, Jr., “Analyzing the Turbo decoder using the Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.
- [18] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 40, pp. 1727–1737, Oct. 2001.
- [19] G. Caire, R. Mueller, and T. Tanaka, “Iterative multiuser joint decoding: optimal power allocation and low-complexity implementation,” *IEEE Trans. Inform. Theory*, vol. 50, pp.1950–1973, Sept. 2004.
- [20] S. Verdú and S. Shamai, “Spectral efficiency of CDMA with random spreading,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 622–640, Mar. 1999.
- [21] A. J. Viterbi, “Very low rate convolutional codes for maximum theoretical performance of spread spectrum multiple-access channels,” *IEEE J. Select. Areas Commun.*, vol. 8, pp. 641–649, Aug. 1990.
- [22] N. Chayat and S. Shamai, “Convergence properties of iterative soft onion peeling,” in *Proc. ITW 99*, June 1999, pp. 9–11.