Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 151057, 9 pages
doi:10.1155/2010/151057

Research Article

Suboptimal Partial Transmit Sequence-Active Interference Cancellation with Particle Swarm Optimization

Poramate Tarasak, Zhiwei Lin, Xiaoming Peng, and Francois Chin

*Department of Modulation and Coding, Institute for Infocomm Research, Agency for Science Technology and Research,
1 Fusionopolis Way, no. 21-01 Connexis, Singapore 138632*

Correspondence should be addressed to Poramate Tarasak, ptarasak@i2r.a-star.edu.sg

Received 31 December 2009; Revised 16 June 2010; Accepted 11 August 2010

Academic Editor: Chi Ko

Active interference cancellation (AIC) is an effective technique to provide interference avoidance feature for an ultrawideband (UWB) OFDM transmitter. Partial transmit sequence-AIC (PTS-AIC), which was recently proposed as an improvement of AIC, requires high computational complexity by doing the exhaustive search of all possible weighting factors whose number grows exponentially with the number of subblocks used. To reduce the complexity of PTS-AIC, this paper proposes a suboptimal way, called particle swarm optimization (PSO), to choose the weighting factors suboptimally without much performance degradation. Both continuous and discrete versions of PSO have been evaluated, and it has been shown that the discrete PSO is able to reduce the complexity significantly without sacrificing the performance of PTS-AIC in many cases.

1. Introduction

Ultrawideband (UWB) communication, a spectrum underlay system, has a very small power spectral density that spans over hundreds of megahertz. While a UWB device must endure interference from the primary narrowband devices, the UWB transmission must not cause interference back to them. There are several works studying the impact of interference if the UWB system is to coexist with other narrowband systems, for example, [1–4]. These studies indicate performance degradation as a result of mutual interference between UWB and narrowband systems.

Due to the secondary nature of UWB devices, it is their requirement to avoid causing the interference in the first place. One approach is to enhance the UWB device with the "detect-and-avoid" (DAA) [5] capability, sensing any ongoing narrowband transmissions and intelligently keeping away from the overlapped spectrum. If the narrowband transmission is found, the UWB device will adjust its transmission such that the effect of the UWB transmission will be negligible at the primary device receiver. DAA has been an interesting research topic on UWB recently. The strong interest of DAA is attributed to widespread usage of wireless applications sharing the same or overlapped part of the spectrum band. For example, current important narrowband systems that share parts of the UWB spectrum are WiMAX at the 3.5 GHz frequency range and IEEE802.11a at the 5 GHz frequency range. In the future, it seems inevitable for the UWB device to have the DAA feature.

This paper focuses on the avoidance part of DAA for the UWB system that employs OFDM transmission such as WiMedia standard [6]. In [6], tone nulling at the overlapped narrowband spectrum (referred to as an interference band) is suggested as an avoidance technique. Although tone nulling completely removes the interference at the exact center frequency corresponding to the nulled tones, there still exists interference caused by sidelobes of the remaining tones present elsewhere in the interference band [7]. One efficient technique to mitigate sidelobe interference is active interference cancellation (AIC) proposed in [7]. In addition to removing the subcarriers that lie inside the interference band, AIC removes two more subcarriers beside them and replaces the removed tones with the computed "AIC tones".

The purpose of placing AIC tones is to generate "negative interference" in order to cancel the sidelobe interference from the remaining subcarriers.

Many extensions and improvements of AIC have been proposed recently. In a subcarrier-based antenna selection system, a new AIC formulation was proposed in [8]. The problem of sidelobe interference coming from a superposition of the transmissions from all antennas necessitates a new AIC for all antennas.

A few AIC algorithms with low complexity were proposed in [9]. The algorithms are based on simplifying matrix computation in the AIC. The complexity saving in [9] comes at a price of degraded interference cancellation performance although it is claimed that the algorithm is still better than the original AIC in the worst case.

The following works attempt to deepen the notch spectrum obtained with AIC. Extended AIC inserts the so-called extended AIC tones between the usual AIC tones, and they generate better negative interference [10]. However, since the extended AIC tones are placed in between the usual subcarrier positions, orthogonality between the subcarriers is lost and the bit error rate (BER) curve shows an error floor. Reference [11] proposes three enhancements of the AIC by cyclic shifting, phase shifting, or joint cyclic and phase shifting the data subcarriers. Doing so leads to modification of the spectrum and yields smaller remaining interference after the AIC operation. The algorithms in [11] increase the cancellation performance of AIC significantly with the drawbacks of high complexity and the requirement of side information. Recently, another technique, so-called partial transmit sequence-AIC (PTS-AIC), was proposed in [12]. It is essentially a novel application of partial transmit sequence that has been applied in OFDM in order to reduce peak-to-average power ratio (PAPR) [13, 14]. Adjacent subblock partitioning and interleaved subblock partitioning were proposed for the PTS-AIC. It was shown that PTS-AIC enhances the performance of AIC and provides more flexibility in parametrizing the algorithm. However, selecting the optimal parameters for PTS-AIC is very complex, especially when a large number of subblocks are used. High complexity comes from the exhaustive search for the optimal weighting factors whose number grows exponentially with the number of subblocks [12].

Particle swarm optimization (PSO) is a heuristic bio-inspired optimization algorithm that is well suited to solve high-dimensional and multimodal optimization problems [15–18]. The PSO-based CDMA multiuser detection has been reported in [19–21] where exponential complexity in the number of users has been resolved by the PSO. Power allocation problem in CDMA was solved by the PSO in [22] where a few constraint handlings were investigated and extensive studies on the parameters of PSO were given. PSO was applied in PTS for OFDM PAPR reduction in [23] to achieve much lower complexity. In [24], exponential complexity in the number of sensors was solved by the PSO in a sensor scheduling problem which optimizes a group of sensors for target tracking under the performance and cost constraints. More recently, PSO was exploited in determining linear precoding for a linear MMSE multiuser MIMO receiver, and it was shown to outperform the block diagonalization approach [25].

