

Sotirios K. Goudos*

A novel generalized oppositional biogeography-based optimization algorithm: application to peak to average power ratio reduction in OFDM systems

DOI 10.1515/math-2016-0066

Received March 31, 2016; accepted September 9, 2016.

Abstract: A major drawback of orthogonal frequency division multiplexing (OFDM) signals is the high value of peak to average power ratio (PAPR). Partial transmit sequences (PTS) is a popular PAPR reduction method with good PAPR reduction performance, but its search complexity is high. In this paper, in order to reduce PTS search complexity we propose a new technique based on biogeography-based optimization (BBO). More specifically, we present a new Generalized Oppositional Biogeography Based Optimization (GOBBO) algorithm which is enhanced with Oppositional Based Learning (OBL) techniques. We apply both the original BBO and the new Generalized Oppositional BBO (GOBBO) to the PTS problem. The GOBBO-PTS method is compared with other PTS schemes for PAPR reduction found in the literature. The simulation results show that GOBBO and BBO are in general highly efficient in producing significant PAPR reduction and reducing the PTS search complexity.

Keywords: Evolutionary algorithms, Biogeography-based optimization (BBO), Opposition based Learning, Combinatorial optimization, OFDM, PAPR, PTS

MSC: 94A14, 94A12, 68Q32, 68T05, 68W40

1 Introduction

Orthogonal frequency division multiplexing (OFDM) is widely used in several high-bit-rate digital communication systems such as Digital Audio Broadcasting (DAB), Digital Video broadcasting (DVB) and wireless local area networks [1, 2]. OFDM systems still have several research challenging issues. A major drawback of OFDM signals is the high value of peak to average power ratio (PAPR). The OFDM receiver detection efficiency is sensitive to non-linear devices like the High Power Amplifier (HPA) [3]. Thus, it is important to reduce the PAPR of OFDM signal in order to fully utilize the OFDM technical features.

Partial transmit sequences (PTS) [4], is a popular PAPR reduction method with good PAPR reduction performance. However, PTS requires an exhaustive search in order to find the optimal phase factors. Thus, the search complexity is high. Several methods have been published in the literature for PAPR reduction using PTS with low search complexity [4–33]. The Evolutionary Algorithms (EAs) mimic biological entities behaviour and they are inspired from Darwinian evolution in nature. The EAs have been extensively studied and applied to several problems in wireless communications [34] and to the PAPR reduction problem. These EAs among others include Genetic Algorithms (GAs) [12, 29–32], Particle Swarm Optimization (PSO) [9, 25], Differential Evolution (DE)

*Corresponding Author: Sotirios K. Goudos: Radiocommunications Laboratory, Section of Applied and Environmental Physics, Department of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece, E-mail: sgoudo@physics.auth.gr

[24, 33], Artificial Bee Colony (ABC) optimization [22], and parallel Tabu Search [18]. In [6] a quantum-inspired evolutionary algorithm (QEA) is proposed using PTS for PAPR reduction. An adaptive clipping control combined with a GA is applied in [35]. A parallel artificial bee colony (P-ABC) using selected mapping (SLM) is applied to the PAPR reduction problem in [36]. The authors in [37] propose a modified chaos clonal shuffled frog leaping algorithm (MCCSFLA) using PTS for PAPR reduction. In addition, other stochastic optimization techniques that have been applied to the PAPR reduction problem include Simulated Annealing (SA) [15], the Cross Entropy (CE) method [23], and Electromagnetism-like (EM) mechanism [13]. A comprehensive survey of PAPR reduction techniques for OFDM signals is given in [38].

In this paper, we propose a new PTS scheme based on biogeography-based optimization (BBO) [39]. BBO is a recently introduced evolutionary algorithm. BBO is based on mathematical models that describe how species migrate from one island to another, how new species arise, and how species become extinct. The way the problem solution is found is analogous to nature's way of distributing species. Learning from nature is the main motivation for the engineer to apply BBO to real world optimization problems [40–42]. In [43] a new BBO algorithm based on opposition-based learning (OBL) called Oppositional Biogeography-Based Optimization (OBBO) was introduced. The basic idea of the OBL concept is to calculate the fitness not only of the current individual but also to calculate the fitness of the opposite individual. The benefits of using such a technique are that convergence speed may be faster and that a better approximation of the global optimum can be found. OBL techniques were also applied successfully to Differential Evolution in [44]. In all the above papers, OBL was applied to continuous domain problems. In [45] the OBBO concept was applied to specific discrete domain problems like the traveling salesman (TSP) and the vertex coloring problem. However, in the above paper the definition of the opposite point was problem-dependent. In this paper, we propose a new Generalized Oppositional BBO (GOBBO) that can be applied to the PTS problem and to other discrete domain problems as well. The basic concept of the proposed algorithm is to decide, using a predefined opposition probability, if each decision variable in every D -dimensional individual is replaced by its opposite or not.

We compare the GOBBO-PTS scheme PAPR reduction performance with several techniques found in the literature. The simulation results show that the new GOBBO-PTS scheme achieves better performance than the above-mentioned techniques. To the best of the author's knowledge, this is the first time that BBO in general is applied to the PTS problem.

This paper is organized as follows: In Section 2 we describe the PTS algorithms details. Section 3 presents the problem description. A parameter setting study for GOBBO is shown in Section 4, while Section 5 has the numerical results and the statistical tests. Finally, we give the conclusion in Section 6.

2 Algorithms description

In this section we briefly describe different algorithms and methods used for PTS PAPR reduction. These can be classified into two major categories: heuristics and metaheuristics. The heuristics refer to problem specific methods that are applicable to the PTS problem only, while the metaheuristics are global optimizers that are problem-independent, and thus can be applied to a variety of problems. EAs are metaheuristics.

2.1 Heuristic PTS methods

Among others, these include the iterative flipping algorithm for PTS (IPTS) [4] and the gradient descent method (GD) [10]. In the iterative flipping algorithm [4], each input data block is divided into M subblocks to form partial transmit sequences as in the ordinary PTS technique. We assume that $b_m = 1$ for all m and compute the PAPR of the combined signal. Then invert the first phase factor to $b_1 = -1$ and recompute the resulting PAPR. If it is lower than the previous value then keep this value for b_1 , otherwise change b_1 back to its initial value. The algorithm continues that way until all other phase factors have been explored. The name “flipping” came from the fact that flipping the signs of the phase factors occurs. The search complexity of this technique is proportional to $(M - 1)W$.

The gradient descent method (GD) starts with a pre-determined vector of phase factors. Next, it finds an updated vector of phase factors in its “neighborhood” that results in the largest reduction in PAPR. A Neighborhood of radius r is defined as the set of vectors with Hamming distance equal to or lesser than r from its origin. The performance and complexity of the technique is dependent on the value of r . The search complexity is given by $C_{M-1}^r W^r I$, where C_{M-1}^r is the binomial coefficient defined by $C_{M-1}^r = \binom{M-1}{r}$

2.2 ABC-PTS

Artificial Bee Colony (ABC) [46] is a Swarm Intelligence (SI) algorithm, which has been applied to several real-world engineering problems. The ABC algorithm models and simulates the honey bee behavior in food foraging. In ABC algorithm, a potential solution to the optimization problem is represented by the position of a food source while the nectar amount of a food source corresponds to the quality (objective function fitness) of the associated solution. In order to find the best solution the algorithm defines three classes of bees: employed bees, onlooker bees and scout bees. The employed bee searches for the food sources, the onlooker bee makes a decision to choose the food sources by sharing the information of employed bee, and the scout bee is used to determine a new food source if a food source is abandoned by the employed bee and onlooker bee. For each food source there exists only one employed bee (i.e. the number of the employed bees is equal to the number of solutions). The employed bees search for new neighbor food source near to their hive.

