A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems

Jie Zhou, Eryk Dutkiewicz, *Member, IEEE,* Ren Ping Liu, *Senior Member, IEEE,* Xiaojing Huang, *Senior Member, IEEE,* Gengfa Fang, *Member, IEEE,* and Yuanan Liu, *Member, IEEE,*

***Abstract*—Significant reduction of the peak-to-average power ratio (PAPR) is an implementation challenge in orthogonal frequency division multiplexing (OFDM) systems. One way to reduce PAPR is to apply a set of selected partial transmission sequence (PTS) to the transmit signals. However, PTS selection is a highly complex NP-hard problem and the computational complexity is very high when a large number of subcarriers are used in the OFDM system. In this paper, we propose a new heuristic PTS selection method, the modified chaos clonal shuffled frog leaping algorithm (MCCSFLA). MCCSFLA is inspired by natural clonal selection of a frog colony, it is based on the chaos theory. We also analyze MCCSFLA using the Markov chain theory and prove that the algorithm can converge to the global optimum. Simulation results show that the proposed algorithm achieves better PAPR reduction than using others genetic, quantum evolutionary and selective mapping algorithms. Furthermore, the proposed algorithm converges faster than the genetic and quantum evolutionary algorithms.**

***Index Terms*—OFDM, PAPR, clonal selection algorithm, shuffled frog leaping algorithm.**

I. Introduction

ORTHOGONAL Frequency Division Multiplexing (OFDM) is a multicarrier technique with high bandwidth efficiency. By dividing the bandwidth into many orthogonal subcarriers, it can minimize the impact of multi-path delay and multipath fading [1]. However, one major problem is the high peak-to-average power ratio, which not only reduces system power efficiency, but also results in significant nonlinear distortion when signal passes the amplifier.

In order to reduce the PAPR, techniques such as signal scrambling and signal pre-distortion have been proposed. Signal scrambling techniques include coding methods [2], phase optimization [3][4], partial transmission sequence (PTS) method and selective mapping (SLM). Signal pre-distortion techniques include clipping methods [5]-[7].

Manuscript submitted October 2014. This work has been published in part at the IEEE TENCON 2014, October 2014 [28].

J. Zhou, E. Dutkiewicz and G. Fang are with the Department of Engineering, Macquarie University, Sydney, NSW 2109, Australia (e-mail: jie.zhou8@students.mq.edu.au; eryk.dutkiewicz@mq.edu.au; Gengfa.Fang@mq.edu.au). J. Zhou is also with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China.

R. P. Liu is with the ICT Centre, The Commonwealth Scientific and Industrial Research Organization, Marsfield, N.S.W. 2122, Australia (e-mail: ren.liu@csiro.au).

X. Huang is with the University of Technology, Sydney. Email: Xiaojing.Huang@uts.edu.au

Y. Liu is with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yuliu@bupt.edu.cn).

PTS method is a popular technique for PAPR reduction in OFDM systems. However, the partial transmit sequence selection is a highly complex NP-hard problem and the computational complexity is very high for a large number of OFDM subcarriers. Previous studies aimed at finding a set of selected partial transmit sequences with heuristic and evolutionary algorithms. Among the studies, simulated annealing (SA) and particle swarm optimization (PSO) [8] can achieve lower PAPR than SLM. However, in practice heuristic algorithms suffer from a low convergence rate. The SLM method is proposed in [9] for QAM modulated OFDM signals. It has a lower computational complexity, but the PAPR reduction is not as good as those of the heuristic algorithms. Chen et al. tried to solve the PAPR reduction problem with a quantum evolutionary algorithm (QEA) [10][11], which provides a wider search space. However, quantum evolutionary algorithms also suffer from a low convergence speed. As one of the iteration based approaches, Y. Wang et al. proposed a PAPR reduction method based on Parametric Minimum Cross Entropy for OFDM system [13]. Their method not only reduces the PAPR significantly, but also decreases the computational complexity. Another useful method using real-valued genetic approach has been proposed by J.-K. Lain et al. [13].Their design is a similar concept to genetic algorithm.

Recently, nature inspired approaches have been proved to be very effective in searching for optimal solutions, such as the genetic algorithm (GA) [14], the ant colony optimization (ACO) [12], and the artificial bee colony algorithm (ABC) [15]. They have been used to solve discrete and continuous nonlinear optimization problems. In [16], Eusuff et al. proposed a shuffled frog-leaping algorithm for discrete optimization by using a population-based cooperative search metaphor inspired by natural memetics. Their design is conceptually similar to the genetic algorithm, and is effective for solving combinatorial optimization problems. In their method, the worst frog in each group first jumps to the best frog in the same group to create a new frog. If the new frog is better, the new frog will replace the worst frog. Otherwise, the worst frog will jump to the global best frog. However, their algorithm is easy to fall into premature convergence during the evolutionary process. As a result, it is hard to find a good solution in the billions of possible combinations for PAPR reduction problem in a limited number of iterations.

In this paper, we propose a novel PTS method based on a modified chaos clonal shuffled frog leaping algorithm called MCCSFLA. It is inspired by the natural biological behavior of frogs, and motivated by the chaos theory and clonal selection.

MCCSFLA combines the nature inspired local search with the global information exchange between groups and takes advantage of clonal selection. With such combined strategies, MCCSFLA is able to avoid local suboptimal points, and direct the search toward the global optimum PTS that minimizes PAPR. We present a detailed algorithm design of MCCSFLA for PAPR reduction. The convergence of MCCSFLA is proved through Markov chain theory, where the MCCSFLA iteration process is modeled with Markov chains. Extensive simulations are conducted comparing the proposed algorithm with the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm, and the original method without PTS. Simulation results demonstrate the superior performance of the proposed MCCSFLA in both PAPR reduction as well as fast convergence.

This paper is organized as follows. The system model is given in Section II. In Section III, the modified chaos clonal shuffled frog leaping algorithm for PAPR reduction is presented. Section IV analyzes the convergence of the proposed algorithm with Markov chain theory. In Section V the simulation results are presented and discussed. Finally, Section VI draws the conclusions.

II. System Model

This section describes the system model for PAPR reduction in OFDM systems. In [10], the authors proposed an OFDM PAPR reduction model with binary Signal Sign-Selection from the set $\{-1, 1\}$. Later in [11], the same authors showed a more flexible model with signal sign-selection from the set $\{1, -1, i, -i\}$. In order to facilitate performance comparisons between different PAPR reduction algorithms, in this paper we consider a similar system model as in [10]. Assume an OFDM system with L subcarriers. The discrete time transmitted signal can be represented as:

$$x_n = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} Z_l e^{i2\pi nl(\frac{1}{L})} \tag{1}$$

where n is the discrete time index, i equal to $\sqrt{-1}$, and $Z = [\ Z_0 \quad Z_1 \quad \cdots \quad Z_{L-1}\ ]$ is the input symbol sequence.

As defined in [10], the PAPR of the transmitted signal can be represented as:

$$PAPR = 10\log_{10} \frac{\max\left\{|x_n|^2\right\}}{E\left\{|x_n|^2\right\}} \tag{2}$$

where E is the expected value operation.

Then the input data $Z = [\ Z_0 \quad Z_1 \quad \cdots \quad Z_{L-1}\ ]$ can be divided in to V non-overlapping sub-blocks $\{Y_v, v = 0, 1, \cdots, V-1\}$, which can be shown as

$$Y = [\ Y_1 \quad Y_2 \quad \cdots \quad Y_{V-1}\ ]\,. \tag{3}$$

The objective of the PTS method is to generate an appropriate phase weighting sequence that reduces the PAPR.

The phase weighting sequence is a vector with length V, which can be represented as:

$$D = [\ d_0 \quad d_1 \quad \cdots \quad d_{V-1}\ ] \tag{4}$$

where $d_v = \exp(j\varphi_v)$ is the phase weighting factor, $v \in [1, V-1]$, and $\{\varphi_v, v = 0, 1, \cdots, V-1\}$ are phase factors selected from the range $\varphi_v \in [0, 2\pi)$. However in practice the phase factors are selected from a limited set, which can be represented as:

$$\varphi_v \in \{2\pi\omega/W | \omega = 0, 1, \cdots W-1\} \tag{5}$$

where W is the set of permitted phase factors. In this paper we only consider $\omega = 0, 1, 2, 3$, which means $d_v \in \{1, i, -1, -i\}$.

After selecting a proper phase weighting factor, it is multiplied by the input data to reduce the PAPR, which can be represented as:

$$Y' = [\ d_0 Y_0 \quad d_1 Y_1 \quad \cdots \quad d_{V-1} Y_{V-1}\ ]. \tag{6}$$

After the phase weighting factor optimization, the discrete time transmitted signal can be represented as ${x_n}'(D)$.