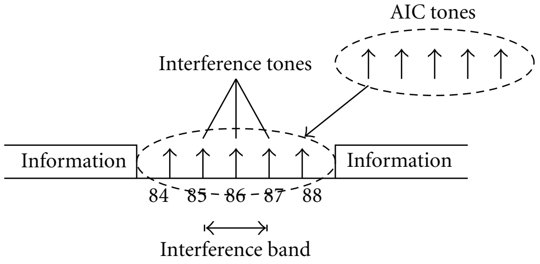


Figure 1: Interference band, interference tones, and AIC tones. The computed AIC tones replace the interference tones and the two tones beside them.

This paper proposes a PSO as a suboptimal approach to optimize weighting factors for PTS-AIC with the main purpose of reducing complexity. Unlike [23], both continuous and discrete (binary) version [16] of PSO are considered. It is shown that PSO can be applied to PTS-AIC effectively and can approach the performance of optimal PTS-AIC in many cases with much lower complexity. The PSO algorithm allows us to enhance the performance of PTS-AIC by using a larger number of subblocks whose complexity is prohibitive for the optimal exhaustive search.

The paper is organized as follows. Section 2 provides the background on AIC. Section 3 reviews the PTS-AIC as well as two types of subblock partitioning proposed in [12]. Section 4 discusses PSO in both continuous and discrete versions as well as their complexity analysis. Section 5 shows the simulation results and discussion. Conclusion is given in Section 6.

Notations. Bold letters represent matrices or vectors. $(\cdot)^t$ is a transpose. $(\cdot)^H$ is a Hermitian transpose. $|\cdot|$ is an absolute value. $\|\cdot\|$ is the l_2-norm of a vector.

2. AIC

We hereby describe the AIC algorithm in a matrix formulation. Detailed description can be found in the original paper [7].

Some definitions are required as follows. Interference band is the frequency band that overlaps with the narrowband spectrum. Interference tones are the UWB OFDM subcarriers that are present in the interference band. AIC tones obtained by the AIC algorithm are to replace the interference tones and two subcarriers beside the interference tones. Figure 1 shows an example of interference band, interference tones, and AIC tones.

Let $\mathbf{X} = [X(0), \dots, X(N-1)]^t$, where $X(k), k = 0, \dots, N-1$ represents original frequency-domain data symbols and N is the number of subcarriers (FFT size). We can write

$$\mathbf{Y} = \frac{1}{N}\mathbf{P}\mathbf{X}, \tag{1}$$

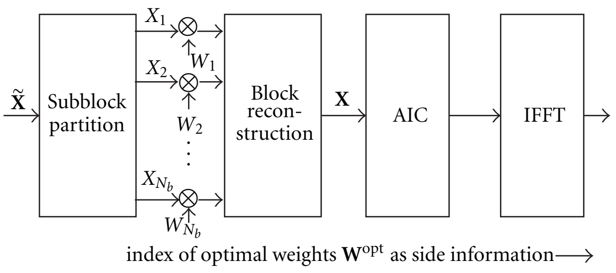


Figure 2: PTS-AIC transmission model.

where $\mathbf{Y} = [Y(0),\ldots,Y(MN-1)]^t$ is an upsampled symbol vector with an upsampling factor M and $\mathbf{P}$ is an upsampling matrix of size $MN \times N$ with element $P(l,k) = \sum_{n=0}^{N-1} \exp(j2\pi(n/N)(k-l/M))$.

Suppose the interference tones start from the pth to $(p + N_i - 1)$th subcarriers, where N_i is the number of interference tones. Define a nulling matrix $\mathbf{T}$ which is constructed from an $N \times N$ identity matrix, and put zeros at the $(p-1)$th to $(p+N_i)$th diagonal elements. The (upsampled) interference generated from the sidelobes of the other subcarriers can be computed as

$$\mathbf{d}_I = \mathbf{P}_s\mathbf{T}\mathbf{X}, \qquad (2)$$

where $\mathbf{P}_s$ is a submatrix of $\mathbf{P}$ by taking its row corresponding to upsampled spectrum in the interference band, that is, from the $(Mp+1)$th to $(M(p+N_i-1)+1)$th rows of $\mathbf{P}$. Thus, the size of $\mathbf{P}_s$ is $(M(N_i-1)+1)\times N$. Let a new set of AIC subcarriers be a column vector $\mathbf{h}$ of length N_i+2. The AIC subcarriers are computed to generate "negative interference" to cancel $\mathbf{d}_I$, that is,

$$\mathbf{P}_n\mathbf{h} = -\mathbf{d}_I, \qquad (3)$$

where $\mathbf{P}_n$ is a submatrix of $\mathbf{P}_s$ by taking its column corresponding to the positions of AIC tones, that is, from the pth to $(p+N_i+1)$th columns of $\mathbf{P}_s$. The size of $\mathbf{P}_n$ is then $(M(N_i-1)+1)\times(N_i+2)$. Since $\mathbf{P}_n$ is not a square matrix, to solve (3), one constructs a least-squares problem which finds $\mathbf{h}$ that minimizes the squared error [7],

$$\mathrm{SE} = \|\mathbf{P}_n\mathbf{h} + \mathbf{d}_I\|^2. \qquad (4)$$

A well-known least-squares solution to (4) is the Moore-Penrose generalized inverse (pseudoinverse) [26], in the form

$$\mathbf{h} = -\left(\mathbf{P}_n^H\mathbf{P}_n\right)^{-1}\mathbf{P}_n^H\mathbf{d}_I = -\mathbf{V}\mathbf{T}\mathbf{X}, \qquad (5)$$

where $\mathbf{V} = (\mathbf{P}_n^H\mathbf{P}_n)^{-1}\mathbf{P}_n^H\mathbf{P}_s$ is of size $(N_i+2)\times N$ and can be precomputed. Then, $\mathbf{h}$ is inserted at the nulled tone positions. IFFT performs on this new block with the AIC tones in place to construct an OFDM symbol.