2.3 ACO-PTS

Ant colony optimization (ACO) [47, 48] is a meta-heuristic inspired by the ants’ foraging behavior. At the core of this behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. Ants can sense pheromone. When they decide to follow a path, they tend to choose the one with strong pheromone intensities way back to the nest or to the food source. Therefore, shorter paths would accumulate more pheromone than longer ones. This feature of real ant colonies is exploited in ACO algorithms in order to solve combinatorial optimization problems considered to be NP -Hard.

2.4 PSO-PTS

PSO is an evolutionary algorithm that mimics the swarm behavior of bird flocking and fish schooling [49]. The most common PSO algorithms include the classical Inertia Weight PSO (IWPSO) and Constriction Factor PSO (CFPSO) [50]. PSO is an easy to implement algorithm with computational efficiency. In PSO, the particles move in the search space, where each particle position is updated by two optimum values. The first one is the best solution (fitness) that has been achieved so far. This value is called $pbest$. The other one is the global best value obtained so far by any particle in the swarm. This best value is called $gbest$. After finding the $pbest$ and $gbest$, the most commonly used velocity update rule of each particle for every problem dimension is given by :

$$u_{G+1,ni} = wu_{G,ni} + c_1 rand_{1(0,1)}(pbest_{G+1,ni} - x_{G,ni}) + c_2 rand_{2(0,1)}(gbest_{G+1,ni} - x_{G,ni}) \quad (1)$$

where $u_{G+1,ni}$ is the i -th particle velocity in the n -th dimension, $G + 1$ denotes the current iteration and G the previous, $x_{G,ni}$ is the particle position in the n -th dimension, $rand_{1(0,1)}, rand_{2(0,1)}$ are uniformly distributed random numbers in $(0,1)$, w is a parameter known as the inertia weight, and c_1 and c_2 are the learning factors.

2.5 MSFLA-PTS

The authors in [51], introduced the shuffled frog leaping algorithm (SFLA), which is inspired by the natural behavior of the frog. SFLA uses a population-based cooperative search metaphor inspired by natural memetics. The basic idea

in SFLA is to divide the population into different frog groups. Each group consists of a fixed frog number. In SFLA, the information is carried by a meme, groups of memes are called meme complexes, or "memeplexes". The authors in [37] propose a modified shuffled frog leaping algorithm (MSFLA) using PTS for PAPR reduction.

2.6 BGA-PTS

In a binary-coded GA (BGA) each chromosome encodes a binary string [52]. The most commonly used operators are crossover, mutation, and selection. The selection operator selects two parent chromosomes from the current population according to a selection strategy. The crossover operator combines the two parent chromosomes in order to produce one new child chromosome. The mutation operator is applied with a predefined mutation probability to a new child chromosome. We have selected for this paper the BGA with the same features used in [35].

2.7 The BBO-PTS algorithm

The mathematical models of Biogeography are based on the work of Robert MacArthur and Edward Wilson in the early 1960s. Using this model, they have been able to predict the number of species in a habitat. The habitat is an area that is geographically isolated from other habitats. The geographical areas that are well suited as residences for biological species are said to have a high habitat suitability index (HSI). Therefore, every habitat is characterized by the HSI which depends on factors like rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. Each of the features that characterize habitability is known as suitability index variables (SIV). The SIVs are the independent variables while HSI is the dependent variable.

In BBO-PTS a solution to the M -dimensional problem can be represented as a vector of SIV variables $[SIV_1, SIV_2, \dots, SIV_M]^T$, which is a habitat or island. The SIV variables represent the phase vector $\mathbf{b} = [b_1, b_2, \dots, b_M]^T$. The value of HSI of a habitat is the value of the PTS objective function that corresponds to that solution and it is found by:

$$HSI = F(\text{habitat}) = F(SIV_1, SIV_2, \dots, SIV_M) = F(\mathbf{b}) \quad (2)$$

Habitats with a low PAPR value are good solutions of the objective function, while poor solutions are those habitats with high PAPR value. The Habitats with low PAPR are those that have large population and high emigration rate μ . For these habitats, the immigration rate λ is low. The poor solutions are those that have high PAPR, which means they have high immigration rate λ and low emigration rate μ . The immigration and emigration rates are functions of the number of species in the habitats. These are given by:

$$\mu_k = E \left(\frac{k}{S_{\max}} \right) \quad (3)$$

$$\lambda_k = I \left(1 - \frac{k}{S_{\max}} \right) \quad (4)$$

where I is the maximum possible immigration rate, E is the maximum possible emigration rate, k is the rank of the given candidate solution, and S_{\max} is the maximum number of species (e.g. population size). Therefore, the best candidate solution has a rank of S_{\max} and the worst candidate solution has a rank of one.

BBO-PTS uses both mutation and migration operators. The application of these operators to each SIV in each solution is decided probabilistically. For each generation, there is a probability $P_{\text{mod}} \in [0, 1]$ that each candidate solution will be modified by migration. P_{mod} is a user-defined parameter that is typically set to a value close to one, and is analogous to crossover probability in GAs. The migration for NP habitats can be described in Algorithm 1.

The X_i in the above algorithm is habitat i . The information sharing between habitats is accomplished using the immigration and emigration rate. The λ_i is proportional to the probability that SIVs from neighbouring habitats will migrate into habitat X_i . The μ_i is proportional to the probability that SIVs from habitat X_i will migrate into neighboring habitats. The mutation rate m of a solution S is defined to be inversely proportional to the solution

probability and it is given by:

$$m_S = m_{\max} \left(\frac{1 - P_S}{P_{\max}} \right) \quad (5)$$

where P_S is the probability that a habitat contains S species, P_{\max} is the maximum P_S value over all $s \in [1, S_{\max}]$, and m_{\max} is a user-defined parameter. Simon in [35] described how P_S changes from time t to time $t + \Delta t$ as:

$$P_S(t + \Delta t) = P_S(t) (1 - \lambda_S \Delta t - \mu_S \Delta t) + P_{S-1}(t) \lambda_{S-1} \Delta t + P_{S+1}(t) \mu_{S+1} \Delta t. \quad (6)$$

Algorithm 1 BBO migration

```

1: for  $i=1$  to  $NP$  do
2:   Select  $X_i$  with probability based on  $\lambda_i$ 
3:   if  $rnd(0, 1) < \lambda_i$  then
4:     for  $j=1$  to  $NP$  do
5:       Select  $X_j$  with probability based on  $\mu_j$ 
6:       if  $rnd(0, 1) < \mu_j$  then
7:         Randomly select a SIV  $\sigma$  from  $X_j$ 
8:         Replace a random SIV in  $X_i$  with  $\sigma$ 
9:       end if
10:    end for
11:  end if
12: end for

```

In this paper, we use the binary mutation operator suggested in [53], which uses the inverse operation to update the habitat. The binary Mutation procedure is described in Algorithm 2.

Algorithm 2 BBO binary mutation

```

1: for  $i=1$  to  $NP$  do
2:   Compute the probability  $P_i$ 
3:   Select SIV  $X_i(j)$  with probability based on  $P_i$ 
4:   if  $rnd(0, 1) < m_i$  then
5:     Replace  $X_i(j)$  with  $1 - X_i(j)$  to generate a new SIV
6:   end if
7: end for

```

The m_i in Algorithm 2 is the mutation rate of solution i . As with other evolutionary algorithms, BBO also incorporates elitism. This is implemented with a user-selected elitism parameter p . This means that the p best phase vectors remain from one generation to the other.

2.8 Opposition Based Learning (OBL)

The basic concept of OBL was originally introduced by Tizhoosh in [54]. The basic idea of OBL is to calculate the fitness not only of the current individual but also to calculate the fitness of the opposite individual. Then the algorithm selects the individual with the lower (higher) fitness value. At first we give the definitions for the basic concepts of OBL [54–56].