So the objective function of the PAPR reduction problem is equivalent to the phase factors search problem, which can be expressed as:

Minimize

$$f(D) = \frac{\max\left\{|{x_n}'(D)|^2\right\}}{E\left\{|{x_n}'(D)|^2\right\}} \tag{7}$$

subject to

$$\varphi_v \in \{2\pi\omega/W | \omega = 0, 1, \cdots W-1\}\,. \tag{8}$$

The goal of our algorithm is to minimize the fitness function $f(D)$. As each OFDM symbol has V sub-blocks, and each phase factor is selected from the set $\varphi_v \in \{2\pi\omega/W | \omega = 0, 1, \cdots W-1\}$, the solution space is W^V. As changing a common angle on the sub-blocks cannot change PAPR, we can set φ_0 to a fixed value, so the solution space can be reduced to W^{V-1}.

Sometimes we have to make a balance between computing power and bandwidth. For example, with sub-blocks of $M = 8$ and $W = 2$, the possible combination of the phase weighting sequence is 2^7=128. In this case exhaustive search can be used to obtain the optimal PTS. However, with the length of the phase weighting sequence $V = 16$ and QPSK, which are the parameter settings considered in this paper, the possible combination of the phase weighting sequence is 4^{15}=$1,073,741,824$. In this case, exhaustive search is not possible in real time. As such, some lower complexity schemes must be used to reduce the computational complexity. The common approach is to design low complexity algorithms to obtain sub-optimal PTS.

In order to obtain the original signal, the receiver must obtain the PTS phase information from the transmitter. In practical systems, the side information can be transmitted via control channels. In some other systems, there can be reserved sub-carriers for side information. In this paper, we assume the side information is transmitted to the receiver via one of the above two methods.

III. PAPR Reduction Based On Modified Chaos Clonal Shuffled Frog Leaping Algorithm

To select a proper PTS, we propose a phase optimization scheme based on MCCSFLA for PAPR reduction. Inspired by the natural population of frogs, MCCSFLA combines a local search with global information exchange among groups and makes use of the advantages of clonal selection. This balanced strategy enables MCCSFLA to avoid local suboptimal points and direct the search towards a low PAPR solution. In our proposed MCCSFLA, the key improvement is that a clone selection operator is added to the traditional SFLA algorithm. By cloning the best individual and carrying out mutation to each copy, the clone selection operator can significantly improve convergence speed of the MCCSFLA algorithm.

In this section, we present the design of MCCSFLA. We first review the basic principle of the original shuffled frog leaping algorithm, and explain encoding and population representation. We then proceed to the main parts of the algorithm in terms of initialization, sorting and grouping, searching, clonal selection and shuffling, as well as the termination condition. We present MCCSFLA in Algorithm 1, followed by its complexity analysis.

A. Brief Description of Basic SFLA

The shuffled frog leaping algorithm (SFLA) is inspired from the natural behavior of the frog [17]. SFLA has been used for a group of discrete and continuous non-linear optimization problems. In SFLA, the population is partitioned into different groups of frogs, and each group consists of a fixed number of frogs. Each frog is considered as a solution in the process of evolution, and frogs in one group can be influenced by frogs in another group by the means of a shuffling process. In SFLA, the information is carried by a meme, which is similar to the gene in genetic algorithms. Groups of memes are called meme complexes, or "memeplexes" and the evolution process of SFLA is also called a memetic evolution. During a memetic evolution, a frog can improve its memes by leaping towards another frog with a better fitness. After the initialization, each group of frogs conducts a local search. Within each group or memeplex, frogs can exchange information with each other, but only the worst frog can jump to another position. After each memeplex finishes the local search, a shuffling strategy is applied to all memeplexes by ordering and then all the frogs are reorganized into new memeplexes according to their fitness. Both local search and shuffling process are repeated until the termination condition is met.

B. Solution Encoding and Population Representations

Assume that there are M memeplexes in the entire population, and each memeplex contains N frogs. The whole frog population is represented as $P = \left\{ \begin{array}{cccc} F_1 & F_2 & \cdots & F_M \end{array} \right\}$, where F_m is the m_{th} memeplex, and $m \in [1, M]$ is the order number of the memeplex in the population. Each memeplex is represented as $F_m = \left\{ \begin{array}{cccc} D_1 & D_2 & \cdots & D_N \end{array} \right\}$, where D_n is the n_{th} frog in the m_{th} memeplex, and $n \in [1, N]$ is the frog index. Assume an OFDM system with V non-overlapping sub-blocks, the n_{th} frog is represented by matrix $D_n = \left[\begin{array}{cccc} d_1 & d_2 & \cdots & d_{V-1} \end{array} \right]$, where $d_v \in \{1, i, -1, -i\}$ and $v \in [1, V-1]$. In this paper, we set the maximum number of iterations of the local search to 10.

C. Generation of Initial Population with a Logistic Map

Before the first iteration, initial memeplexes should be generated. MCCSFLA uses a Logistic map to generate each frog in each memeplexes. The Logistic map is a polynomial map with low complex and chaotic behavior, which was first proposed in [18]. We first generate a random number between 0 and 1 with a Logistic map as:

$$x_{v+1} = 4x_v(1 - x_v). \tag{9}$$

For an OFDM system with V non-overlapping sub-blocks, a feasible solution space for the problem can be represented as $D_n = \left\{ \begin{array}{cccc} d_1 & d_2 & \cdots & d_{V-1} \end{array} \right\}$. So for each frog, there are four options for each meme. We use a simple map to fix d_v as:

$$d_v = \begin{cases} 1 & 0 < x_v < 0.25 \\ -1 & 0.25 \leq x_v < 0.5 \\ i & 0.5 \leq x_v < 0.75 \\ -i & 0.75 \leq x_v < 1 \end{cases} \tag{10}$$

where d_v are the phase weighting factors from the set $\{1, -1, i, -i\}$, and x_v is the chaotic sequence generated from (9). MCCSFLA also needs to set the global iteration counter $Gc = 0$ before the iteration starts.

D. Fitness Calculation

In this paper, we compute fitness $f(D)$ with (7) for each frog. If the value of $f(D)$ is smaller, the frog is better.

E. Shuffling

Before the search process in each iteration, MCCSFLA does the shuffling process for the whole population. First all the frogs in different memeplexes are merged into one population, followed by sorting and grouping. The shuffling process helps the frogs exchange their information among different memeplexes, and promotes the algorithm convergence to the global optimum. In the sorting and grouping process, we sort the frogs in a descending order according to the fitness value calculated by (7), and then partition the generated frogs into M memeplexes. The process of partitioning is as follows:

Suppose there are M memeplexes represented as $\left\{ \begin{array}{cccc} F_1 & F_2 & \cdots & F_M \end{array} \right\}$ and each memeplex contains N frogs, so there are $M \times N$ frogs in the entire population. At the beginning, we put the No. 1 frog into F_1, and put the No. 2 frog into F_2, etc. After we put the No. M frog into F_M we then put the No. $M+1$ frog back to F_1, then put the No. $M+2$ frog to F_2, and so on until the $(M \times N)_{th}$ frog goes to F_M. Through this operation, within each memeplex, the frog ranked first has the largest fitness and PAPR, and the frog ranked last has least fitness and PAPR. That is, within each memeplex, the last frog is the best one. Also, we record the last frog in the last memeplex as the global best frog D_g.

F. Local Search

After the initialization, MCCSFLA starts a local search process to improve the quality of the frog fitness. Each memeplex conducts a local search independently according to a specific strategy. Within each memeplex, we record the frog ranked first which has the largest fitness as the worst frog D_w, and the frog ranked last which has the least fitness as the best frog D_b. The fitness of the worst frog and the best frog in the memeplex can be shown as $f(D_w)$ and $f(D_b)$. Before we enter the local search, we set the local search counter $Lc = 0$. Then the worst frog in each memeplex is updated in the following steps:

1) Jump Toward the Local Best: First, the worst frog in the memeplex D_w changes its position and jumps toward the best frog in the memeplex D_b. Unlike the traditional SFLA, we do not set maximum and minimum jump distance limitations. The meme of the worst frog D_w is changed as follows:

$$D_{sub} = rand \cdot (D_b - D_w) \tag{11}$$

$$D^1_{new} = D_w + D_{sub} \tag{12}$$

where in (11) $rand$ is a binary sequence with length $V - 1$ generated by the Logistic map. In sequence $rand$, if the result of the Logistic map is smaller than 0.5, the number on the corresponding position is equal to 0. Otherwise it is equal to 1. In this way, D_{sub} is the jump sequence with length $V - 1$. Then we calculate the fitness of D^1_{new}. If the fitness of a newly generated frog $f(D^1_{new})$ is smaller than $f(D_w)$, which means the PAPR of the solution D^1_{new} is lower, the algorithm replaces D_w with D^1_{new} and goes to step *5)*. Otherwise the algorithm goes to step *2)*.

