Particle Swarm Interleavers Algorithm of IDMA System

Zhao Zhijin  Yue Keqiang  Shen Lei
School of Telecommunication, Hangzhou Dianzi University
Hangzhou, 310018, China

Abstract

Interleavers of IDMA system are employed as theory means of user separation which are implemented independently and randomly. But correlation coefficients of some Interleavers sequences are near to 1, which seriously affect the performance of communication. In this paper, interleavers based on Particle Swarm algorithm is proposed. A fitness function based on covariance matrix is used and the optimum interleavers are computed by Particle Swarm algorithm. The simulation results show that the proposed algorithm has better performance than un-random interleavers and independent random interleavers in the case of large users.

Key words: Particle Swarm Algorithm; covariance matrix; IDMA

1. INTRODUCTION

Code-division multiple access (CDMA) is an attractive multiple-access technique that has been widely used. It possesses many attractive features such as dynamic channel sharing and robustness against fading. Interleave Division Multi Access (IDMA) [1-4] as a special case of CDMA employs chip-level interleavers for user separation and inherits many advantage from CDMA e.g., diversity against fading and mitigation of the worst-case other-cell user interference. Furthermore, IDMA allows very simple chip-by-chip (CBC) iterative multi-user detection (MUD) strategy which has low complexity.

The key principle of IDMA is interleavers. Interleavers are generally generated independently and randomly in the previous work [1-4]. There is a potential risk that some interleavers of users may be with high correlation. A novel interleavers [5] based on evolutionary algorithm has been proposed. In [5], it can find optimal interleavers sequence for communication, but the evolutionary algorithm usually converges to local optimal solution. The particle swarm algorithm has excellent global search capability. This paper uses the particle swarm algorithm to find global optimization, and can get the optimal solution in the case of lesser number of iterations. The simulations show that the proposed algorithm has better performance than that in [5].

2. IDMA COMMUNICATION SYSTEM AND DETECTION METHOD

Efficiency and adaptability are the major features for the IDMA mobile communication system. As in [2] the usual transmitter structure for an IDMA system with $K$ users is depicted in figure 1.

As show in Fig1, there are $K$ users in IDMA system, and in the transmitter end, the code sequence of $k$th $d_k = [d_k(1), ..., d_k(I)]$, where $I$ is the length of information stream. $d_i \in [-1, 1]$. $d_k$ is spread by a length $-S$ spreading sequence $g_k$. The information stream after spreading is then $C_k = [C_k(1), ..., C_k(J)]$, $g_k$ is the same for different users, where $J = IS$ is the length of chip. A chip-level interleaver $\gamma_k$ of length $J$ is then applied to produce the transmitter signals $X_k = [x_k(1), ..., x_k(J)]$. $h_k$ the channel coefficient for user $k$. 

Figure 1 IDMA transmitter and receiver structures

In the received end, the iterative sub-optimal receiver structures are elementary signal estimator (ESE) and posteriori probability (APP) decoders (DECs) for $K$ single-user. The received signal $r(j)$ for a fully synchronized single-path channel without memory is expressed as (after chip-matched filtering [2]):

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

(1)

Where $n(j)$ is a noise sampled. For a particular $k$, rewrite Eq.1 as

$$r(j) = h_k x_k(j) + \xi_k(j)$$

(2)

$$\xi_k(j) = \sum_{k \neq k} h_k x_k(j) + n(j)$$

(3)
Where $\xi_k(j)$ is the summation of the received signals from other $k-1$ users (except user $k$) plus noise. According to central limit theorem, $\xi_k(j)$ in (2) can be approximated by a Gaussian random signal for a large $k$. The means $E(.)$ and variances $V(.)$ of all $\xi_k(j)$ are available.

The chip-chip detection algorithm can be summarized as [2, 4]:

$$E(x_k(j)) = \tanh\left\{e_{DEC}\left[\xi_k(j)\right]/2\right\}, \forall k, j \quad (4)$$

$$V(x_k(j)) = 1 - \left\{E[\xi_k(j)]\right\}^2, \forall k, j \quad (5)$$

$$E(\xi_k(j)) = \sum_{k' \neq k} h_k E(x_k(j)), \forall k, j \quad (6)$$

$$V(\xi_k(j)) = \sum_{k' \neq k} |h_k|^2 V(x_k(j)) + \sigma^2, \forall k, j \quad (7)$$

$$e_{ESE}(x_k(j)) = \begin{bmatrix} V(\xi_k(j))\end{bmatrix}^{-1/2} \left\{r(j) - E(\xi_k(j))\right\}, \forall k, j \quad (8)$$

$e_{DEC}(x_k(j))$ is the interleaved APP which obtained from $DEC$, at first iteration, $e_{DEC}(x_k(j)) = 0$. After updating $e_{DEC}(x_k(j))$ can update $E(x_k(j))$ and $V(x_k(j))$, according to the equation (4) and (5). $e_{ESE}(x_k(j))$ can be updated according to equation (6)-(8), de-interleaving $e_{ESE}(x_k(j)) = \frac{2h_k}{V(x_k(j))}\{r(j) - E(\xi_k(j))\}$ is performed and then $APP$ decoding is performed in the $DEC$. During the final iteration, the $DEC$ produces hard decisions on information bits $\{\hat{d}_k(i), \forall k, i\}$.