Definition (Opposite Number) let $x \in [a, b]$ be any real number. The opposite number is defined by

$$x_O = a + b - x \quad (7)$$

Definition (Opposite Point). Similarly if we extend the above definition to D -dimensional space then let $P(x_1, x_2, \dots, x_D)$ be a point where $x_1, x_2, \dots, x_D \in \mathfrak{R}$ and $x_j \in [a_j, b_j] \forall j \in \{1, 2, \dots, D\}$. The opposite point $P_O(x_{O1}, x_{O2}, \dots, x_{OD})$ is defined by its components

$$x_{Oj} = a_j + b_j - x_j \quad (8)$$

Definition (Semi-opposite Point) [57]. If we change the components of a point by its opposites only in some components and the other remain unchanged then the new point is a semi-opposite point. This is defined by

$$P_{SO}(x_{SO1}, x_{SO2}, \dots, x_{SOj}, \dots, x_{SOD}) \text{ where } \forall j \in \{1, 2, \dots, D\} x_{SOj} = \{x_j \text{ or } x_{Oj} \quad (9)$$

For example in a two-dimensional space where each dimension can be either 0 or 1 we consider the point $P1(0, 1)$. Then the two semi-opposite points are $P2(0, 0)$ and $P3(1, 1)$, while the opposite point is $P4(1, 0)$.

2.9 Proposed algorithm

In this paper we propose a OBBO version based on semi-opposite points. We call this algorithm Generalized OBBO (GOBBO). We define a new control parameter named opposition probability $p_o \in [0, 1]$. This parameter controls if a SIV variable in a habitat will be replaced by its opposite or not. Moreover, as in previous opposition-based algorithms [43–45] we use the jumping rate parameter $j_r \in [0, 1]$ which controls in each generation if the opposite population is created or not. The opposite based algorithms require two additional parts to the original algorithm code; the opposition-based population initialization and the opposition-based generation jumping [43–45]. The opposition based population initialization for GOBBO is described in Algorithm 3. For this case $low_j, upper_j$ are the lower and upper limits in the j -th dimension respectively.

Algorithm 3 Opposition-based population initialization

- 1: Generate uniform distributed random population P
 - 2: **for** $i=1$ to NP **do**
 - 3: Generate semi-opposite population OP_s
 - 4: **for** $j=1$ to D **do**
 - 5: **if** $rnd[0, 1] < p_o$ **then**
 - 6: $x_{osi,j} = low_j + upper_j - x_{i,j}$
 - 7: **else**
 - 8: $x_{osi,j} = x_{i,j}$
 - 9: **end if**
 - 10: **end for**
 - 11: **end for**
 - 12: Initial population= the fittest among P and OP_s
-

The opposition-based generation jumping follows a similar approach. The algorithm description is given in Algorithm 4. The min_j, max_j are the minimum and maximum values of the j -th dimension in the current population respectively.

The GOBBO algorithm for PAPR reduction is outlined below:

- 1) Initialize the GOBBO control parameters. Map the problem solutions to phase vectors and habitats. Set the habitat modification probability P_{mod} , the maximum immigration rate I , the maximum emigration rate E , the maximum migration rate m_{max} and the elitism parameter p (if elitism is desired). Set the jumping rate j_r and the opposition probability p_o .
- 2) Initialize a random population of NP habitats (phase vectors) from a uniform distribution. Set the number of generations G to one.
- 3) Initialize the opposite population.

Algorithm 4 Opposition-based generation jumping

```

1: if  $rnd[0, 1] < j_r$  then
2:   for  $i=1$  to  $NP$  do
3:     Generate semi-opposite population  $OP_s$ 
4:     for  $j=1$  to  $D$  do
5:       if  $rnd[0, 1] < p_o$  then
6:          $x_{osi,j} = \min_j + \max_j - x_{i,j}$ 
7:       else
8:          $x_{osi,j} = x_{i,j}$ 
9:       end if
10:    end for
11:  end for
12: end if
13: Select fittest among current population  $P$  and  $OP_s$ 

```

4) Map the PAPR value to the number of species S , the immigration rate λ_k , the emigration rate μ_k for each solution (phase vector) of the population.

5) Apply the migration operator for each non-elite habitat based on immigration and emigration rates using (3) and (4).

6) Update the species count probability.

7) Apply the mutation operator.

8) Evaluate objective function value.

8) If $rnd[0, 1] < j_r$ calculate the opposite population.

9) Sort the population according to the PAPR value from best to worst.

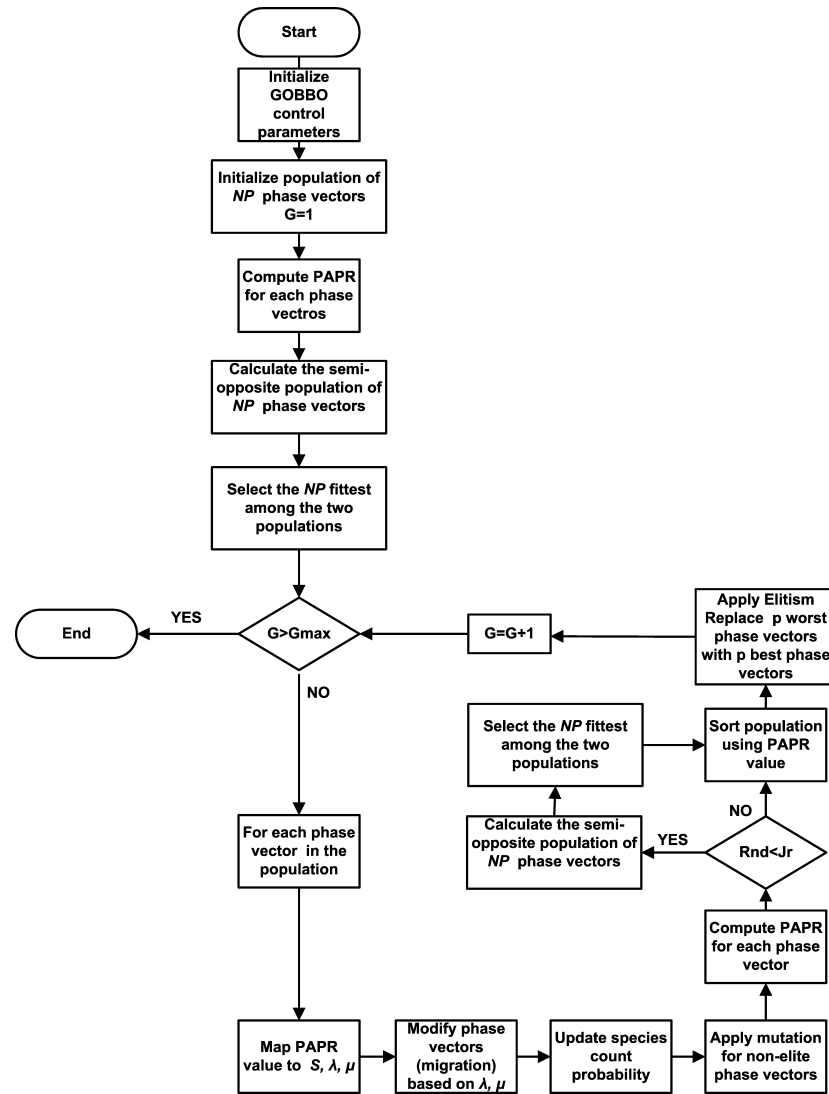
10) Apply elitism by replacing the p worst habitats of the previous generation with the p best ones.

11) Repeat step 3 until the maximum number of generations G_{\max} or the maximum number of objective function evaluations is reached.