2) Jump Toward the Global Best: Replace D_b in (11) with D_g, and repeat the procedure *1)*. D_g is the global best frog defined in section E. So (11) can be renewed as:

$$D_{sub} = rand \cdot (D_g - D_w) \tag{13}$$

and then a new frog is generated with:

$$D^2_{new} = D_w + D_{sub}. \tag{14}$$

We next evaluate the fitness of D^2_{new} with (7). If the fitness of the newly generated frog D^2_{new} is better than D_w, the algorithm replaces D_w with D^2_{new} and goes to step *5)*. If there is still no improvement in fitness, the algorithm goes to step *3)*.

3) Clonal Selection: Do the clonal selection with the clonal selection operator. The detailed procedure is described in section G.

4) Replace with Random Frog: If there is still no improvement, we generate a random frog D^3_{new} with the Logistic map to replace the worst frog D_w in the memeplex. We generate the frog with the same rule as we generate the initial population of frogs, but this time we generate only one frog. After completion of the replacement, the algorithm goes to step *5)*.

5) Reorder: Reorder all the frogs in the memeplex in a descending order according to the fitness value, and set $Lc = Lc + 1$. By doing this a round of local search is completed. In this procedure, if any frog's fitness is better than the global best frog D_g, it replaces D_g.

The above procedure continues until Lc reaches the maximum number of the local search iterations. Each memeplex repeats a certain round of local searches independently for a specific number of generations.

The local search makes a memetic to the exploitation capability of the algorithm by making the information pass in the local space, and it improves the average fitness of the memeplexes.

G. The Clonal Selection Operator

Unlike the traditional SFLA, a chaotic clonal selection operator is applied in MCCSFLA to enhance the efficiency of the local search. Before generating a random frog solution in the local search step, a number of copies of the best solutions are made, and then the frog jumps to the position where PAPR goes down. The main steps of the clonal selection operator are: cloning the best solution, chaotic mutation, frog jump, and fitness calculation.

1) Cloning the Best Solution: The local best frog and the global best frog were chosen for cloning. Both the local best frog and global best frog are cloned to a fixed number of copies. Unlike the traditional clonal selection algorithm, the number of copies has no relationship to the fitness.

2) Chaotic Mutation: The chaotic mutation operator randomly chooses some memes on the frog with a fixed rate and replaces them with the value from the set $\{1, -1, i, -i\}$. The chaotic mutation operator can add additional diversity to the cloned frog, and prevent premature convergence on the offspring.

3) Frog Jump: We create new frogs with (15) and (16):

$$D_{sub} = rand \cdot (D_c - D_w) \tag{15}$$

$$D^4_{new} = D_w + D_{sub} \tag{16}$$

where in (15), D_c is the cloned frog which passed the chaotic mutation operation. The other parameters are the same as in Section III-F. After that, these frogs are evaluated and ranked in the descending order of fitness. If the fitness is improved, the frog rank with the lowest PAPR will replace the worst frog in the memeplex. Otherwise the frog will be replaced with a random frog as in step *4)* in Section III-F.

H. Population Mutation

In order to increase the diversity, a population mutation operator is applied to the whole population. The population mutation is the last operation in each iteration. In this procedure we randomly select a very small percentage of memes in the whole population and replace these memes with random values from the set $\{1, -1, i, -i\}$.

I. Upgrade the Elite Frog

We keep an elite frog D_e which has the lowest PAPR in all the generations of MCCSFLA. If the frog with the lowest PAPR after population mutation D_m has lower PAPR than D_e, we replace all values of D_e with the corresponding values of D_m.

J. Termination Condition

After shuffling, MCCSFLA will check whether the termination condition is satisfied. The termination condition is when the algorithm global iteration counter Gc reaches the designated number of iterations.

K. Basic Steps

The basic steps of MCCSFLA are described in Algorithm 1:

Algorithm 1 The modified chaos clonal shuffled frog leaping algorithm based PTS method

Begin
Generate the initial population with Logistic map
Evaluate the fitness value
Set $Gc = 0$
while Gc has not reached designated global iterations **do**
  Sort and group the population into memeplexes
  Set $Lc = 0$
  while Lc has not reached designated global iterations **do**
    Frog jumps to the local best frog
    if no improvement **then**
      Frog jumps to the global best frog
    end if
    if no improvement **then**
      Clonal selection operator
    end if
    if no improvement **then**
      Replace with a random frog
    end if
    Reorder all the frogs and set $Lc = Lc + 1$
  end while
  Evaluate the fitness value
  Shuffle different memeplex into one population
  Population Mutation
  Upgrade the Elite Frog
  $Gc = Gc + 1$
end while
Evaluate the fitness value
Output the best frog with lowest PAPR
End

L. Computational complexity analysis

As the computing power is mainly consumed in the IFFT operation and PAPR calculation, we only consider the computational complexity of these two parts. There are V sub-blocks, and each sub-block need to be modulated with a L-point IFFT. Thus $(LV\log_2 L)/2$ complex multiplications and $LV\log_2 L$ complex additions are needed for IFFT. As the computational complexity of one complex multiplication equal to four real multiplications and two real additions, and one complex addition equal to two real additions, the IFFT operation need $2LV\log_2 L$ real multiplications and $3LV\log_2 L$ real additions.

For each sample of phase weighting sequence, we need $L(V-1)$ complex additions to generate the sample, and L real additions and $2L$ multiplications to calculate the PAPR[21]. For conventional PTS, the number of sample is W^{V-1}. So the number of real multiplications and real additions for conventional PTS (CON-PTS) are

$$CP_{mul} = 2W^{V-1}L + 2LV\log_2 L \tag{17}$$

$$CP_{add} = W^{V-1}L + 2W^{V-1}(V-1)L + 3LV\log_2 L \tag{18}$$

For GA-PTS, the samples number equals to $S_1 = Pop1 \times Gen1$, where $Pop1$ is the number of individuals in the population and $Gen1$ is the maximum number of generations in GA-PTS. So the number of real multiplications and real additions for GA-PTS are

$$GA_{mul} = 2LS_1 + 2LV\log_2 L \tag{19}$$

$$GA_{add} = LS_1 + 2(V-1)LS_1 + 3LV\log_2 L \tag{20}$$

For QEA-PTS, the samples number equals to $S_2 = Pop2 \times Gen2$, where $Pop2$ is the number of individuals in the population and $Gen2$ is the maximum number of generation in QEA-PTS. So the number of real multiplications and real additions for QEA-PTS are

$$QEA_{mul} = 2LS_2 + 2LV\log_2 L \tag{21}$$

$$QEA_{add} = LS_2 + 2(V-1)LS_2 + 3LV\log_2 L \tag{22}$$

For MCCSFLA-PTS, as only the worst frog in each group (memeplex) is renewed, the samples number equals to the $S_3 = AJ \times M \times Gen3$, where $Gen3$ is the maximum number of generations in MCCSFLA, and M is the number of groups in MCCSFLA. The AJ is the average number of jumps for the worst frog in each group. If a better solution can be found via jump toward the local best, the number of jumps is 1. If a better solution can be found via jump toward the global best, the number of jumps is 2. If a better solution can be found via clonal selection, the number of jumps is equal to the number of clones plus 2. Otherwise the number of jumps is equal to the number of clones plus 3, as we need replace the worst frog in the group with a random frog.

$$MCCSFLA_{mul} = 2LS_3 + 2LV\log_2 L \tag{23}$$

$$MCCSFLA_{add} = LS_3 + 2(V-1)LS_3 + 3LV\log_2 L \tag{24}$$

For SLM, the samples number equals to the number of distinct sign sequences S_4. So the number of real multiplications and real additions for GA-PTS are

$$SLM_{mul} = 2LS_4 + 2LV\log_2 L \tag{25}$$

$$SLM_{add} = LS_4 + 2(V-1)LS_4 + 3LV\log_2 L \tag{26}$$

IV. Convergence Analysis of MCCSFLA for PAPR Reduction in OFDM Systems

In this section we analyse the convergence of MCCSFLA for OFDM systems. We first model the iteration process of MCCSFLA as a Markov chain[19], and derive its properties. The base Markov chain is then extended to include the elite frog, and the properties of such extended Markov chain are investigated further. Under these properties, we prove MCCSFLA converges to global optimum.

A. The Base Markov Chain

First, we give some basic definitions in Markov chain. A discrete-time Markov chain is a random process that can only take values from a discrete state space. In Markov chain, the next state only depends on the current state and not on the previous states[22]. A Markov chain is homogeneous if it is inrelevant to time (time-invariant). In a homogenous Markov chain, a transition matrix can be represent as a two-dimensional matrix $P = [p_{ij}]$. For a homogenous Markov chain, with the initial distribution $\vec{\pi}(0)$, the probability distribution of in the time n is only depends on $\vec{\pi}(0)$ and P. Some homogenous Markov chain is Ergodic, which is both irreducible and aperiodic.

In the iteration process of MCCSFLA, the population renews from one iteration to another. If we use standard Markov tools to analyze this evolutionary process, MCCSFLA runs from one population distribution at one iteration to another at the next generation, and this can be viewed as a random process, and the population distribution at each iteration may be considered as a state. As the evolution process of the elite frog D_e is independent from the other frogs, first we just consider the evolution process without elite frog D_e.