### 3. PARTICLE SWARM ALGORITHM INTERLEAVER

#### 3.1 Related definition and operation of PSO

Particle swarm optimization (PSO) [6,7,8] is a new evolutionary computation technique combining the social psychology principles in socio-cognition human agents and evolutionary computation. It is shown that PSO has advantages over evolutionary algorithm (EA) for efficiently finding the optimal or near-optimal solutions. The number of particles in the population is denoted as $N_p$. Let $x_i^t = [x_{i1}^t, x_{i2}^t, \ldots, x_{id}^t]$ be the position of particle $i$ $(1 \leq i \leq N_p)$ with $D$ bits at iteration $t$, where $x_{id}^t$ is the $d$th $(1 \leq d \leq D)$ bit of the position of particle $i$. The velocity of particle $i$ at iteration $t$ is denoted as $v_i^t = [v_{i1}^t, v_{i2}^t, \ldots, v_{id}^t]$, $v_{id}^t \in R$. Each particle in the swarm is assigned a fitness value indicating the merit of this particle such that the swarm evolution is navigated by best solutions. Let $p_i^t = [p_{i1}^t, p_{i2}^t, \ldots, p_{id}^t]$ be the best solution that particle $i$ has obtained until iteration $t$, and $p_n^t = [p_{n1}^t, p_{n2}^t, \ldots, p_{nd}^t]$ be the best solution obtained from $p_i^t$ in the population at iteration $t$. The objective function provides the mechanism for evaluating the fitness of each particle.

In order to address the interleave problem by PSO, the operation of velocity and solution of particles are defined as follows:

1. **Velocity**: A velocity is defined as a permutation of $D$ elements, which is a list of transpositions. The velocity is defined as: $v = \{(i_j, j_k), i_j, j_k \in D, k \in \{1, 2, \ldots, n\}\}$ where $n$ is length of this list.

2. **Swap Sequence**: Assuming a particle position is $x_i$, Swap Operator $\{m_i, n_i\}$ is defined as exchanging the value of $m_i$ with $n_i$ in position $x_i$, that is $x_i = x_i + \{m_i, n_i\}$, where $x_i$ is a new location. For example: $x_i = (1, 2, 3, 4, 5)$, $(m_i, n_i) = (1, 2)$, then $x_i = (2, 1, 3, 4, 5)$.

3. **Plus Operator**: It includes position plus velocity, and velocity plus velocity. Let $v_i, v_j$ be three velocities and $x_i$ be a position. $v_i \oplus v_j$ is merging two velocities into a new velocity. $v_i + x_k$ is a new position and is found by applying the first transposition of $v_i$ to $x_k$. For example: $v_i = (1, 2) v_j = (3, 4)$, and $x_k = (2, 1, 3, 4, 5)$, then $v_i \oplus v_j = (1, 2) \oplus (3, 4)$; $v_i + x_k$ can express as $x_k$ merging $(1, 2)$ which get a new position $x_k = (1, 2, 3, 4, 5)$.

4. **Minus operator**: Here giving position minus position. Let $x_k$ and $x_i$ be two positions. The $x_k - x_i$ is defined as the velocity $v$ which is a swap sequence. We can swap the values in $x_i$ according to $x_k$ from left to right to get $v$. So there must be an equation $x_i = x_k + v$. For example: $x_i = (1, 2, 3, 4, 5)$, $x_k = (2, 1, 3, 4, 5)$, because of $x_i(1) = x_k(1) = 1$, so the first swap operator is $v_1 = (1, 3), x_j = x_k + v_1$, and we can get the following result: $x_j = (1, 2, 3, 4, 5)$, because of $x_j(2) = x_i(3) = 2$, so the second swap operator is $v_2 = (1, 2), x_i = x_j + v_2$. Finally, we get $x_k - x_i = v = v_1 \oplus v_2 = (1, 2) \oplus (1, 3), (2, 3)$.