A flowchart of the GOBBO algorithm is given in Fig. 1. The time complexity of the original BBO algorithm at each iteration is $\mathcal{O}(NP^2M + NPf)$, where f is the time complexity of the objective function and M is the problem dimensions. Sorting the population in algorithms 3 and 4 using the quick sort algorithm has time complexity $\mathcal{O}(NP^2)$. Algorithms 3 and 4 both have time complexity $\mathcal{O}(NP^2 + NPM + NPf)$. The time complexity of the GOBBO algorithm at each iteration is therefore $\mathcal{O}(NP^2(M + 1) + NPM + NPf)$, which reduces to $\mathcal{O}(NP^2(M + 1) + NPf)$.

Similarly to the other evolutionary algorithms (EAs), such as GAs, ABC, PSO, in the BBO approach there is a way of sharing information between solutions [39]. This feature makes BBO suitable for the same types of problems that the other algorithms are used for, namely high-dimensional data. Additionally, BBO has some unique features that are different from those found in the other evolutionary algorithms. For example, quite differently from GAs, Ant Colony Optimization (ACO) [48] and PSO, from one generation to the next the set of the BBO's solutions is maintained and improved using the migration model, where the emigration and immigration rates are determined by the fitness of each solution. BBO differs from PSO in the fact that PSO solutions do not change directly; the velocities change. The BBO solutions share directly their attributes using the migration models. The migration operator provides BBO with a good exploitation ability. These differences can make BBO outperform other algorithms [39, 58, 59]. It must be pointed out that if PSO or ABC are constrained to discrete space then the next generation will not necessarily be discrete [59]. However, this is not true for BBO; if BBO is constrained to a discrete space then the next generation will also be discrete to the same space. As the authors in [59] suggest, this indicates that BBO could perform better than other EAs on combinatorial optimization problems, which makes BBO suitable for application to the PTS problem. The main computational cost of EAs is in the evaluation of the objective function. The BBO mechanism is simple, like that of PSO and ABC. Therefore, for most problems, the computational cost of BBO and other EAs will be the same since it will be dominated by objective function evaluation [58]. More details about the BBO algorithm can be found in [39, 58, 59].

Fig. 1. GOBBO-PTS flowchart.



3 Problem description

In an OFDM system, the high-rate data stream is split into N low-rate data streams that are simultaneously transmitted using N subcarriers. The discrete-time signal of such a system is given by [1, 2]:

$$s_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} S_n e^{j2\pi nk/LN} \quad k = 0, 1, \dots, LN - 1 \quad (10)$$

where L is the oversampling factor, $\mathbf{S} = [S_0, S_1, \dots, S_{N-1}]^T$ is the input signal block. Each symbol is modulated by either phase-shift keying (PSK) or quadrature amplitude modulation (QAM). The PAPR of the signal in (10) is defined as the ratio of the maximum to average power and is expressed in dB as [1, 2]:

$$PAPR(s) = 10 \log_{10} \frac{\max_{0 \leq k \leq LN-1} |s_k|^2}{E[|s_k|^2]} \quad (11)$$

where $E[.]$ is the expected value operation. In the PTS approach the input data OFDM block is partitioned into M disjointed subblocks represented by the vector \mathbf{S}_m $m = 1, 2, \dots, M$ and oversampled by inserting $(L - 1)N$ zeros.

Then the PTS process is expressed as [4]:

$$\mathbf{S} = \sum_{m=1}^M \mathbf{S}_m \tag{12}$$

Next, the subblocks are converted to time domain using LN point inverse fast fourier transform (IFFT). The representation of the OFDM block in time domain is expressed by:

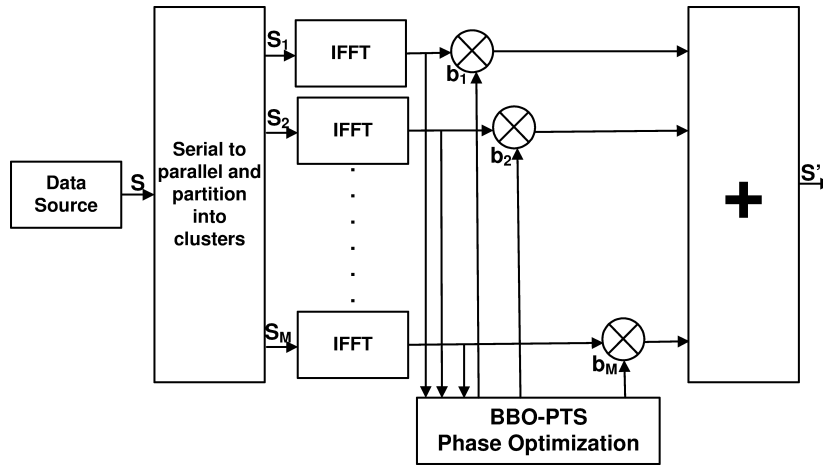
$$\mathbf{s} = IFFT \left\{ \sum_{m=1}^M \mathbf{S}_m \right\} = \sum_{m=1}^M IFFT \{ \mathbf{S}_m \} = \sum_{m=1}^M \mathbf{s}_m \tag{13}$$

The PTS objective is to produce a weighted combination of the M subblocks using $\mathbf{b} = [b_1, b_2, \dots, b_M]^T$ complex phase factors to minimize PAPR. The transmitted signal in time domain after this combination is given by [4]:

$$\mathbf{s}'(\mathbf{b}) = \sum_{m=1}^M b_m \mathbf{s}_m \tag{14}$$

The block diagram of the PTS technique is shown in Fig. 2.

Fig. 2. Block diagram of the BBO-PTS technique.



In order to reduce the search complexity the phase factor possible values are limited to a finite set. The set of allowable phase factors is [22]:

$$A = \left\{ e^{\frac{j2\pi n}{W}} \mid n = 0, 1, \dots, W - 1 \right\} \tag{15}$$

where W is the number of allowed phase factors. Therefore, in case of M subblocks and W phase factors the total number of possible combinations is W^M . In order to reduce the search complexity we usually set fixed one phase factor. The optimization goal of the PTS scheme is to find the optimum phase combination for minimum PAPR. Thus, the objective function can be expressed as [22]:

Minimize

$$F(\mathbf{b}) = 10 \log_{10} \frac{\max_{0 \leq k \leq LN-1} |s'(\mathbf{b})|^2}{E[|s'(\mathbf{b})|^2]} \tag{16}$$

subject to

$$\mathbf{b} \in \left\{ e^{j\phi_m} \right\}^M \text{ where } \phi_m \in \left\{ \frac{2\pi n}{W} \mid n = 0, 1, \dots, W - 1 \right\} \tag{17}$$

For $W = 2$ $\mathbf{b} \in \{-1, 1\}^M$. We can set a phase factor fixed without any performance loss. In that case, there are $M - 1$ decision variables to be optimized. Therefore, the search space size is 2^{M-1} . The search complexity increases exponentially with M .

4 Tuning control parameters

In order to explore the GOBBO sensitivity to control parameter selection we have evaluated the GOBBO-PTS method using different parameter settings. The GOBBO control parameters in all simulations are given below. The habitat modification probability, P_{mod} , is set to one, and the maximum mutation rate, m_{max} , is set to 0.005. The maximum immigration rate I , and the maximum emigration rate E are both set to one.

First, we evaluate the effect of the new parameter of the opposition probability P_o on GOBBO performance. We compare it with the Random Search (RS) [4] method by selecting 600 and 1200 random phase factors. We set the jumping rate constant at 0.3 for all cases. Table 1 holds the comparative results for this case. We notice that for $P_o = 0.3$ we obtain the best result of 6.30dB, which is lower than the RS method with 1200 search complexity. We also notice that the GOBBO performance is relatively robust regarding this parameter for values less or equal to 0.4. In case of $P_o = 1$ where the original OBBO algorithm is selected the results seem to deteriorate. Figs 3a and 3b depict the PARP reduction performance comparison among different opposition probability values. We notice the $P_o = 0.3$ results are better than the others.