Lemma 1: The population sequence $\{Y(n)\}_{n=1}^{\infty}$ of MCCSFLA without elite frog D_e constitutes a Markov chain.

Proof: Let $Y(n)$ be the whole frog population at iteration n with multiple memeplex. From the steps of MCCSFLA, we can know that the population $Y(n+1)$ is obtained from the population $Y(n)$ with a sequence of operations, which means that the distribution probability of the next iteration has nothing to do with the former iteration $Y(n-1)$ and initial population $Y(0)$. Thus, MCCSFLA can be modelled with a Markov chain and its character may be studied by the Markov chain theory.

In the procedure of analyze, we use a binary string to represent the whole population in one state[23]. In this way, each gene in a frog is mapped to two binary bits, and the number of bits for each frog is $2(V-1)$. As there are M memeplex in population and N frogs in each memeplex, the whole population in state n is mapped to $Y(n) = \{\ D_1,\ \cdots,\ D_{MN}\ \} = \{\underbrace{x_1 \cdots x_{2(V-1)}}_{D_1} \cdots \underbrace{x_{2(MN-1)(V-1)+1} \cdots x_{2MN(V-1)}}_{D_{MN}}\}$. For each frog, the number of possible strings for individual space is $2^{2(V-1)}$, so the solution space for each generation is $|\Phi| = 2^{2(V-1)MN}$, so the state space of the Markov chain is finite.

From the operation steps of MCCSFLA, we can obtain the following conclusions.

Lemma 2: The Markov chain of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is time homogeneous.

Proof: Because the transition probability of the operation on each iteration remains fixed, the Markov chain of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is homogeneous, and the transition probability $P_{ij}^{(n)}$ is irrelevant to step n, which can be denoted as $P = [p_{ij}]$ with size $2^{2(V-1)MN} \times 2^{2(V-1)MN}$. So the Theorem is proved.

Now we focus on some important properties of the operations in MCCSFLA. Each iteration process of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is divided into two steps. The step 1 is the operations before the population mutation, and the corresponding line numbers in flowchart **Algorithm 1** is from 6 to 22. We use the transition matrix G to represent the first step. The step 2 is the population mutation, and the corresponding line numbers in flowchart **Algorithm 1** is 23. We use the transition matrix C to represent the second step. We don't consider the elite operation in line 24 in the base Markov chain. So the whole transition matrix P is the product of the transition matrices in two steps.

Lemma 3: The transition matrix G of step 1 for the population sequence $\{Y(n)\}_{n=1}^{\infty}$ in MCCSFLA is stochastic.

Proof: Mathematically, these operations map probabilistically from one state to another in space Φ with transition matrix $G = [g_{ij}]$. Due to the length binary-coded frog is $2^{2((V-1)MN)}$, which is unchanged in the whole process of algorithm, the total probability from one state to all states that from $[0, 0, \cdots, 0]_{1\times 2^{2((V-1)MN)}}$ to $[1, 1, \cdots, 1]_{1\times 2^{2((V-1)MN)}}$ in the next iteration is 1. Thus $g_{ij} \geq 0$ and $\sum_{j=1}^{2^{2(V-1)MN}} g_{ij} = 1$ for any $i \in [1, 2^{2(V-1)MN}]$. Therefore, the transition matrixes before the mutation operation is stochastic.

Lemma 4: The state transition matrix C of step 2 for the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is both stochastic and positive.

Proof: The population mutation operation in step 2 randomly maps from one state to another, and it works on each bit string independently. Let $C = [c_{ij}]$ be the state transition matrix of the population mutation operation, and $Pc \in (0, 1)$ be the mutation probability on each bit. The transition probability can be shown as

$$c_{ij} = Pc^{H_{ij}}(1 - Pc)^{2(V-1)MN - H_{ij}} > 0 \qquad (27)$$

where c_{ij} is transition probability, and H_{ij} is the Hamming distance between state i and state j [24]. As all c_{ij} are positive, the state transition matrix C is positive. From the above formula we can also get $\sum_{j=1}^{2^{2(V-1)MN}} m_{ij} = 1$ for any $i \in [1, 2^{2(V-1)MN}]$. Thus, C is stochastic.

From the above two lemmas, we can get the following conclusions.

Lemma 5: Let G be the transition matrix in step 1, and C be the state transition matrix of step 2 in MCCSFLA. Then the product GC is a positive and stochastic matrix.

Proof: Let transition matrix $P = GC$. So we have

$$P=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,2^{2(V-1)MN}} \\ p_{21} & p_{22} & \cdots & p_{2,2^{2(V-1)MN}} \\ \vdots & \vdots & \ddots & \vdots \\ p_{2^{2(V-1)MN},1} & p_{2^{2(V-1)MN},2} & \cdots & p_{2^{2(V-1)MN},2^{2(V-1)MN}} \end{bmatrix} \tag{28}$$

Since for any j in stochastic matrix G, we have $\sum_{j=1}^{2^{2(V-1)MN}} g_{ij} = 1$, so there is at least one positive element in each row in matrix G. As C is a stochastic positive matrix, for all i and j $c_{ij} > 0$, so for all i and j $p_{ij} = \sum_{k=1}^{2^{2(V-1)MN}} g_{ik} \cdot c_{kj} > 0$. Thus G is a positive and stochastic matrix.

Definition 1: [20] A Markov chain is called an ergodic chain if it is possible to go from every state to every state (not necessarily in one move).

Theorem 1: The population sequence $\{Y(n)\}_{n=1}^{\infty}$ of MCCSFLA is an ergodic Markov chain.

Proof: According to Lemma 5 $p_{ij} > 0$ for any i and j, which means that it is possible to reach every state from every state in just one move. As the solution space $|\Phi| = 2^{2(V-1)MN}$ for each generation is finite, according to the definition 1, the population sequence of MCCSFLA is an ergodic Markov chain.

According to the properties of the ergodic Markov chain, the population sequence $\{Y(n)\}_{n=1}^{\infty}$ has a stationary probability distribution $P^{\infty} = e' \cdot p^{\infty}$ when $n \to \infty$, where $p^{\infty} = (p_1, p_2, \cdots, p_m)$ and $\sum_{l=1}^{m} p_l = 1$. As transition matrix $P > 0$, according to the definition of the primitive matrix, P is also primitive.

B. The Extended Elite Markov Chain

As in MCCSFLA we keep the elite frog D_e with lowest PAPR found by the past and current generations, we can enlarge the population by adding D_e in front of other frogs. In this way, the new population at generation n can be expressed as $Y^{+}(n) = \{ D_e, \ D_1 \cdots, \ D_{MN} \} = \{\underbrace{x_1 \cdots x_{2(V-1)}}_{D_e} \cdots \underbrace{x_{2MN(V-1)+1} \cdots x_{2(MN+1)(V-1)}}_{D_{MN}}\}$.

Lemma 6: The stochastic process of the enlarged population sequence $\{Y^{+}(n)\}_{n=1}^{\infty}$ of MCCSFLA constitutes a Markov chain.

Proof: Let $Y^{+}(n)$ be the whole frog population at iteration n with multiple memeplexes. From the steps of MCCSFLA, we know that population $Y^{+}(n+1)$ is obtained from population $Y^{+}(n)$ with a sequence of operations, which means that the distribution probability of next iteration has nothing to do with former iteration $Y^{+}(n-1)$. Thus, MCCSFLA can be modelled with a Markov chain and its properties may studied by the Markov chain theory.

For each frog in population $Y^{+}(n)$, the number of possible states for individual space is $2^{2(V-1)}$, and with the best frog the total number of frogs in the population is $MN+1$, so the solution space for each generation is $|\phi'| = 2^{2(V-1)(MN+1)}$, so the state space of Markov chain is also finite.

We divide the evolution process of population $Y^{+}(n)$ into two steps. We use transition matrix $B(n)$ to indicate the whole state transition matrix of the two steps. In the first step, all frogs will be renewed except the elite frog D_e. In the algorithm, this stage includes all steps in the outermost loop before upgrading the elite frog, and the corresponding line numbers in flowchart **Algorithm 1** is from 6 to 23. Mathematically, population $Y^{+}(n)$ will upgrade from $Y^{+}(n) = \{D_e, D_1, \cdots, D_{MN}\}$ to $Y^{+}(n) = \{D_e, D_1^{new}, \cdots, D_{MN}^{new}\}$. We use transition matrix $Q(n)$ to indicate this step. In the second step, the elite frog D_e will be upgraded, and the corresponding line numbers in flowchart **Algorithm 1** is 24. If the frog with lowest PAPR in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ has lower PAPR than D_e, we replace all values of D_e with this frog. We used the transition matrix $U(n)$ to represent this upgrade stage.

Lemma 7: The enlarged population sequence $\{Y^{+}(n)\}_{n=1}^{\infty}$ of MCCSFLA constitutes a time homogeneous Markov chain.