5. **Multiplication operator**: Here giving a real coefficient multiply velocity. Let $c$ be a real coefficient and $v$ be a velocity, where $c \in (0, 1)$. Multiplication means truncate the length of $v$, then let the length of the new velocity $c \times v$ be the greatest integer smaller than or equal to $v$. For example: $v = (2, 3, 1, 4, 5), i = 4$, $c = 0.8$, then $c \times v = (3.2, 1.6, 0, 0.8, 0.4)$. At last, we can get that $c \times v = \{(2, 3), (1, 3), (0, 4, 5)\}$.

Based on above definition, the formula of PSO can be modified as follows:

$$v_{id}^{t+1} = \omega v_{id}^t \oplus c_1 r_1 (p_{id}^t - x_{id}^t) \ominus c_2 r_2 (p_{id}^t - x_{id}^t) \quad (9)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (10)$$

Where $c_1$ and $c_2$ are acceleration coefficients, and $r_1$ and $r_2$ are uniform random numbers distributed in $[0, 1]$.

### 3.2 Particle Swarm Interleavers (PSOI)

The chip length of the sequence is $L$, so theoretically the population of interleaver is $L!$. Interleavers are used as signature sequences for user-separation. Although there are $L!$ possible solutions, when $L$ is small, and the number of user is...
large, correlation coefficients of some Interleavers sequences are near to 1. Then, the Particle Swarm interleavers can get the optimum solutions which can reduce the correlation between the sequences.

Define $I$ the length of information stream of every user, $S$ the length of spreading sequence, $J = I \times S$ the length of chip and is also the length of interleaver. The object of PSOI is to search the top $K$ solutions among $L$ possible solutions with low correlation. To reduce the complexity, we define $N$ as the number of possible solutions.

In Particle Swarm interleaver algorithm, covariance matrix [5] is used as fitness function. $X$ is the $N \times J$ data matrix ( $N$ possible solutions and everyone with a $J$ -length interleaver) , $X(n,j) \in \{-1,1\}$ , where $(n = 1,...,N, j = 1,...,J)$; INDEX is a $N \times J$ array whose elements are the corresponding index of $X$, and $R$ is correlation coefficients matrix for $X$ with $N \times N$ elements. $R(i,j)$ is the elements of $R$ which is generated as follows:

$$COV = E\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$ (11)

$$R(i,j) = \frac{Cov(i,j)}{\sqrt{Cov(i,i)Cov(j,j)}}$$ (12)

Where $Cov$ is the covariance matrix, $E$ is mathematical expectation and $\mu_i = E(X_i)$ . $X_i$ is the row vector of $X$. After computing the $R(i,j)$, we should record index which is address of the elements of $R(i,j)$.

Thus, the proposed Particle Swarm interleaver algorithm proceeds as follows:

Step 1: generate an initial population and Velocity, $X$ and its INDEX

Step 2: compute the fitness of each particle using Equations (11) and (12), then set $p^g_{gd}$ and $p^d_{id}$, where $g$ is the index of the particle which has the highest fitness value.

Step 3: update $v^d_{gd}$ according to Equations (9);

Step 4: update INDEX according to Equations (10), at the same time, update the values in $X$ according the INDEX ;

Step 5: compute the fitness value of each particle again, and update $p^g_{gd}$ and $p^d_{id}$;

Step 6: if it reaches the maximum generation, then go to the end; if not, go to step 2.

Select the top $K$ interleavers from $N$ possible solutions which have the lower correlation coefficients.

4. SIMULATION RESULTS

Particle Swarm interleavers (PSOI) are compared with the results of random (Random) interleavers, un-random (Un-random) interleavers and Evolutionary [5](EI) interleavers. We consider a system without code in AWGN channel [2]. The simulation parameters are set as follows: $I = 10$, $S = 6$, $\omega = 1, c_i = 2$, $h_k = k, k \in (1,2,...,K)$, $K$ is the user number. The times of Monte Carlo simulation is 10000. 8 iterations are used.

Figure 2 gives out the BER (Bit Error Ratio) versus SNR (Signal to Noise Ratio) when $K$ is 5. As can be seen from the figure 2, PSOI BER performance is much better than EI, Un-random and Random. Un-random is the worst performance because that the correlation coefficients of un-random interleavers is high. When $SNR > 9dB$, the BER of PSOI is nearly to zero.

Figure 3 shows the BER versus SNR when $K$ is 25. We can get the same conclusion as in figure 3. The correlation coefficients of un-random interleavers is high, the performance is poor.

Figure 4 gives out the BER versus the number of users when SNR is $6dB$. As can be seen from the figure 4 the proposed
PSOI has lower BER than EI, Un-random and Random.

![Graph showing BER versus number of users]

**Figure 4**: BER versus number of users

5. CONCLUSIONS

The algorithm of using PSO to compute interleavers is presented. The proposed algorithm is compared with un-random interleavers, random interleavers and Evolutionary interleavers, experimental results show that PSO interleavers has better performance.

REFERENCES