Fig. 3. PARP reduction performance comparison of GOBBO-PTS algorithm with different opposition probability values.

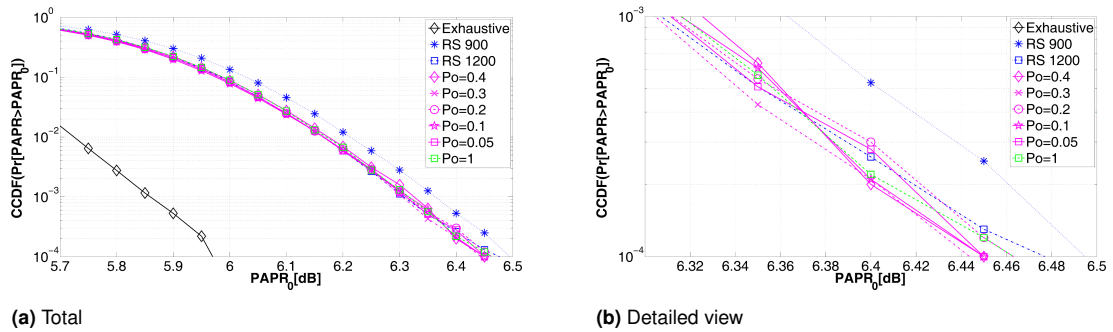


Table 1. Comparison of computational complexity for $CCDF = 10^{-3}$ among different opposition probability values. The smaller value is in bold font.

Opposition Probability	PAPR (dB)
P_o	
0.05	6.31
0.1	6.31
0.2	6.31
0.3	6.30
0.4	6.31
1.0	6.32

Next, we evaluate the GOBBO performance regarding the jumping rate. The suggested value for jumping rate found in the literature is 0.3 [44]. We will test if this value is suitable for GOBBO. We set for this case the opposition probability to 0.3 as it was found to be more suitable value. Table 2 reports the comparative results for different jumping rates. We notice that the best obtained values are for jumping rates 0.3 and 0.2. In all other cases the GOBBO performance is similar to that of the original BBO algorithm performance. Figs 4a and 4b present the PARP reduction performance using different jumping rates. It is obvious that the best performance is for jumping rates 0.3 and 0.2. Therefore, the above two cases have shown that for the PTS scheme the best control parameters values are 0.3 for both jumping rate and opposition probability.

Moreover, we compare the BBO-PTS and GOBBO-PTS performance using different values of population size and iterations. We also compare results with the iterative flipping algorithm for PTS (IPTS) [4], the gradient descent

Fig. 4. PARP reduction performance comparison of GOBBO-PTS algorithm with different jumping rate values.

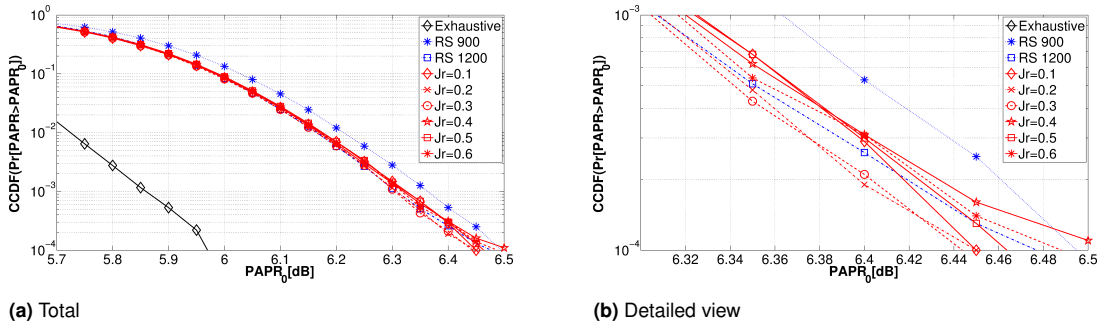


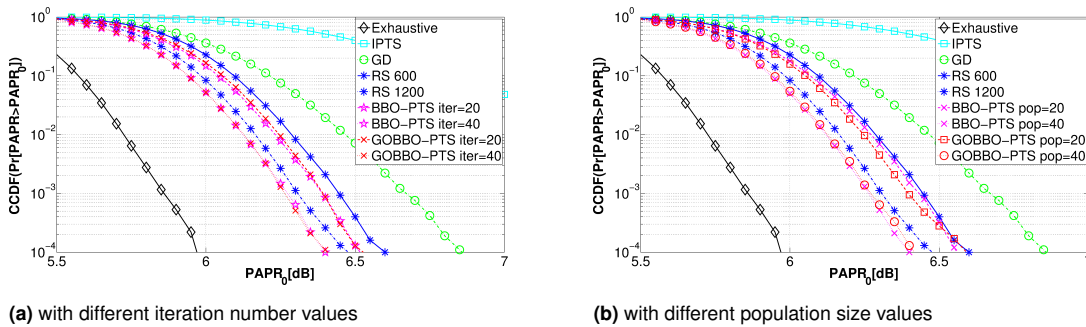
Table 2. Comparison of computational complexity for $CCDF = 10^{-3}$ among different jumping rate values. The smaller value is in bold font.

Jumping Rate	PAPR (dB)
j_r	
0.1	6.32
0.2	6.31
0.3	6.30
0.4	6.32
0.5	6.32
0.6	6.32

method (GD) [10] and the original BBO algorithm. For GOBBO the total number of objective function evaluations remains constant to $NP \times G$.

First, we evaluate performance using a constant population size of 30 for different iteration number values. The PAPR valued obtained by BBO-PTS when $Pr(PAPR > PAPR_0) = 10^{-3}$ for iteration number 20 and 40 is 6.39dB and 6.28dB, respectively. The PAPR by GOBBO-PTS with population size 30 and iteration number 20 and 40 is 6.38dB and 6.26dB, respectively. Using the RS method with 600 and 1200 random phase factors the PAPRs are reduced to 6.44dB and 6.31dB, respectively. We notice that BBO-PTS and GOBBO-PTS in all cases perform better than the RS with the same search complexity. GOBBO-PTS performs slightly better than the original BBO. The BBO-PTS and GOBBO PAPR reduction values are also very close to those of the RS method with larger search complexity. Fig. 5a depicts the PARP reduction performance comparison between the different schemes.

Fig. 5. PARP reduction performance comparison of GOBBO-PTS algorithm.



Next, we evaluate the performance with a constant iteration number of 30 for different population size values. The PAPR value found by BBO-PTS with $G = 30$ and population size 20 and 40 is 6.43dB and 6.27dB, respectively. For GOBBO-PTS the corresponding values are 6.39dB and 6.27dB for population size 20 and 40, respectively.

The PAPR reduction performance comparison for this case is shown in Fig. 5b. We notice that GOBBO-PTS clearly outperforms the original BBO for population size 20, while it performs about the same for population size 40. Again, we notice that both BBO algorithms perform better than the RS methods with the same search complexity. One may also notice that the performance of both BBO algorithms for a given number of objective function evaluations is slightly better for a larger number of iterations instead of a larger population size. Finally, Table 3 summarises the suggested GOBBO-PTS control parameters for solving the PAPR reduction problem.