3. PTS-AIC

Figure 2 illustrates the transmission model for PTS-AIC. First, $\tilde{\mathbf{X}}$, a symbol block of length N, is partitioned into N_b subblocks of equal size N/N_b. Each subblockX_{N_b} is multiplied by a weighting factorW_{N_b}. The weighting factor in this paper is chosen from a usual unit-energy complex PSK constellation Θ of size N_w, for example, $\Theta = \{1,-1\}$ when $N_w = 2$, and $\Theta = \{1, j, -1, -j\}$ when $N_w = 4$. Then, a symbol block is reconstructed from each subblock as $\mathbf{X}$ before it is processed with the AIC algorithm. The PTS-AIC algorithm determines an optimum N_b-tuple weighting factor $\mathbf{W}^{\text{opt}} = \{W_1, W_2, \ldots, W_{N_b}\}$ such that the remaining interference power inside the interference band after performing the AIC is minimum, that is,

$$\mathbf{W}^{\text{opt}} = \arg\min_{\mathbf{W}\in\Theta^{N_b}} \|-\mathbf{P}_n\mathbf{V}\mathbf{T}\mathbf{X}(\mathbf{W}) + \mathbf{P}_s\mathbf{T}\mathbf{X}(\mathbf{W})\|^2, \qquad (6)$$

where $\mathbf{X}(\mathbf{W})$ is a reconstructed symbol block computed by an N_b-tuple weighting factor $\mathbf{W}$ and Θ^{N_b} is a set of N_b-tuple weighting factor where its element is chosen from Θ.

The PTS-AIC algorithm is characterized by the parameters N_b and N_w. We can adjust both parameters such that the interference cancellation performance meets the target while the complexity is affordable. As either N_b or N_w increases, the performance improves while the complexity increases. The complexity of PTS-AIC is determined by the number of all possible weighting factors to find $\mathbf{W}^{\text{opt}}$, which is $(N_w)^{N_b}$. Since the complexity grows exponentially with N_b, one cannot increase the number of subblocks to a very large value to improve the performance, as the number of comparisons is prohibitive.

There are two types of subblock partitioning considered in this paper. The first type is *adjacent subblock partitioning* in which each subblock is constructed from adjacent subcarriers of the original symbol block. The second type is *interleaved subblock partitioning* in which each subblock is constructed from the subcarriers of distance N_b in the original symbol block. Both types of partitioning are depicted in Figure 3.

Since PTS-AIC modifies the transmission block by the weighting factors, the receiver must be aware which set of weighting factors is applied so that it can recover the original symbol block. This can be done by sending the index of the optimum weighting factor as side information that amounts to $N_b\log_2(N_w)$ bits. Once the receiver knows the applied weighting factors, multiplication of their complex conjugates to the received signal block after FFT returns the original data block.

Note one major difference between PTS conventionally applied to reduce PAPR and the proposed PTS-AIC. While the conventional PTS measures the PAPR of the (upsampled) time-domain signal *after* IFFT, the proposed PTS-AIC measures the interference power of the upsampled frequency spectrum *before* IFFT.

3.1. Performance Measure. In interference avoidance mechanisms, it is interesting to know how much interference power is remaining inside the interference band. For AIC and PTS-AIC, this is equivalent to the squared error terms after performing interference cancellation as defined in (4).

Generally, the "instantaneous" remaining signal power inside the interference band can be found as

$$I = \sum_{l=Mp}^{M(p+N_i-1)} \left| \widetilde{Y}(l) \right|^2, \tag{7}$$

where $\widetilde{Y}(l)$ is the upsampled frequency-domain signal with AIC or PTS-AIC processed. I is the remaining power of the upsampled spectrum at the interference band after AIC or PTS-AIC being performed on a particular symbol block.

Although the mean of the remaining interference power, $E[I]$, is normally used to compare the algorithms as in [11], a more complete picture is captured by computing the complementary cumulative distribution function (CCDF), $P(I > I_s)$, where I_s is a target remaining interference power. This bears an analogy with the performance measure of a PAPR reduction algorithm of OFDM for which the CCDF is widely used. With a given I_s, the CCDF determines the probability of remaining interference power above the target I_s. We refer to I_s of which $P(I > I_s) = x\%$ as *x%-excess interference power*.

4. Particle Swarm Optimization (PSO)

PSO algorithm was described in analogy with an activity of bird flocking or fish schooling [17]. Imagine a group of birds trying to locate a position in the field with the highest concentration of food. Each bird flies over the field and detects the concentration of food at its location. Each bird has an ability to remember its own best location and is aware of the group's best location. Its flying path depends on the previously observed location and is influenced by its own best location and the group's best location. Each bird has a chance to "fly over" the best location previously found and therefore observes the surrounding for a possibly better location. As time passes, most birds will be crowded at the best location they found as a group.

In the optimization problem, each bird is called a particle and its location represents an n-dimension solution candidate in the n-dimensional solution space. In the context of the PTS-AIC, a particle represents a vector of weighting factors $\mathbf{W}$. The location of the particle is reflected in the elements of $\mathbf{W}$. The concentration of food corresponds to the remaining interference power of PTS-AIC which is an objective value of the objective function in (6). The best location corresponds to the particle whose objective value is the minimum among other particles.