Proof: In the first stage, all frogs are renewed except the elite frog D_e. We have proved this process is time homogeneous in Lemma 2, so the transition matrix $Q(n)$ can be written as Q. In the second stage, the upgrade only depends on the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ in the current generation. So the upgrade is irrelevant to step n, and the transition matrix $U(n)$ can be represented as U. So the enlarged population sequence $\{Y^{+}(n)\}_{n=1}^{\infty}$ of MCCSFLA is time homogeneous and the transition matrix $B(n)$ can be represented as B. So the whole transition matrix B is the product of the transition matrices Q and U in two steps.

Theorem 2: The transition matrix B of population sequence $Y^{+}(n)$ in MCCSFLA is stochastic.

Proof: Mathematically, these operations map probabilistically from one state to another in solution space $|\phi'| = 2^{2(V-1)(MN+1)}$ with transition matrix B, and the total probability from one state to all states in the next iteration is 1. Thus, if we use $B = [b_{ij}]_{2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}}$ to represent the transition matrix $B(n)$, we can get $\sum_{j=1}^{2^{2(V-1)(MN+1)}} b_{ij} = 1$ for any $i \in \left[1, 2^{2(V-1)(MN+1)}\right]$. Therefore, the transition matrixes before the mutation operation is stochastic.

To simplify the description, we encode the state number in the solution space according to the PAPR of elite frog D_e. $Y^{+}(n)$ with better D_e has the higher position in solution space, and the states with the same D_e are listed together. In other words, D_e with lower PAPR will be listed higher.

To make the discussion easier, we assume there is only one global best frog D_{best} in the whole solution space. The globe best frog has the lowest PAPR. As the size of matrix B is $2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}$, the states with the elite frog equal to D_{best} are listed with state number $i \in [1, 2^{2(V-1)MN}]$. Similarly, the states with the elite frog with the highest PAPR are listed with the state number $i \in \left[2^{2(V-1)MN}(2^{2(V-1)} - 1) + 1, 2^{2(V-1)(MN+1)}\right]$.

In the first stage, since the elite frog D_e is not affected by transition matrix P in MCCSFLA, state transition matrix Q can be represented as

$$Q = \begin{pmatrix} P & & & \\ & P & & \\ & & \ddots & \\ & & & P \end{pmatrix} \quad (29)$$

where in state transition matrix Q there are $2^{2(V-1)}$ matrices P on the diagonal, and the size of each P is $2^{2(V-1)MN} \times 2^{2(V-1)MN}$.

In the second stage, as we need to upgrade the elite frog D_e with the best frog found in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$, we need to define the upgrade transition matrix $U = [u_{ij}]$. The upgrade transition matrix simply replaces the elite frog D_e with the best frog D_m found after the population mutation in the current generation if the PAPR of D_m is lower than D_e, or otherwise does nothing.

Lemma 8: Upgrade transition matrix U is a lower triangular matrix.

Proof: Suppose D_m is the best frog found in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ after the population mutation at generation n, and D_e is the elite frog at generation $n-1$.

(1) For $Y^+(n)$ with state number i in generation n, if the PAPR of D_m is lower than D_e, then D_m will replace D_e in generation n, and the state will transtion from state i to state j. According to the sequence of the state number, state j has a higher position in the solution space, so $i > j$ and $u_{ij} = 1$. In this case, the other elements in row i in the upgrade transition matrix U are 0.

(2) For state $Y^+(n)$ with state number i in generation n, if the PAPR of D_m is higher than D_e, then the state i will not change. In this case, $u_{ii} = 1$ and the other elements in row i in the upgrade transition matrix U are 0.

The above results can directly prove upgrade transition matrix U is a lower triangular matrix.

We split U into matrix blocks with the same size as matrix P. In this way, U can be expressed as:

$$U = \begin{pmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{2^{2(V-1)},1} & U_{2^{2(V-1)},2} & \cdots & U_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix} \quad (30)$$

Lemma 9: U_{11} is a unit matrix.

Proof: As the states with the elite frog equal to D_{best} are listed with state number $i \in [1, 2^{2(V-1)MN}]$, the elite frog in these states will not be replaced by other frogs, as it has the lowest PAPR in the whole solution space.

In matrix U, each row represent a current state, and in such state the frog with lowest PAPR is fixed, so the next state is also fixed. With the upgrade matrix, the next state will either go up with a smaller state number, or remain in the current state. With the above analyze, we can immediately draw the following conclusions: In each row there is only one $u_{ij} = 1$, other elements equal to 0, and $i \geq j$.

Lemma 10: $U_{21} \neq 0$ and $U_{22} \neq 0$.

Proof: Let $Y^+(n) = \{D_e, D_1^{new}, \cdots, D_{MN}^{new}\}$. Since the relevant rows of transition matrix U_{21} and U_{22} contain all the binary combinations of $\{D_1^{new}, \cdots, D_{MN}^{new}\}$, in some rows the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ contains D_{best}, for example $D_1^{new} = D_{best}$, in this case the elite frog D_e will be replaced by D_1^{new}. So there are some elements in U_{21} with the form $u_{i1} = 1$. So $U_{21} \neq 0$. In some rows D_e has the lowest PAPR, so the status will not be changed. In this case $u_{ii} = 1$. So $U_{22} \neq 0$.

In this way, as U_{11} is a unit matrix, the total transition matrix can be expressed as [26]

$$B = QU = \begin{pmatrix} P & & & \\ PU_{21} & PU_{22} & & \\ \vdots & \vdots & \ddots & \\ PU_{2^{2(V-1)},1} & PU_{2^{2(V-1)},2} & \cdots & PU_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix} \quad (31)$$

C. The Convergence for MCCSFLA

Before the convergence proof, we need to define the concept of convergence of MCCSFLA for PAPR reduction.

Definition 2: Let D_e be the elite frog found by MCCSFLA at generation n, and D_{best} be the best frog with lowest PAPR f_{best} in the whole population space Φ. If $\lim_{n\to\infty} \Pr(f(D_e) = f_{best}) = 1$, then population sequence $\{P(n)\}_{n=1}^{\infty}$ converges.

In order to analyze the convergence of MCCSFLA with the Markov chain theory, we give the following Lemma.

Lemma 11: [22]: Let P be a $m \times m$ primitive stochastic matrix that converges to $P^\infty = (\pi^T, \pi^T, \cdots, \pi^T)^T$ with $\pi^T = (p_1, p_2, \cdots, p_m)$, $R, T \neq 0$, $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times k}$. If a $k \times k$ non-negative stochastic phalanx B has the form $B = \begin{pmatrix} P & 0 \\ R & T \end{pmatrix}$, then $B^\infty = \lim_{n\to\infty} \begin{pmatrix} P^n & 0 \\ \sum_{m=0}^{n-1} T^m R P^{n-m-1} & T^n \end{pmatrix} = \begin{pmatrix} P^\infty & 0 \\ R^\infty & 0 \end{pmatrix}$ is a stable stochastic matrix, and $B^\infty = g' \cdot b^\infty$, where $b^\infty = (p_1, p_2, \cdots, p_m, 0, 0, \cdots 0)$ and $\sum_{l=1}^{m} p_l = 1$.

The detailed proof can be found in [22].

Based on the above analysis we can prove MCCSFLA converges to the global optimum when the number of iterations tends to be infinite.

Theorem 3: MCCSFLA can guarantee convergence to the global optimum.

Proof: Let the total transition matrix be expressed as $B = \begin{pmatrix} P & 0 \\ R & T \end{pmatrix}$, with $R = \begin{pmatrix} PU_{21} \\ \vdots \\ PU_{2^{2(V-1)},1} \end{pmatrix}$ and $T = \begin{pmatrix} PU_{22} & & \\ \vdots & \ddots & \\ PU_{2^{2(V-1)},2} & \cdots & PU_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix}$. In the above we proved that P and B are primitive stochastic matrices. As $P > 0$, $U_{21} \neq 0$, the product $PU_{21} \neq 0$, so $R \neq 0$. As $P > 0$, $U_{22} \neq 0$, the product $PU_{22} \neq 0$, so $T \neq 0$. According to the Theorem 1, $\{Y(n)\}_{n=1}^{\infty}$ converges to $B^\infty = g' \cdot b^\infty$, where $b^\infty = (p_1, p_2, \cdots, p_{2^{2(V-1)MN}}, 0, 0, \cdots 0)$, and $\sum_{l=1}^{2^{2(V-1)MN}} p_l =$

1. So we have

$$
\begin{aligned}
B^{\infty} &= \lim_{n\to\infty}\begin{pmatrix} P^{n} & 0 \\ \sum_{m=0}^{n-1} T^{m}RP^{n-m-1} & T^{n}\end{pmatrix}\\
&= \begin{pmatrix} P^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix}\\
&= \begin{pmatrix} p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \\ p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \end{pmatrix}
\end{aligned} \tag{32}
$$

The size of matrix B is $2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}$, so it has $2^{2(V-1)(MN+1)}$ states. As for status with state number $i \in [1, 2^{2(V-1)MN}]$ the elite frog $D_e = D_{best}$, according to the definition of the convergence in Definition 2, $\{Y(n)\}_{n=1}^{\infty}$ converges to the global optimal solution.