Table 3. Suggested GOBBO control parameters

Maximum mutation probability (m_{\max})	0.005
Maximum immigration rate (I)	1
Maximum emigration rate (E)	1
Habitat modification probability (P_{mod})	1
Opposition probability (P_o)	0.3
Jumping rate (j_r)	0.3

5 Simulation results

We have evaluated the original BBO-PTS algorithm, the proposed GOBBO-PTS algorithm and the original OBBO-PTS performance by conducting several simulations. We have used two main measurement criteria namely the complementary cumulative distribution function (CCDF) and the computational complexity. In all our simulations, 10^5 random OFDM blocks are generated. The transmitted signal is oversampled by a factor $L = 4$. We consider 16-QAM modulation with $N = 256$ sub-carriers which are divided into $M = 16$ random subblocks. The phase factors for $W = 2$ are selected. We consider the first phase factor to be fixed so the total number of unknown phase factors is $M - 1$.

The BBO-PTS control parameters in all simulations are given below. For all BBO algorithms the habitat modification probability, P_{mod} , is set to one, and the maximum mutation rate, m_{\max} , is set to 0.005. The maximum immigration rate I , and the maximum emigration rate E are both set to one. The elitism parameter p is set to two. For GOBBO the jumping rate is set to 0.3 and the opposition probability is set to 0.3 also. In the ABC algorithm, the *limit* parameter is set to 5 as in [7]. In the PSO algorithm c_1 and c_2 are set to 2.05, while the inertia weight is linearly decreasing starting from 0.9 to 0.4. For ACO the initial pheromone value τ_0 is set to $1.0e-6$, the pheromone update constant Q is set to 20, the exploration constant q_0 is set to one, the global pheromone decay rate ρ_g is 0.9, the local pheromone decay rate ρ_l is 0.5, the pheromone sensitivity α is one, and the visibility sensitivity is β is five. The population size of MSFLA is also set to 30. The number of groups in MSFLA is set to 5, therefore each group consists of 6 frogs. For the BGA, the crossover probability is set to 0.9, and the mutation probability to 0.05, respectively.

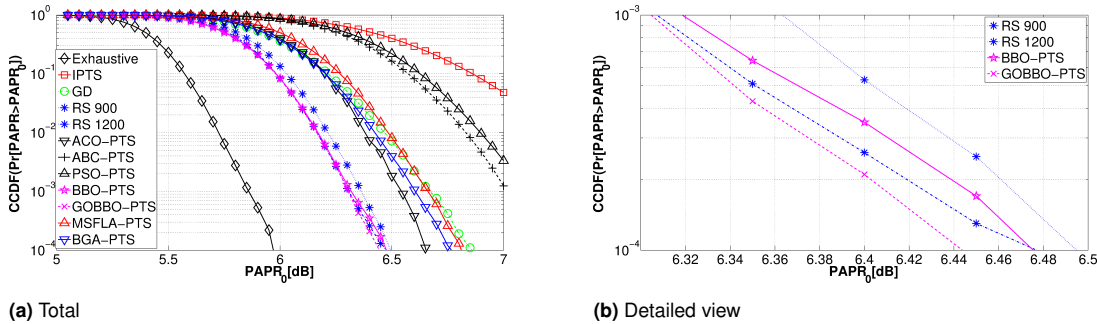
Figs 6a and 6b show the comparison between the CCDF by BBO-PTS and other PTS reduction techniques. For $\Pr(PAPR > PAPR_0) = 10^{-3}$ the PAPR of the original OFDM transmitted signal is 10.84dB. For all evolutionary algorithms, the population size NP is set to 30 and the maximum number of generations G is set to 30. Thus, the computational complexity of the three BBO-PTS schemes is $NP \times G = 900$. The PAPR by the BBO-PTS, GOBBO-PTS and OBBO-PTS ($P_o = 1$), is 6.32dB, 6.30dB and 6.32dB respectively. Hence, there is no apparent advantage in using the OBBO instead of the original BBO for this case. However, the use of semi-opposite points in the GOBBO algorithm produces better results. The computational complexity of the exhaustive search is $W^{M-1} = 32768$ while the PAPR for this case is 5.86dB. The PAPR by the iterative flipping algorithm for PTS (IPTS) [4] is 7.55dB with search complexity $(M - 1)W = 30$. The PAPR by the gradient descent method (GD) [10] with search complexity $C_{M-1}^r W^r I = C_{15}^2 2^2 3 = 1260$ is 6.96dB. We also use the Random Search (RS) [4] method by selecting 900 and

Table 4. Comparison of computational complexity for $CCDF = 10^{-3}$ among different PTS schemes.

Method	Computational Complexity	PAPR (dB)
Original	0	10.84
Exhaustive	$W^{M-1} = 32768$	5.86
IPTS	$(M - 1)W = 30$	7.55
GD	$C_{M-1}^r W^r I = C_{15}^2 2^2 3 = 1260$	6.96
RS	900	6.37
RS	1200	6.31
PSO-PTS	$NP \times G = 900$	7.13
ABC-PTS	$NP \times G = 900$	7.01
OBBO-PTS	$NP \times G = 900$	6.32
GOBBO-PTS	$NP \times G = 900$	6.30
BBO-PTS	$NP \times G = 900$	6.32
ACO-PTS	$NP \times G = 900$	6.52
MSFLA-PTS	$NP \times G = 900$	6.66
BGA-PTS	$NP \times G = 900$	6.61

1200 random phase factors. The PAPR for this case is 6.37dB and 6.31dB respectively. The PAPR by ABC-PTS [22], PSO-PTS [25], and ACO-PTS, is 7.01dB, 7.13dB, and 6.52dB, respectively. Moreover, the PAPR value by MSFLA-PTS [37] and, BGA-PTS [35] is 6.66dB, and 6.61dB, respectively. Table 4 holds the comparison of the search complexity among the different methods for $CCDF = 10^{-3}$, $NP= 30$, and $G=30$. It is obvious that GOBBO-PTS presents the better performance among the other methods with the same search complexity. The GOBBO-PTS obtains better PAPR value than the one of the RS method with 1200 random phase factors. Thus, the GOBBO-PTS using OBL concepts with 25% lower search complexity than the RS-1200 has obtained better results. In addition, the PAPR reduction by BBO-PTS with 900 search complexity is 0.01 dB larger to that of the RS with 1200 search complexity.

Fig. 6. PARP reduction performance comparison of the BBO-PTS algorithms with other PTS schemes for $NP = 30, G = 30$.



Additionally, the PAPR values for additional CCDF values are reported in Table 5. We notice that in most cases GOBBO-PTS outperforms the other algorithms. Moreover, non-parametric statistical tests were conducted in order to further elaborate the algorithms results. These are the Friedman test and the Wilcoxon signed-rank test, which are commonly used for the performance evaluation of EAs [60, 61]. Both tests are conducted based on the obtained PAPR values, which are summarized in Tables 4 and 5. The results are presented in Tables 6 and 7. GOBBO-PTS emerged as the best algorithm in the Friedman test, and the RS 1200 is second. The significant level set for the Wilcoxon signed-rank test was 0.05. From the Wilcoxon signed rank test, we see that most p -values are below the significance level (0.05), so we can accept that a significant difference exists between the different methods, with GOBBO-PTS to be significantly better than ACO-PTS, ABC-PTS, PSO-PTS, IPTS, GD, RS900, MSFLA-PTS, BGA-PTS, and OBBO-PTS. Although GOBBO-PTS is not significantly better than the original BBO-PTS and RS 1200, it outperforms them according to the results of average ranking.