One round of observations from all particles corresponds to one iteration in the PSO. Suppose P is the number of particles and Q is the number of iterations. The nth dimension of the current location of the particle p is x_n^p while that of the particle's best location is $pbest_n^p$. The nth dimension of the group's best location is $gbest_n$. The PSO is initialized by generating random locations for every particle. In one iteration, each particle evaluates the objective function and replaces its own best and the group's best locations if a better solution is found. The particle's best and the group's best locations will determine the change of location for the next iteration through a velocity variable which indicates the amount and direction (positive or negative) of change in distance from the current location. The update of the location for the next iteration is done by updating the velocity of each dimension

$$\begin{aligned} v_n^p(i+1) = c_0 \cdot v_n^p(i) + c_1 \cdot r_1 \cdot \left(pbest_n^p - x_n^p(i)\right) \\ + c_2 \cdot r_2 \cdot \left(gbest_n - x_n^p(i)\right), \end{aligned} \tag{8}$$

where $v_n^p(i)$ is the velocity of the nth dimension of the pth particle at the ith iteration. The velocity is initialized to be zero in the first iteration. c_0, c_1, and c_2 are parameters of the PSO used to adjust the influence to the path of solutions from a particle's best location and the group's best location. r_1, r_2 are uniform random variables whose values are between 0 and 1 so as to introduce uncertainty of the influence. The update of the location is simply

$$x_n^p(i+1) = x_n^p(i) + v_n^p(i+1), \tag{9}$$

where the time duration (supposed to be multiplied with the velocity) is assumed to be one. This completes the task for one particle in one iteration. The algorithm is repeated from the point of evaluation of the objective function for all particles and for Q iterations.

The original PSO algorithm was designed for a problem with continuous parameters. Since the optimization parameters in PTS-AIC are the weighting factors which are discrete, the location of a particle has to be quantized to the nearest point in the constellation set before the objective function is evaluated. This is done after the location update. The number of dimensions of each location in PTS-AIC (for the binary case) is N_b. PSO for PTS-AIC with $N_w = 2$ (binary case) can be described by Algorithm 1.

Another way to tackle discrete parameters is to apply the PSO algorithm modified for binary parameters proposed in [16]. The idea is to work with dummy continuous parameters, transform them to have the range from 0 to 1, and consider the results as probabilities of the optimization parameters taking value 0 or 1. Therefore, there is no need to round off the parameters as in the previous algorithm. The transformation function proposed in [16] is a sigmoid limiting function, $S(v) = 1/(1 + e^{-v})$, whose domain is $(-\infty, \infty)$ and whose range is $(0, 1)$. $S(v_n^p(i + 1))$ is the probability of $x_n^p(i + 1)$ equal to one, and the location is updated by generating a random number uniformly distributed over $[0, 1]$ and comparing it with $S(v_n^p(i + 1))$. We refer to this discrete version as discrete PSO (DPSO). The only change from Algorithm 1 is at Step (6), which should be replaced by the location update algorithm according to Algorithm 2.

To handle nonbinary parameters ($N_w > 2$) for both PSO and DPSO, we can simply extend the dimension of location and velocity vectors and convert the binary tuples into symbols on the constellation. For example, for $N_w = 4$, the dimension of each vector will be $2 \cdot N_b$ and two dimensions in the location are converted into one QPSK symbol.

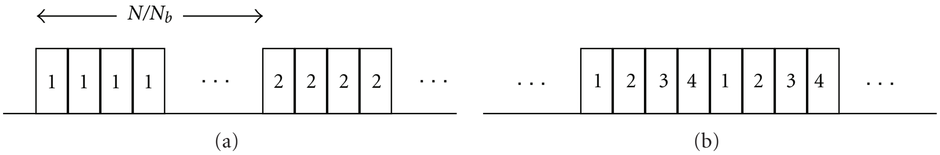


Figure 3: (a) Adjacent subblock partitioning. (b) Interleaved subblock partitioning.

```
(1) Initialize parameters $P, Q, c_0, c_1, c_2$. Reset the vectors
      pbest($p$),gbest for particle best and group best locations.
      Reset $Ipbest(p), Igbest$ for particle best and group best objective values.
(2) Generate $P$ random locations, $x(p)$. Each location is an
      $N_b$-dimension binary vector.
(3) Consider the first particle.
(4) Evaluate the objective function (7) with $x(p)$ as weighting
      factors and update pbest($p$), gbest, $Ipbest(p), Igbest$ if necessary.
(5) Update the velocity according to (8) for each dimension.
(6) Update the location according to (9) for each dimension.
(7) Quantize the location to the nearest binary vector.
(8) Repeat from 4: for the next particle until all $P$ particles are considered.
(9) Repeat from 3: for $Q$ iterations.
(10) Return gbest as the solution of a vector of weighting factors.
```

Algorithm 1: PSO PTS-AIC.

4.1. Complexity Analysis. The main complexity of PTS-AIC lies in computing the remaining interference power in (7). For PTS-AIC, exhaustive search requires this computation and comparison for $(N_w)^{N_b}$ times. For PSO/DPSO PTS-AIC, the number of computations and comparisons of (7) is $P \cdot Q$. The updating operations of PSO/DPSO PTS-AIC require further consideration. For PSO PTS-AIC, updating location and velocity variables in (8), (9) requires 5 multiplications, 5 additions/subtraction, and 1 comparison in quantization, per dimension. For DPSO PTS-AIC, updating velocity variables in (8) requires 5 multiplications, 4 additions, and updating location variables requires computing the sigmoid function and generating a random binary variable, per dimension. For PSO/DPSO, two more comparisons are needed for updating each particle's best value and the group's best value per particle per iteration. The complexity of PTS-AIC and PSO/DPSO PTS-AIC is summarized in Table 1.

5. Simulation Results and Discussion

Random QPSK symbols are generated to form OFDM data blocks. Each block contains 128 symbols corresponding to the 128-point FFT/IFFT. The subcarrier indices from the 85th to 87th are assumed to be the interference tones. From the AIC algorithm, the 84th to 88th subcarriers are removed and replaced by the AIC tones. The frequency-domain upsampling factor is four. To compute CCDF of the remaining interference power, I, 10,000 data blocks are simulated in each case.

The parameters of PSO are set as $c_0 = 1$, $c_1 = 2$, and $c_2 = 2$ resulting in equal influence from its own particle and from the group. This choice of parameters was recommended in an early work on PSO [15].