V. Experimental Study

In this section we present the simulation results of MCCSFLA. In order to demonstrate the algorithm's capabilities, we compare it against the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm and the original method without using partial transmission sequences.

In our simulation, we consider the QPSK, 16QAM and 64QAM modulation respectively. We set the number of non-overlapping sub-blocks $V = 16$. The objective function in (7) is used to evaluate each PAPR reduction algorithm. For simplicity, the input symbol sequence is considered to be randomly distributed. The population sizes of both GA and QEA are 40, and the number of frogs in MCCSFLA is also 40. We set the number of groups in MCCSFLA to 4, which means there are 10 frogs in each group. In GA, we use a similar setting to that in [6]. We set the crossover probability to 0.9, and set the mutation probability to 0.05. In QEA, we use the same lookup table as in [27]. For comparison purposes, we set the number of distinct sign sequences of SLM to 8. In each simulation, we test the algorithms with 128, 256, and 512 subcarriers in one symbol respectively.

In order to compare the performance of MCCSFLA, the complementary cumulative distribution function (CCDF) is used to evaluate the algorithms. The CCDF is given as:

$$CCDF = P\{PAPR > PAPR_0\} \tag{33}$$

where P is the probability function.

Fig. 1 to Fig. 9 show the CCDF curves of MCCSFLA, GA, QEA, SLM and "original" with the QPSK, 16QAM and 64QAM modulation respectively. The maximum generation of MCCSFLA, GA and QEA is set to 20. "Original" means the PAPR without using PTS. We simulate the OFDM system with 128, 256 and 512 subcarriers respectively. Each case is tested with 1×10^5 symbols independently. As can be seen from Fig.1, among all the different subcarriers, MCCSFLA provides better performance than GA, QEA, SLM and "original". For example, with 20 iterations and 128 subcarriers, when $CCDF = 10^{-3}$, the average PAPR obtained by MCCSFLA is around 5.59 dB, which is the lowest among all the algorithms under evaluation. In contrast, GA and QEA achieve results for PAPR of around 6.25 dB and 7.15 dB. In comparison, SLM can only obtain a higher PAPR of 7.53 dB due to the few partial transmit sequences. "Original", however, suffers a substantial performance loss, with the highest PAPR of around 10.40 dB. Similar conclusions can be observed from Fig. 2 to Fig. 9.

As a numerical example for computational complexity, we use the parameters in Fig. 1, that is $W = 4$, $V = 16$, $L = 128$ $Gen1 = Gen2 = Gen3 = 20$, $Pop1 = Pop2 = 40$, $M = 4$, and set the number of distinct sign sequences in SLM is set to 8. With 1×10^5 symbols simulation, we can get the average number of jumps for the worst frog in each group is $AJ = 8.7$. So we have $W^{V-1} = 1073741824$. According to section III-L, the computational complexity of PAPR reduction schemes is calculated and shown in table I. From the tables we can see that the computational complexity of MCCSFLA-PTS is a little bit lower than GA-PTS and QEA-PTS with $L = 128$. The proposed MCCSFLA-PTS achieved best PAPR reduction as shown in Fig. 1 to Fig. 9. As we can see from Fig. 1 and table I, although SLM has a lower computational complexity as shown in Table I, its PAPR reduction performance is the worst among heuristic algorithms.

TABLE I
THE NUMBER OF REAL MULTIPLICATIONS AND REAL ADDITIONS OVER THE CONVENTIONAL PTS

Scheme	Real Multiplications	Real Additions
CON-PTS	274877935616	4260607600644
GA-PTS	233472	3217408
QEA-PTS	233472	3217408
MCCSFLA-PTS	206848	2804736
SLM	30720	74752

Fig. 10, Fig. 11 and Fig. 12 show the convergence, defined as the average PAPR for 100 symbols, for QPSK modulation, MCCSFLA and GA with 128, 256 and 512 subcarriers respectively. Fig. 10 shows clearly that MCCSFLA performs significantly better than GA in terms of the convergence speed with 128 subcarriers. It can be seen in Fig. 10, at the first generation, the average PAPR of MCCSFLA is lower than GA as the local search of MCCSFLA is more effective than GA. Within the initial 50 iterations, the PAPR of MCCSFLA decreased quickly with the growth of the generations, as the convergence speed improves with the local search and the shuffling of the population. On the other hand, GA shows a slower convergence rate than MCCSFLA. From 50 to 100 iterations, MCCSFLA has approached close to 4.55 dB, while GA is still far from it. Over all 100 iterations, MCCSFLA provides a lower average PAPR than GA, and MCCSFLA converges with a faster rate. From Fig. 11 and Fig. 12, similar conclusions can also be obtained when the number of the subcarriers is 256 and 512 respectively. Overall, MCCSFLA is more effective and suitable for PAPR reduction than GA. Moreover, similar conclusions can also be obtained from the simulation results that MCCSFLA converges faster than QEA with the same number of generations.

VI. Conclusion

In this paper, we propose a novel modified chaos clonal shuffled frog leaping algorithm for partial transmit sequence

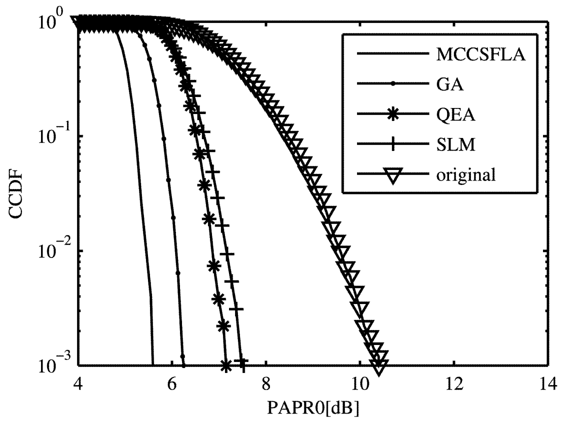


Fig. 1. QPSK, CCDF of the PAPR with subcarrier=128.

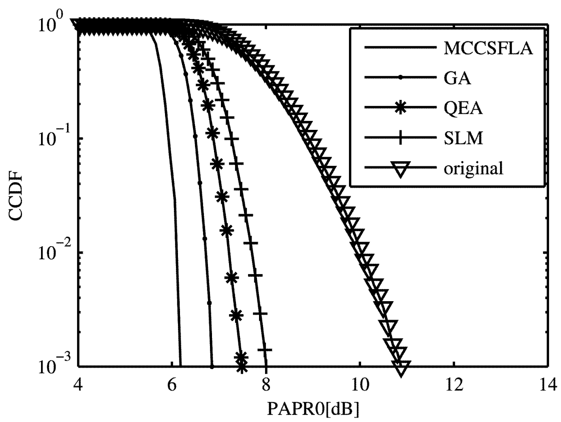


Fig. 2. QPSK, CCDF of the PAPR with subcarrier=256.

selection in OFDM systems. We analyze the proposed algorithm using Markov chain theory and prove that the algorithm converges to the global optimum. Simulation experiments are conducted to compare the proposed algorithm with the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm and the original approach respectively. The results show that the modified chaos clonal shuffled frog leaping algorithm is more efficient in terms of both PAPR reduction and convergence speed.

ACKNOWLEDGMENT

Jie Zhou is a recipient of a Macquarie University Research Excellence Scholarship and is under a cotutelle PhD arrangement between Macquarie University and Beijing University of Posts and Telecommunications. This work was supported in part by the National Natural Science Foundation of China (No.61170275), the National Science and Technology Major Project (No.2012ZX03001001-002), and the research project of Guangdong (No.2011B090400433).

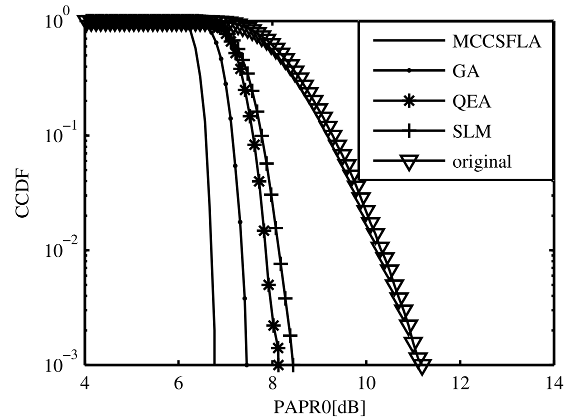


Fig. 3. QPSK, CCDF of the PAPR with subcarrier=512.