Table 5. Comparison of PAPR values (dB) for different *CCDF* values among different PTS schemes

Method	CCDF values					
	1.00E-01	5.00E-02	1.00E-02	5.00E-03	5.00E-04	1.00E-04
IPTS	6.86	6.99	7.25	7.36	7.66	7.89
GD	6.21	6.30	6.47	6.53	6.74	6.86
RS 900	6.03	6.09	6.21	6.26	6.40	6.50
RS 1200	5.98	6.05	6.17	6.21	6.35	6.47
ACO-PTS	6.20	6.26	6.38	6.45	6.58	6.65
ABC-PTS	6.57	6.66	6.83	6.89	7.07	7.21
PSO-PTS	6.63	6.72	6.90	6.96	7.17	7.32
BBO-PTS	5.98	6.05	6.17	6.22	6.37	6.47
GOBBO-PTS	5.98	6.04	6.17	6.21	6.34	6.44
OBBO-PTS	5.99	6.05	6.17	6.22	6.36	6.46
MSFLA-PTS	6.27	6.34	6.48	6.54	6.71	6.81
BGA-PTS	6.20	6.28	6.42	6.48	6.66	6.76

Table 6. Average Rankings achieved by Friedman test.

Method	Average Rank	Normalized values	Rank
IPTS	12.00	8.40	12
GD	8.36	5.85	8
RS 900	5.00	3.50	5
RS 1200	2.36	1.65	2
ACO-PTS	6.07	4.25	6
ABC-PTS	10.00	7.00	10
PSO-PTS	11.00	7.70	11
BBO-PTS	3.14	2.20	4
GOBBO-PTS	1.43	1.00	1
OBBO-PTS	3.07	2.15	3
MSFLA-PTS	8.64	6.05	9
BGA-PTS	6.93	4.85	7

Moreover, in order to give a measure of the computational time required for each algorithm we run 20 simulations of 10^2 random OFDM blocks. The average computational times are reported in Table 8. We notice that the heuristic method IPTS is the fastest, due to its simplicity, however the IPTS results are the worst. BGA-PTS has produced the second best time, which is again a simple algorithm. Both GOBBO-PTS and OBBO-PTS obtain almost the same computational time, which is quite faster than the other remaining algorithms. BBO-PTS is also more computationally efficient than the remaining other methods. The fact that both GOBBO-PTS and OBBO-PTS are faster than original BBO can be expected, since the opposition-based requires simpler operations than the algorithm itself.

Table 7. Wilcoxon signed-rank test between GOBBO-PTS and the other methods.

GOBBO-PTS vs	p-value
IPTS	0.0156
GD	0.0156
RS 900	0.0156
RS 1200	0.1250
ACO-PTS	0.0156
ABC-PTS	0.0156
PSO-PTS	0.0156
BBO-PTS	0.0625
OBBO-PTS	0.0313
MSFLA-PTS	0.0156
BGA-PTS	0.0156

Table 8. Average computational time required for 100 algorithm runs.

Method	Time (s)
IPTS	0.41
GD	58.48
RS 900	31.07
RS 1200	51.27
ACO-PTS	31.40
ABC-PTS	36.75
PSO-PTS	27.68
BBO-PTS	24.62
GOBBO-PTS	23.25
OBBO-PTS	23.23
MSFLA-PTS	26.53
BGA-PTS	18.80

6 Conclusion

In this paper, we propose a new Oppositional BBO algorithm. We apply this algorithm to the PTS problem. The comparison with other existing PTS reduction methods shows that GOBBO-PTS performs better than other methods with the same or larger computational complexity. The proposed algorithm outperformed the original BBO algorithm and the OBBO algorithm. The numerical results that we have shown have proven the effectiveness of this approach. In addition, the other two BBO-PTS schemes can achieve almost the same performance as the RS method with larger computational complexity and outperform other algorithms. All the BBO algorithms used are capable of achieving PAPR reduction, while the GOBBO-PTS performs better than the others.

References

- [1] Hara S. and Prasad R., Overview of multicarrier cdma, *IEEE Communications Magazine*, 1997, 35, 126-133.
- [2] Van Nee R. and Prasad R., *Ofdm for wireless multimedia communications*, OFDM for Wireless Multimedia Communications, Artech House Publishers, 2000.
- [3] Helaly T., Dansereau R., and El-Tanany M., Ber performance of ofdm signals in presence of nonlinear distortion due to sspa, *Wireless Personal Communications*, 2012, 64, 749-760.
- [4] Cimini Jr L. J. and Sollenberger N. R., Peak-to-average power ratio reduction of an ofdm signal using partial transmit sequences, *IEEE Communications Letters*, 2000, 4, 86-88.
- [5] Alavi A., Tellambura C., and Fair I., Papr reduction of ofdm signals using partial transmit sequence: An optimal approach using sphere decoding, *IEEE Communications Letters*, 2005, 8, 982-984.
- [6] Chen J. C., Partial transmit sequences for papr reduction of ofdm signals with stochastic optimization techniques, *IEEE Transactions on Consumer Electronics*, 2010, 56, 1229-1234.
- [7] Cho Y. J., No J. S., and Shin D. J., A new low-complexity pts scheme based on successive local search using sequences, *IEEE Communications Letters*, 1470-1473, 2012, 16.
- [8] Duanmu C. and Chen H., Reduction of the papr in ofdm systems by intelligently applying both pts and slm algorithms, *Wireless Personal Communications*, 2014, 74, 849-863 .
- [9] Gao J., Wang J., and Wang B., Improved particle swarm optimization for papr reduction of ofdm systems, 2010 International Conference on Networking, Sensing and Control, ICNSC 2010, 2010, 621-624.
- [10] Han S. H. and Lee J. H., Papr reduction of ofdm signals using a reduced complexity pts technique, *IEEE Signal Processing Letters*, 2004, 11, 887-890.
- [11] Hasan M. M., Vlm precoded slm technique for papr reduction in ofdm systems, *Wireless Personal Communications*, 2013, 73, 791-801.
- [12] Hung H. L., Using evolutionary computation technique for trade-off between performance peak-to average power ration reduction and computational complexity in ofdm systems, *Computers and Electrical Engineering*, 2011, 37, 57-70.
- [13] Hung H. L. and Huang Y. F., Peak to average power ratio reduction of multicarrier transmission systems using electromagnetism-like method, *International Journal of Innovative Computing, Information and Control*, 2011, 7, 2037-2050.
- [14] Hung H. L., Huang Y. F., Yeh C. M., and Tan T. H., Performance of particle swarm optimization techniques on papr reduction for ofdm systems, *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2008, 2390-2395.
- [15] Jiang T., Xiang W., Richardson P. C., Guo J., and Zhu G., Papr reduction of ofdm signals using partial transmit sequences with low computational complexity, *IEEE Transactions on Broadcasting*, 2007, 53, 719-724.
- [16] Kwon O. J. and Ha Y. H., Multi-carrier pap reduction method using sub-optimal pts with threshold, *IEEE Transactions on Broadcasting*, 2003, 49, 232-236.
- [17] Lim D. W., Heo S. J., No J. S., and Chung H., A new pts ofdm scheme with low complexity for papr reduction, *IEEE Transactions on Broadcasting*, 52, 2006, 77-82.
- [18] Taşpınar N., Kalinli A., and Yıldırım M., Partial transmit sequences for papr reduction using parallel tabu search algorithm in ofdm systems, *IEEE Communications Letters*, 2011, 15, 974-976.
- [19] Tellambura C., Improved phase factor computation for the par reduction of an ofdm signal using pts, *IEEE Communications Letters*, 2001, 5, 135-137.
- [20] Varahram P. and Ali B. M., A low complexity partial transmit sequence for peak to average power ratio reduction in ofdm systems, *Radioengineering*, 2011, 20, 677-682 .
- [21] Varahram P., Mohammady S., and Ali B. M., A robust peak-to-average power ratio reduction scheme by inserting dummy signals with enhanced partial transmit sequence in ofdm systems, *Wireless Personal Communications*, 2013, 72, 1125-1137 .
- [22] Wang Y., Chen W., and Tellambura C., A papr reduction method based on artificial bee colony algorithm for ofdm signals, *IEEE Transactions on Wireless Communications*, 2010, 9, 2994-2999.
- [23] Wang L. and Tellambura C., Cross-entropy-based sign-selection algorithms for peak-to-average power ratio reduction of ofdm systems, *IEEE Transactions on Signal Processing*, 2008, 56, 4990-4994.
- [24] Weng C. E., Chang C. W., Chen C. H., and Hung H. L., Novel low-complexity partial transmit sequences scheme for papr reduction in ofdm systems using adaptive differential evolution algorithm, *Wireless Personal Communications*, 2013, 71, 679-694.
- [25] Wen J. H., Lee S. H., Huang Y. F., and Hung H. L., A suboptimal pts algorithm based on paticle swarm optimization for papr reduction in ofdm systems, *EURASIP J. Wireless Commun. Networking*, 2008, 2008.
- [26] Yang L., Chen R. S., Soo K. K., and Siu Y. M., An efficient sphere decoding approach for pts assisted papr reduction of ofdm signals, *AEU - International Journal of Electronics and Communications*, 2007, 61, 684-688.
- [27] You Y. H., Jeon W. G., Paik J. H., and Jung H. K., Low-complexity par reduction schemes using slm and pts approaches for ofdm-cdma signals, *IEEE Transactions on Consumer Electronics*, 2003, 49, 284-289.
- [28] Yu X., Li S., Zhu Z., and Zhang T., An improved artificial bee colony-partial transmit sequence algorithm for papr reduction in ofdm systems, *International Journal of Wireless and Mobile Computing*, 2013, 6, 473-480.
- [29] Zhang Y., Ni Q., and Chen H. H., A new partial transmit sequence scheme using genetic algorithm for peak-to-average power ratio reduction in a multi-carrier code division multiple access wireless system, *International Journal of Autonomous and Adaptive*