Figure 4 compares CCDF of I among PSO, DPSO, random and optimal PTS-AIC. Adjacent subblock partitioning is assumed. The parameters for PTS-AIC are $N_b = 16, N_w = 2$, and the parameters for PSO are $P = 30, Q = 20$. Optimal PTS-AIC exhaustively searches for the best weighting factors from the whole solution space of $2^{16} = 65536$ candidates. The optimal PTS-AIC serves as the best performance limit. The performance of the random scheme, which randomly chooses weighting factors and selects the best one, is plotted to illustrate the benefit of using the PSO/DPSO algorithms. For the random scheme, 600 random weighting factors are compared. This amount is equivalent to the number of comparisons in PSO/DPSO ($P \cdot Q$).

From the plots, the PSO and DPSO outperform the random scheme, and the DPSO can approach the performance of the optimal PTS-AIC at high interference power. At low interference power, however, the gain of PSO over the random scheme is small. Therefore, DPSO will be applied to the PTS-AIC for the rest of this section. The complexity of DPSO PTS-AIC is only about one percent ($30 \cdot 20/2^{16}$) of the optimal PTS-AIC in terms of the number of comparisons. Table 2 shows the average CPU time per realization of the considered algorithms taking into account all the required operations. It is shown that the PSO/DPSO has lower CPU times than the optimal scheme by almost two orders of magnitude.

To gain further insight on the complexity, Figure 5 shows the plots of average CPU time for optimal PTS-AIC, DPSO, and PSO. The CPU time of the optimal PTS-AIC is plotted versus the number of subblocks while the CPU times of

```
(1) Compute a probability mass function from $S(v) = 1/(1 + e^{-v})$
    where $v$ is the updated velocity for each dimension.
(2) Update the location by generating a random binary vector
    where each component is generated from the distribution
    computed in Step (1).
```

Algorithm 2: Location update for DPSO.

Table 1: Complexity of optimal, PSO/DPSO PTS-AIC.

Technique	No. of computing/comparison of (7)	Updating variables (per particle per iteration)
Optimal	$(N_w)^{N_b}$ times	—
PSO	$P \cdot Q$ times	$(5\times, 5\pm, 1\text{cmp})N_b\log_2(N_w) + 2\text{cmp}$
DPSO	$P \cdot Q$ times	$(5\times, 4\pm, 1\text{cmp}, \text{sigmoid \& gen}) \cdot N_b\log_2(N_w) + 2\text{cmp}$

× is multiplication. ± is addition/subtraction. cmp is comparison. "sigmoid & gen" represents the operations required for computing a sigmoid function and generating a binary random variable.

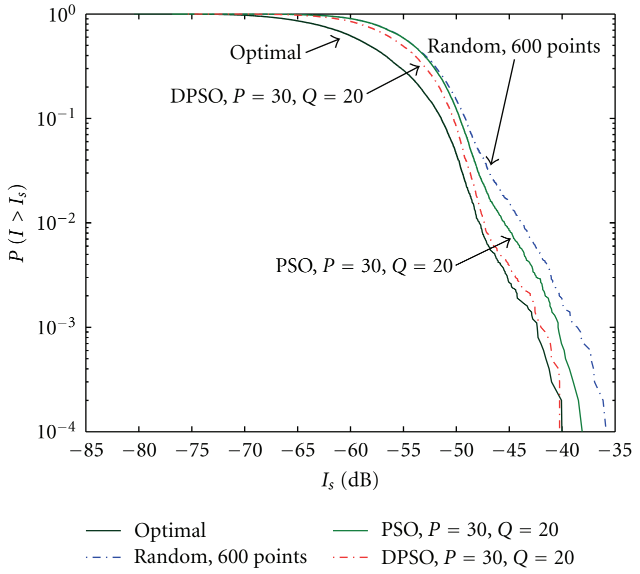


Figure 4: Comparisons of optimal, random and PSO/DPSO PTS-AIC. Adjacent subblock partitioning with $N_b = 16, N_w = 2$.

Table 2: CPU time of Optimal, random and PSO/DPSO PTS-AIC.

Technique	CPU time (ms)
Optimal	6542
Random	61.6
PSO	80.1
DPSO	85.9

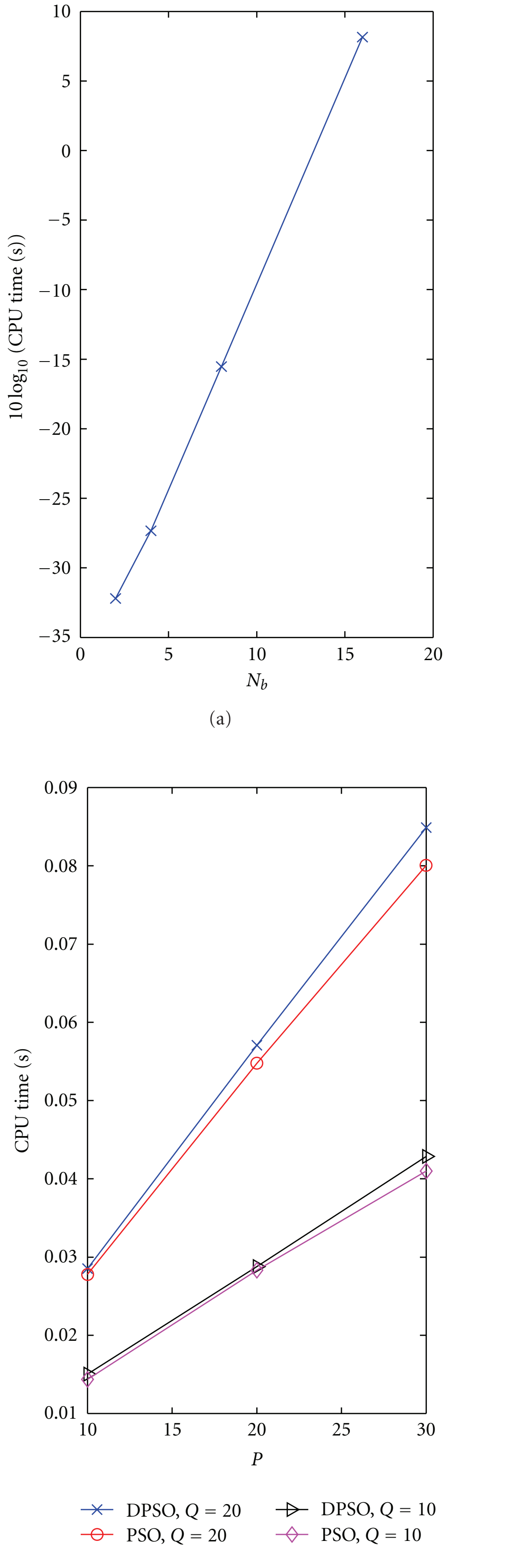
DPSO and PSO are plotted versus the number of particles. Note that the y-axis of Figure 5(a) is in a logarithmic scale. Since the curve is a straight line, it reflects an exponential complexity with respect to the number of subblocks in the case of the optimal PTS-AIC. In Figure 5(b), the CPU times of DPSO are slightly larger than those of PSO. From Table 1, given Q, the complexity is linear with P and vice versa. This is reflected in the results in Figure 5(b). For a fixed Q, the curves are linear in P. When Q is double, the slopes of the curves are also double.