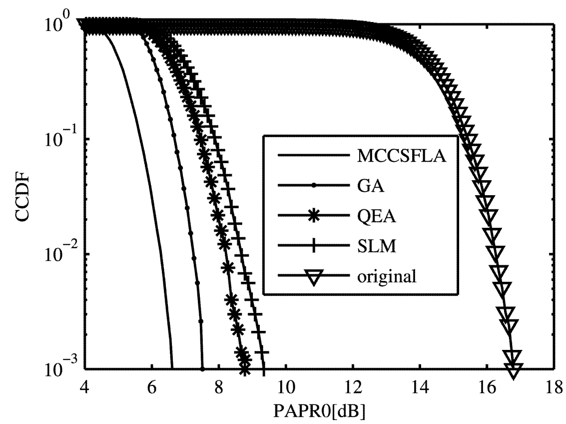


Fig. 4. 16QAM, CCDF of the PAPR with subcarrier=128.

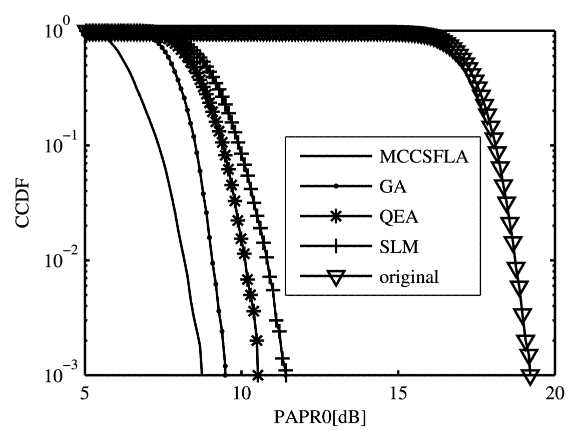


Fig. 5. 16QAM, CCDF of the PAPR with subcarrier=256.

REFERENCES

[1] Y. Rahmatallah and S. Mohan, "Peak-To-Average Power Ratio Reduction in OFDM Systems: A Survey And Taxonomy," *IEEE Commun. Surveys Tuts.*, vol. PP, pp. 1-26, 2013.

[2] O. Muta, "Effect of phase control-based peak-to-average power ratio reduction on multi-input multi-output adaptive modulated vector coding systems," *IET Commun.*, vol. 6, pp. 2769-2774, 2012.

[3] G. Lu, P. Wu and D. Aronsson, "Peak-to-average power ratio reduction in OFDM using cyclically shifted phase sequences," *IET Commun.*, vol. 1, pp. 1146-1151, 2007.

[4] E. Hong, Y. Park, S. Lim and D. Har, "Adaptive phase rotation of OFDM signals for PAPR reduction," *IEEE Trans. Consum. Electron.*, vol. 57, pp.

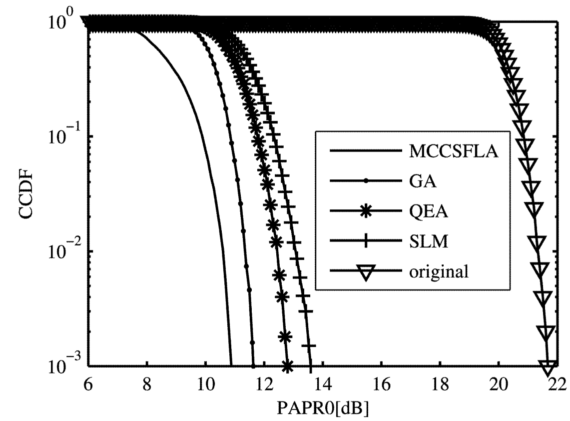


Fig. 6. 16QAM, CCDF of the PAPR with subcarrier=512.

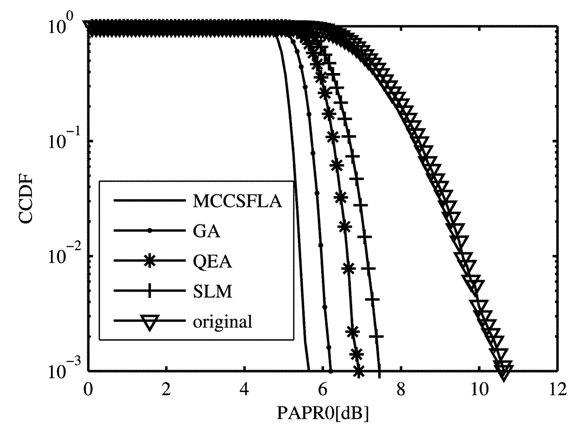


Fig. 7. 64QAM, CCDF of the PAPR with subcarrier=128.

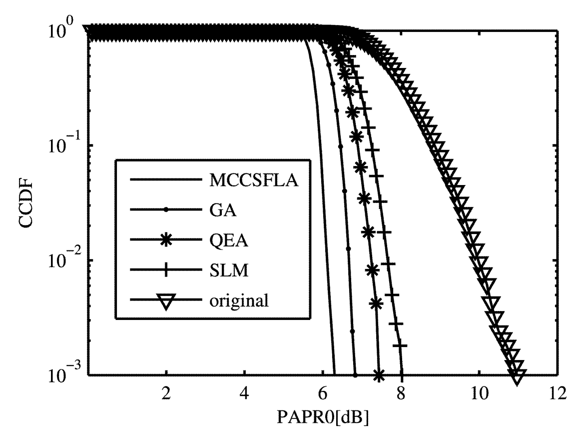


Fig. 8. 64QAM, CCDF of the PAPR with subcarrier=256.

1491-1495, 2011.

[5] X. Zhu, W. Pan, H. Li and Y. Tang, "Simplified Approach to Optimized Iterative Clipping and Filtering for PAPR Reduction of OFDM Signals," *IEEE Trans. Commun.*, vol. 61, pp. 1891-1901, 2013.

[6] Y. Wang, W. Chen and C. Tellambura, "Genetic Algorithm Based Nearly Optimal Peak Reduction Tone Set Selection for Adaptive Amplitude Clipping PAPR Reduction," *IEEE Trans. Broadcast.*, vol. 58, pp. 462-471, 2012.

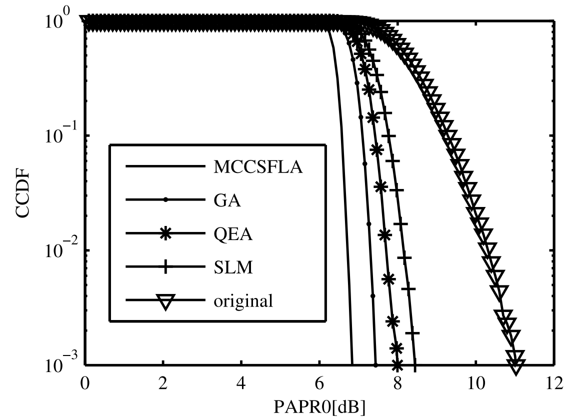


Fig. 9. 64QAM, CCDF of the PAPR with subcarrier=512.

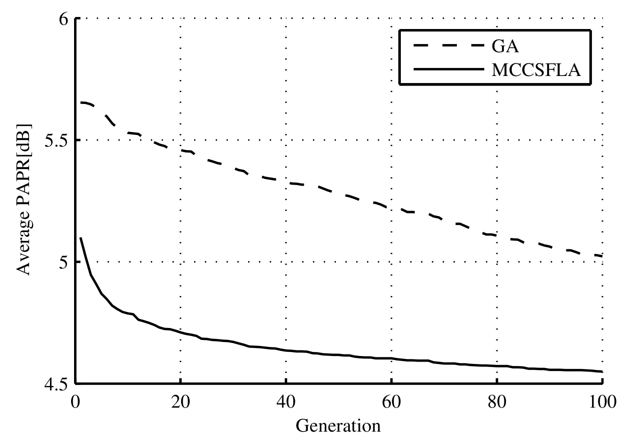


Fig. 10. Average PAPR change by generation with subcarrier=128.

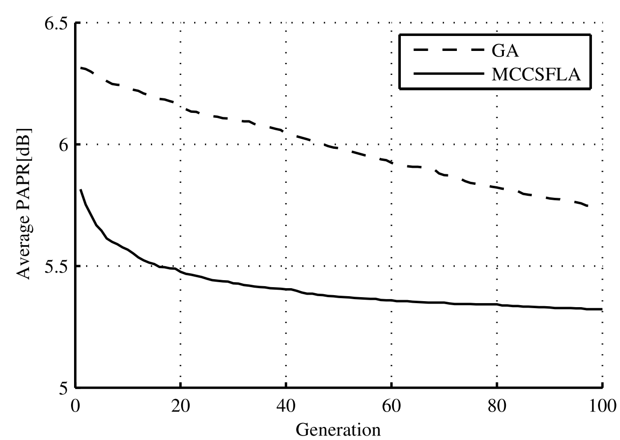


Fig. 11. Average PAPR change by generation with subcarrier=256.

[7] J. Wang, Y. Guo and X. Zhou, "PTS-clipping method to reduce the PAPR in ROF-OFDM system," *IEEE Trans. Consum. Electron.*, vol. 55, pp. 356-359, 2009.