- Communications Systems, 2009, 2, 40-57.
- [30] Zhang Y., Ni Q., Chen H. H., and Song Y., An intelligent genetic algorithm for papr reduction in a multi-carrier cdma wireless system, IWCMC 2008 - International Wireless Communications and Mobile Computing Conference, 2008, 1052-1057.
 - [31] Kim S. S., Kim M. J., and Gulliver T. A., A new pts for papr reduction by local search in ga, in IEEE International Conference on Neural Networks - Conference Proceedings, 2006, 2370-2373.
 - [32] Shukla J., Joshi A., Bansal R., and Tyagi R. K., Papr reduction of ofdm systems using pts with genetic algorithm at low computational complexity, in International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014, 2014.
 - [33] Yao Y. C., Hung H. L., and Wen J. H., Papr reduction in ofdm system using differential evolution-based partial transmit sequences scheme, 3 in Proceedings - International Conference on Machine Learning and Cybernetics, 2011, 1414-1419.
 - [34] Goudos S. K., Evolutionary Algorithms for Wireless Communications — A Review of the State-of-the art. InTech, 2014, 1-24 ch. 1.
 - [35] Wang Y., Chen W. b., and Tellambura C., Genetic algorithm based nearly optimal peak reduction tone set selection for adaptive amplitude clipping papr reduction, IEEE Transactions on Broadcasting, 2012, 58, 462-471.
 - [36] Taspinar N. and Yildirim M., A novel parallel artificial bee colony algorithm and its papr reduction performance using slm scheme in ofdm and mimo-ofdm systems, IEEE Communications Letters, 2015, 19, 1830-1833.
 - [37] Zhou J. b., Dutkiewicz E., Liu R., Huang X., Fang G., and Liu Y., A modified shuffled frog leaping algorithm for papr reduction in ofdm systems, IEEE Transactions on Broadcasting, 2015, 61, 698-709.
 - [38] Kumar A. and Gupta M., A review on ofdm and papr reduction techniques, American Journal of Engineering and Applied Sciences, 2015, 8, 202-209.
 - [39] Simon D., Biogeography-based optimization, IEEE Transactions on Evolutionary Computation, 2008, 12, 702-713.
 - [40] Bhattacharya A. and Chattopadhyay P. K., Biogeography-based optimization for different economic load dispatch problems, IEEE Transactions on Power Systems, 2010, 25, 1064-1077.
 - [41] Boussaïd I., Chatterjee A., Siarry P., and Ahmed-Nacer M., Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks, IEEE Transactions on Vehicular Technology, 2011, 60, 2347-2353.
 - [42] Goudos S. K., Baltzis K. B., Siakavara K., Samaras T., Vafiadis E., and Sahalos J. N., Reducing the number of elements in linear arrays using biogeography-based optimization, in 6th European Conference on Antennas and Propagation, EuCAP 2012, ser. 6th European Conference on Antennas and Propagation, EuCAP 2012, 2012, 1615-1618.
 - [43] Ergezer M., Simon D., and Du D., Oppositional biogeography-based optimization, in Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 2009, 1009-1014.
 - [44] Rahnamayan R. S., Tizhoosh H. R., and Salama M. M. A., Opposition-based differential evolution, IEEE Transactions on Evolutionary Computation, 2008, 12, 64-79.
 - [45] Ergezer M. and Simon D., Oppositional biogeography-based optimization for combinatorial problems, in 2011 IEEE Congress of Evolutionary Computation, CEC 2011, 2011, 1496-1503.
 - [46] Karaboga D. and Basturk B., A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm, Journal of Global Optimization, 2007, 39, 459-471.
 - [47] Dorigo M., Maniezzo V., and Coloni A., Ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26, 29-41.
 - [48] Dorigo M. and Stutzle T., Ant Colony Optimization, Cambridge, MA: The MIT Press, 2004.
 - [49] Kennedy J. and Eberhart R., Particle Swarm Optimization, in IEEE International Conference on Neural Networks, 1942–1948 Piscataway, NJ, 1995.
 - [50] Clerc M., The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, 3 in Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, 1999, 1951-1957.
 - [51] Eusuff M., Lansey K., and Pasha F., Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Engineering Optimization, 2006, 38, 129-154.
 - [52] Haupt R. L. and Haupt S. E., Practical Genetic Algorithms, ser. Practical Genetic Algorithms, New York: Wiley and Sons, 1998.
 - [53] Li X. and Yin M., Multiobjective binary biogeography based optimization for feature selection using gene expression data, IEEE Transactions on Nanobioscience, 2013, 12, 343-353.
 - [54] Tizhoosh H. R., Opposition-based learning: A new scheme for machine intelligence, 1 in Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet, 2005, 695-701.
 - [55] Tizhoosh H. R., Reinforcement learning based on actions and opposite actions, Proceedings of the ICGST International Conference on Artificial Intelligence and Machine Learning, 2005, 94-98.
 - [56] Tizhoosh H. R. and Ventresca M., Oppositional Concepts in Computational Intelligence, ser. Oppositional Concepts in Computational Intelligence, New York: Springer, 2008.
 - [57] Mohseni Pour F. and Gharaveisi A. A., Opposition-based discrete action reinforcement learning automata algorithm case study: Optimal design of a pid controller, Turkish Journal of Electrical Engineering and Computer Sciences, 2013, 21, 1603-1614.
 - [58] Ma H., An analysis of the equilibrium of migration models for biogeography-based optimization, Information Sciences, 2010, 180, 3444-3464.
 - [59] Ma H., Simon D., Fei M., and Chen Z., On the equivalences and differences of evolutionary algorithms, Engineering Applications of Artificial Intelligence, 2013, 26, 2397-2407.

- [60] Derrac J., García S., Molina D., and Herrera F., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, 2011, 1, 3-18.
- [61] García S., Molina D., Lozano M., and Herrera F., A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization, *Journal of Heuristics*, 2008, 15, 617-644.