Figure 6 shows the effect of different number of particles, P, to the performance of DPSO PTS-AIC. By increasing the number of particles, the performance is improved regardless of the interference power. The performance of interleaved subblock partitioning is more sensitive to the number of particles than that of adjacent subblock partitioning. Nevertheless, increasing the number of particles beyond 30 seem does not to achieve much further gain. Therefore, we limit the maximum number of particles at 30 in the following simulations.

Figure 7 shows the average interference power of DPSO PTS-AIC as a function of the number of iterations, Q. The average interference power decreases sharply at small Q's while it decreases gradually at large Q's. The Q value where the average interference power starts to decrease gradually depends on N_b. When N_b is larger, this value gets larger, since the solution space becomes larger, so it takes more iterations to find a near-optimal solution. Increasing the number of iterations beyond 20 does not significantly improve the performance in our considered cases. Therefore, we limit the maximum number of iterations at 20 in the following simulations.

Figure 8 illustrates I for 100 realizations of the optimal PTS-AIC and DPSO PTS-AIC with $P = 20, 30$ performing on the same data block for each realization. It is clear that DPSO PTS-AIC does not outperform the optimal PTS-AIC. Since DPSO PTS-AIC is suboptimal, the $P = 20$ case can sometimes outperform the $P = 30$ case (the particles are newly generated for $P = 20, 30$, so they are different from each other).

Figure 9 shows the CCDF of I when $N_b = 8, N_w = 4$. The DPSO PTS-AIC with $P = 30, Q = 20$ approaches the optimal PTS-AIC performance in the case of adjacent subblock partitioning. However, in the case of interleaved subblock partitioning, the DPSO PTS-AIC is still much worse than the optimal PTS-AIC.



(a)

(b)

Figure 5: CPU time of the algorithms (a) Optimal PTS-AIC (b) DPSO and PSO, $Q = 10, 20$.

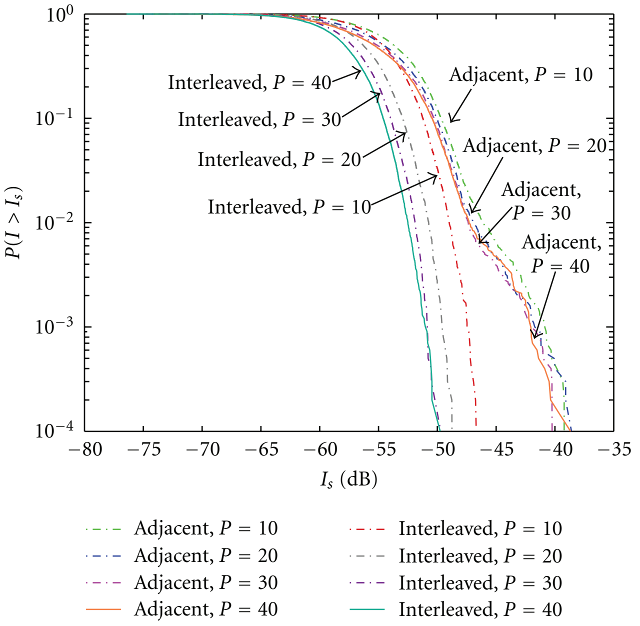


Figure 6: DPSO PTS-AIC with different number of particles, P. $N_b = 16, N_w = 2$. $Q = 20$.

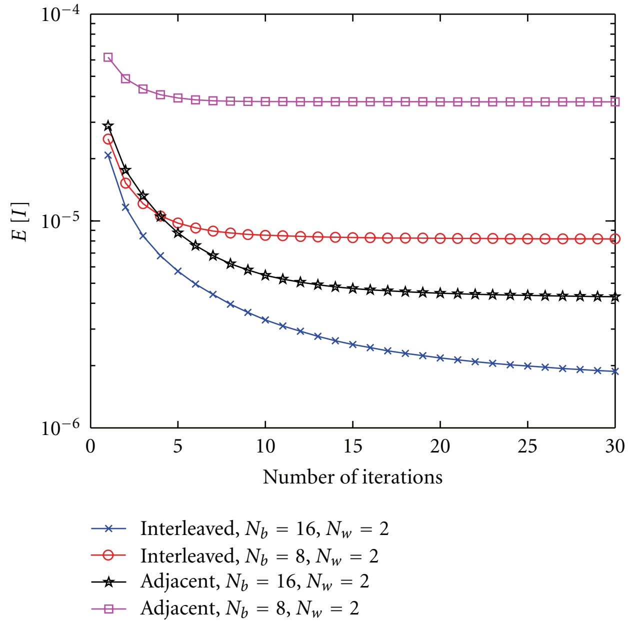


Figure 7: DPSO PTS-AIC with different number of iterations, Q. $P = 30$.