[8] J. Chen, "Partial transmit sequences for PAPR reduction of OFDM signals with stochastic optimization techniques," *IEEE Trans. Consum. Electron.*, vol. 56, pp. 1229-1234, 2010.

[9] H. Jeon, K. Kim, J. No and D. Shin, "Bit-Based SLM Schemes for PAPR Reduction in QAM Modulated OFDM Signals," *IEEE Trans. Broadcast.*,

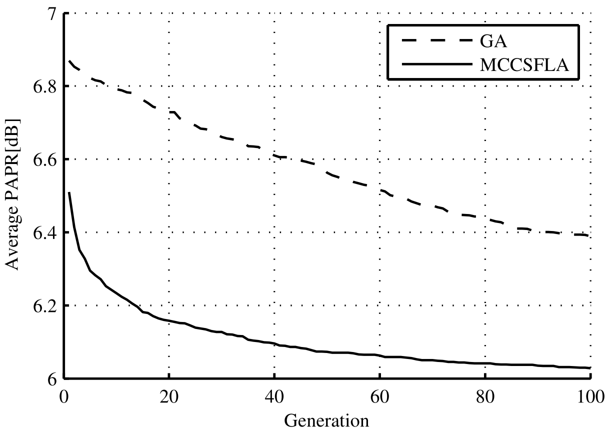


Fig. 12. Average PAPR change by generation with subcarrier=512.

vol. 55, pp. 679-685, 2009.

[10] J. Chen and C. Wen, "A Low-Complexity Scheme to Reduce the PAPR of an OFDM Signal Using Sign-Selection Algorithms," *IEEE Signal Process. Lett.*, vol. 17, pp. 189-192, 2010.

[11] J. Chen, "Application of Quantum-Inspired Evolutionary Algorithm to Reduce PAPR of an OFDM Signal Using Partial Transmit Sequences Technique," *IEEE Trans. Broadcast.*, vol. 56, pp. 110-113, 2010.

[12] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, pp. 29-41, 1996.

[13] Y. Wang, Wen Chen, and C. Tellambura, "PAPR Reduction Method Based on Parametric Minimum Cross Entropy for OFDM Signals," *IEEE Communications Letters*, vol. 14, no. 6, pp.563C565, 2010.

[14] J. K. Lain, S. Y. Wu, and P. H. Yang, "PAPR reduction of OFDM signals using PTS:a real-valued genetic approach," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, pp.1-8, 2011.

[15] F. S. Abu-Mouti and M. E. El-Hawary, "Optimal Distributed Generation Allocation and Sizing in Distribution Systems via Artificial Bee Colony Algorithm," *IEEE Trans. Power Del.*, vol. 26, pp. 2090-2101, 2011.

[16] M. M. Eusuff, K. E. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no.2, pp. 129-154, 2006.

[17] A. Khorsandi, A. Alimardani, B. Vahidi and S. H. Hosseinian, "Hybrid shuffled frog leaping algorithm and Nelder-Mead simplex search for optimal reactive power dispatch," *IET Gener. Transm. Distrib.*, vol. 5, pp. 249-256, 2011.

[18] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol.261, pp.459-467, 1976.

[19] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Transactions on Neural Networks*, vol. 5, pp. 96-101, 1994.

[20] C. M. Grinstead, J. L. Snell, *Grinstead and Snell's Introduction to Probability*, pp.433, 2009.

[21] H. Li, T. Jiang and Y. Zhou, "A Novel Subblock Linear Combination Scheme for Peak-to-Average Power Ratio Reduction in OFDM Systems," *IEEE Transactions on Broadcasting*, vol. 58, pp. 360-369, 2012.

[22] M. Iosifescu, *Finite Markov Processes and Their Applications. Chichester: Wiley*, pp.126, 1980.

[23] D. Simon, M. Ergezer, D. W. Du and R. Rarick, "Markov Models for Biogeography-Based Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, pp. 299-306, 2011.

[24] S. Y. Yuen and K. S. Cheung, "Bounds for probability of success of classical genetic algorithm based on hamming distance," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 1-18, 2006.

[25] O. Francois, "An evolutionary strategy for global minimization and its Markov chain analysis," *IEEE Transactions on Evolutionary Computation*, vol. 2, pp. 77-90, 1998.

[26] J. Suzuki, "A Markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, pp. 655-659, 1995.

[27] R. Nowotniak and J. Kucharski, "GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem," *Bull. Pol. Ac.: Tech.*, vol. 60, pp. 323-330, 2012.

[28] J. Zhou, E. Dutkiewicz, R. P. Liu, G. F. Fang, Y. A. Liu and X. J. Huang, "A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems," in Proc. IEEE TENCON 2014, Bangkok, Thailand, October 22-25, 2014.

Jie Zhou received his B.E. degree in Electronic and Information Engineering from the University of Science and Technology of China in 2006. He is currently a joint PhD student under a cotutelle PhD arrangement between Macquarie University and Beijing University of Posts and Telecommunications. He is now working on Wireless Communications in the Department of Electronic Engineering, Macquarie University, Sydney, Australia. His current research interests include medical body area networks, OFDM system and wireless sensor networks.

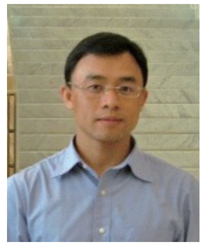
Eryk Dutkiewicz received his B.E. degree in Electrical and Electronic Engineering from the University of Adelaide in 1988, his M.S. degree in Applied Mathematics from the University of Adelaide in 1992 and his PhD in Telecommunications from the University of Wollongong in 1996. He is currently a Professor of Wireless Communications in the Department of Engineering, Macquarie University, Sydney, Australia. He has held visiting professorial appointments at several institutions including the Chinese Academy of Sciences and the Shanghai JiaoTong University. His current research interests cover medical body area networks and cognitive radio networks.

Ren Ping Liu received his B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China, and the Ph.D. degree from the University of Newcastle, Australia. He is a Principal Scientist of networking technology in CSIRO. He is also an Adjunct Professor at Macquarie University, and University of Technology, Sydney. His research interests include MAC protocol design, Markov analysis, QoS scheduling, TCP/IP internetworking, and network security. He has over 100 research publications in leading international journals and conferences.

Professor Liu is a Senior Member of IEEE. He is the founding chair of IEEE NSW VTS Chapter. He served as TPC chair, as OC co-chair, and in Technical Committee in a number of IEEE Conferences. He has also been heavily involved in and led commercial projects delivering networking solutions to government and industry customers.

Xiaojing Huang (M99, SM11) received his Bachelor of Engineering, Master of Engineering, and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1983, 1986, and 1989, respectively, all in electronic engineering. With over 26 years of combined industrial, academic, and scientific research experience, he has published over 230 book chapters, refereed journal and conference papers, and major commercial research reports, and filed 29 patents.

He is currently a professor of information and communications technology in the School of Computing and Communications and the program leader for mobile sensing and communications in the Global Big Data Technologies Centre at the University of Technology, Sydney (UTS), Australia. He is also an Adjunct Professor at Macquarie University, Sydney, Australia. Before joining UTS, he had been a principal research scientist at Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia, and the project leader of CSIRO microwave and mm-wave backhaul projects since 2009, an associate professor at University of Wollongong, Australia, since 2004, and a principal research engineer at Motorola Australian Research Centre since 1998. Prof. Huang is a recipient of CSIRO Chairmans Medal and the Australian Engineering Innovation Award in 2012 for exceptional research achievements in multi-gigabit wireless communications.



Gengfa Fang received his Bachelor in Electronics Engineering from Jilin University, Master in Telecommunications from Zhejiang University and PhD in Wireless Communications from the Chinese Academy of Sciences in 1999, 2002 and 2007 accordingly. From Oct. 2007 to May 2009, he was a Researcher at the Canberra Research Lab of National ICT Australia (NICTA) on WCDMA Femtocell project. In 2009, he joined the Department of Engineering Macquarie University, where he is now a Senior Lecturer. He has published over 60 papers and 5 patents on MAC protocols, wireless resource management and allocation for 5G, Cognitive Radio Networks and Medical Body Area Networks.

Yuanan Liu received the B.E., M.Eng., and Ph.D. degrees in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 1984, 1989, and 1992, respectively.

He joined the 26th Institute of Electronic Ministry of China, Langfang, China, in 1984, where he was involved in developing the inertia navigating system. In 1992, he held his first post-doctoral position with the EMC Laboratory, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. In 1995, he held his second post-doctoral position with the Broadband Mobile Laboratory, Department of System and Computer Engineering, Carleton University, Ottawa, ON, Canada. Since 1997, he has been a Professor with the Wireless Communication Center, College of Telecommunication Engineering, BUPT, where he is involved in the development of next-generation cellular system, wireless LAN, Bluetooth application for data transmission, EMC design strategies for high-speed digital system, and EMI and EMS measuring sites with low cost and high performance.