Figure 10 shows the CCDF of I when $N_b = 4, N_w = 8$. The DPSO PTS-AIC with $P = 30, Q = 20$ approaches the optimal PTS-AIC performance in both cases of adjacent and interleaved subblock partitionings.

Another potential benefit of PSO is apparent when N_b is very large: the optimal PTS-AIC becomes prohibitive. Figure 11 shows that, at 0.1%—excess interference power, 5-

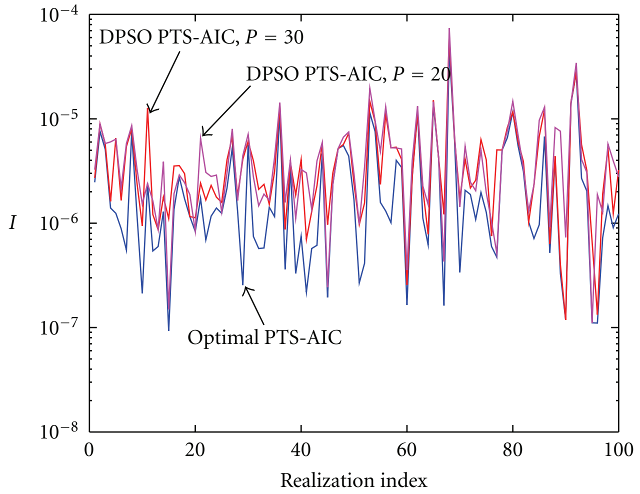


Figure 8: Optimal PTS-AIC and DPSO PTS-AIC with different number of particles, P. $N_b = 16, N_w = 2$. $Q = 20$.

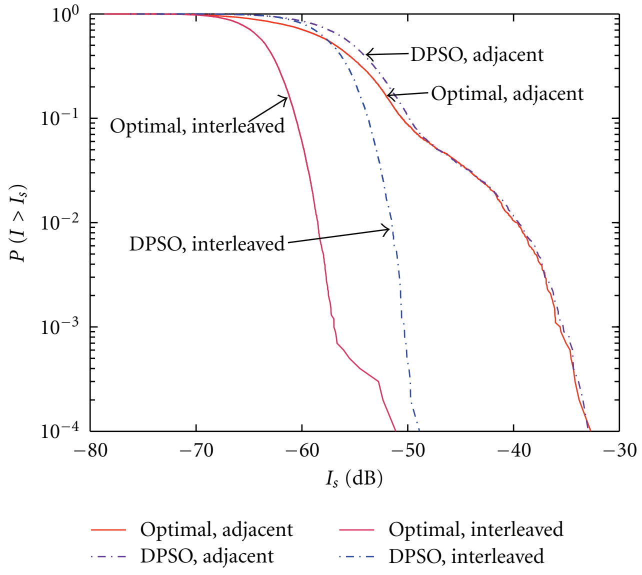


Figure 9: Optimal PTS-AIC and DPSO PTS-AIC, $N_b = 8, N_w = 4$. $P = 30, Q = 20$.

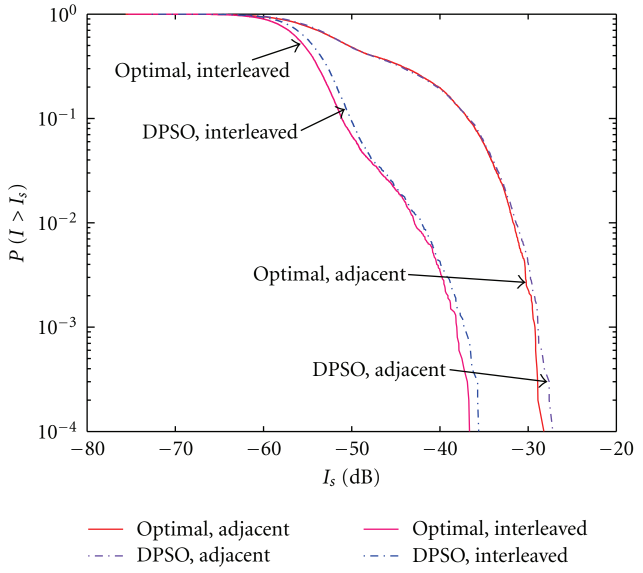


Figure 10: Optimal PTS-AIC and DPSO PTS-AIC $N_b = 4, N_w = 8$. $P = 30, Q = 20$.

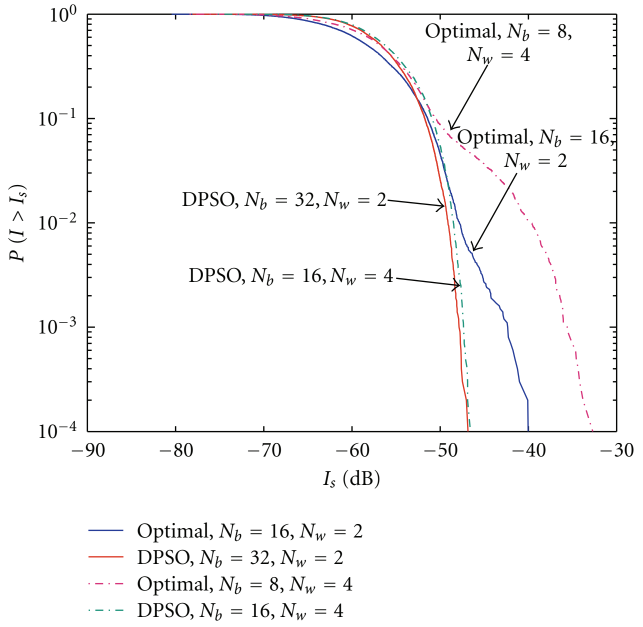


Figure 11: DPSO PTS-AIC with a larger N_b than that of the optimal PTS-AIC.

dB gain over the optimal PTS-AIC with $N_b = 16, N_w = 2$ is achieved by the DPSO with $N_b = 32, N_w = 2$. Similarly, about 10-dB gain over the optimal PTS-AIC with $N_b = 8, N_w = 4$ is achieved by DPSO with $N_b = 16, N_w = 4$. This outcome is attained by the DPSO PTS-AIC with the complexity only about 1% of the optimal PTS-AIC.

6. Conclusion

We propose a suboptimal algorithm, called particle swarm optimization (PSO), for partial transmit sequence active interference cancellation (PTS-AIC) used for interference avoidance feature for UWB OFDM transmitter, to reduce the computational complexity significantly. The PTS-AIC with PSO becomes more attractive when the number of subblocks and/or the constellation set for the weighting factors are large for PTS-AIC. The discrete version of PSO is a better choice for the suboptimal algorithm compared to the continuous version and is able to approach the performance of the optimal PTS-AIC in many cases at much lower complexity. The benefit that is, brought from PSO to PTS-AIC becomes more attractive when the number of subblocks for PTS-AIC is large and makes PTS-AIC implementable in hardware.

References

[1] S. Niranjayan, A. Nallanathan, and B. Kannan, "Modeling of multiple access interference and BER derivation for TH and DS UWB multiple access systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2794–2804, 2006.

[2] C. Snow, L. Lampe, and R. Schober, "Analysis of the impact of WiMAX-OFDM interference on multiband OFDM," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB '07)*, pp. 761–766, Singapore, September 2007.

[3] A. Nasri, R. Schober, and L. Lampe, "Analysis of narrowband communication systems impaired by MB-OFDM UWB interference," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4090–4100, 2007.

[4] K. Shi, Y. Zhou, B. Kelleci, T. W. Fischer, E. Serpedin, and A. I. Kaŗilayan, "Impacts of narrowband interference on OFDM-UWB receivers: analysis and mitigation," *IEEE Transactions on Signal Processing*, vol. 55, no. 3, pp. 1118–1128, 2007.

[5] S. M. Mishra, R. W. Brodersen, S. ten Brink, and R. Mahadevappa, "Detect and avoid: an ultra-wideband/WiMAX coexistence mechanism," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 68–75, 2007.

[6] ECMA International, "High rate ultra wideband PHY and MAC standard," ECMA-368, 2005.

[7] H. Yamaguchi, "Active interference cancellation technique for MB-OFDM cognitive radio," in *Proceedings of the 34th European Microwave Conference*, pp. 1105–1108, October 2004.

[8] Y. Wang and J. Coon, "Active interference cancellation for systems with antenna selection," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 3785–3789, May 2008.

[9] S.-G. Huang and C.-H. Hwang, "Low complexity active interference cancellation for OFDM cognitive radios," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 1279–1283, April 2008.

[10] Z. Wang, D. Qu, T. Jiang, and Y. He, "Spectral sculpting for OFDM based opportunistic spectrum access by extended active interference cancellation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 4442–4446, December 2008.

[11] P. Tarasak, F. Chin, Z. Lin, and X. Peng, "Further enhancement for active interference cancellation on MB-OFDM UWB transmission," in *Proceedings of the 68th Semi-Annual IEEE Vehicular Technology Conference (VTC '08)*, pp. 1–5, Calgary, BC, Canada, September 2008.

[12] P. Tarasak, Z. Lin, X. Peng, and F. Chin, "Partial transmit sequenceactive interference cancellation for UWB OFDM transmission," in *Proceedings of the IEEE Personal, Indoor and Mobile Radio Communications Symposium (PIMRC '09)*, 2009.

[13] S. H. Müller and J. B. Huber, "OFDM with reduced peak-to-average power ratio by optimum combination of partial transmit sequences," *IEE Electronics Letters*, vol. 33, no. 5, pp. 368–369, 1997.

[14] L. J. Cimini Jr. and N. R. Sollenberger, "Peak-to-average power ratio reduction of an OFDM signal using partial transmit sequences," *IEEE Communications Letters*, vol. 4, no. 3, pp. 86–88, 2000.

[15] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 81–86, May 2001.

[16] J. Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4104–4108, October 1997.

[17] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Mateo, Calif, USA, 2001.

[18] J. Kennedy and W. M. Spears, "Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, pp. 78–83, Anchorage, Alaska, USA, 1998.

[19] Z.-S. Lu and S. Yan, "Multiuser detector based on particle swarm algorithm," in *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies (CSSET '04)*, pp. 783–786, June 2004.

[20] Z.-Q. Guo, Y. Xiao, and M. H. Lee, "Multiuser detection based on particle swarm optimization algorithm over multipath fading channels," *IEICE Transactions on Communications*, vol. E90-B, no. 2, pp. 421–424, 2007.

[21] H. Liu and J. Li, "A particle swarm optimization-based multiuser detection for receive-diversity-aided STBC systems," *IEEE Signal Processing Letters*, vol. 15, pp. 29–32, 2008.

[22] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, "Optimization of power allocation for interference cancellation with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 128–150, 2009.

[23] H.-L. Hung, J.-H. Wen, S.-H. Lee, and Y.-F. Huang, "A suboptimal PTS algorithm based on particle swarm optimization technique for PAPR reduction in OFDM systems," *Eurasip Journal on Wireless Communications and Networking*, vol. 2008, Article ID 601346, 8 pages, 2008.

[24] S. Maheswararajah, S. K. Halgamuge, and M. Premaratne, "Sensor scheduling for target tracking by suboptimal algorithms," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1467–1479, 2009.

[25] F. Shu, W. Gang, and L. Shao-Qian, "Optimal multiuser MIMO linear precoding with LMMSE receiver," *Eurasip Journal on Wireless Communications and Networking*, vol. 2009, Article ID 197682, 10 pages, 2009.

[26] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, Md, USA, 1983.