Research Article **Open Access**

Bishnu P. De, Rajib Kar*, Durbadal Mandal, and Sakti P. Ghoshal

Optimal high speed CMOS inverter design using craziness based Particle Swarm Optimization Algorithm

DOI 10.1515/eng-2015-0031

Received July 30, 2014; accepted April 20, 2015

Abstract: The inverter is the most fundamental logic gate that performs a Boolean operation on a single input variable. In this paper, an optimal design of CMOS inverter using an improved version of particle swarm optimization technique called Craziness based Particle Swarm Optimization (CRPSO) is proposed. CRPSO is very simple in concept, easy to implement and computationally efficient algorithm with two main advantages: it has fast, near-global convergence, and it uses nearly robust control parameters. The performance of PSO depends on its control parameters and may be influenced by premature convergence and stagnation problems. To overcome these problems the PSO algorithm has been modified to CRPSO in this paper and is used for CMOS inverter design. In birds' flocking or fish schooling, a bird or a fish often changes direction suddenly. In the proposed technique, the sudden change of velocity is modelled by a direction reversal factor associated with the previous velocity and a "craziness" velocity factor associated with another direction reversal factor. The second condition is introduced depending on a predefined craziness probability to maintain the diversity of particles. The performance of CRPSO is compared with real code.gnetic algorithm (RGA), and conventional PSO reported in the recent literature. CRPSO based design results are also compared with the PSPICE based results. The simulation results show that the CRPSO is superior to the other algorithms for the examples considered and can be efficiently used for the CMOS inverter design.

Keywords: CMOS inverter; switching characteristics; rise time; fall time; propagation delay; Evolutionary Optimization Technique; Particle Swarm Optimization; craziness based Particle Swarm Optimization

1 Introduction

In digital VLSI domain, CMOS inverter design is considered to be a fundamental step as the design procedure of other complex digital integrated circuits is primarily based on the design procedure of CMOS inverter. Due to the comple.gowth in VLSI circuits, the task of optimal integrated circuit design by hand is very difficult. Evolutionary computation may be a competent implement for the automatic design of digital integrated circuits (IC) that has been one of the most challenging topics in VLSI design process. A CMOS-based integrated circuit is made up of NMOS and PMOS transistors, where the transistor geometries (i.e., channel length (L) and channel width (W)) and other necessary circuit component values are considered to be the design parameters. The values of the design parameters are to be found out such that the performances of the circuits are optimized.

Different evolutionary optimization techniques suitably used for different optimization problems are Genetic Algorithm (GA) which is instigated by the Darwin's "Survival of the Fittest" strategy [1], swarm intelligence mimicked in particle swarm optimization (PSO) and its variants [2, 3]. Conventional PSO simulates the behaviour of bird flocking or fish schooling [2, 4–8]. GA is a probabilistic heuristic search optimization technique developed by Holland [9]. GA is applied for the optimal design of two dimensional recursive filters [10] and FIR log filters [11]. GA has also been used for the synthesis of passive analog circuits to get optimal values of R, L and C elements from a given set of specifications [12]. VLSI circuit partitioning [13], placement and area optimization of soft modules in VLSI floor plan stage [14] have been developed using GA.

PSO is an evolutionary algorithm developed by Eberhart *et al.* [15, 16]. PSO is very simple to implement and its convergence may be controlled by a few parameters. PSO was efficiently utilized in various application areas. Design of digital IIR filter using PSO has been suggested in [17]. PSO was chosen for the placement and routing of

Bishnu P. De, Durbadal Mandal: Department of Electronics and Communication Engg., NIT Durgapur, India, E-mail: durbadal.bittu@gmail.com

***Corresponding Author: Rajib Kar:** Department of Electronics and Communication Engg., NIT Durgapur, India, E-mail: rajibkarece@gmail.com

Sakti P. Ghoshal: Department of Electrical Engg., NIT Durgapur, India, E-mail: spghoshalnitdgp@gmail.com

the field programmable gate arrays (FPGA) to minimize the distances between Configurable Logic Blocks (CLBs) [18]. PSO was adopted for image segmentation to estimate the parameters in the mixture density function for minimization of the square error between the density function and the actual histogram [19]. PSO was also applied for the synthesis of microstrip coupler and single shunt stub matching circuits [20]. Optimal design of analog circuits such as differential amplifier with current mirror load and two-stage operational amplifiers have been carried out using PSO algorithm in [21].

The limitations of the conventional PSO are premature convergence and stagnation problem [22, 23]. To overcome these problems, the PSO algorithm is modified by some signum factors and a "Craziness" factor and called as craziness based PSO (CRPSO). CRPSO is a near-global search algorithm initiated from PSO; it mimics the particle behaviours of a swarm in a very close manner. CRPSO adopts the special features such as abrupt change of velocity; a craziness factor; and change of direction of flying towards an apparently non-promising area of food depending upon particle's mood. These features enhance the usefulness of this algorithm. CRPSO has been applied for the optimal design of linear phase FIR high pass filter [7], FIR band stop filter [3] and IIR filter [24].

Fall time (t_f) of the output voltage of CMOS inverter is estimated using PSO in [25]. Design of CMOS inverter having symmetrical waveform of output voltage with equal rise time (t_r) and fall time (t_f) is investigated using PSO in [25, 26]. Design of CMOS inverter with equal delay times (t_f, t_r), and propagation delay times (t_{pHL}, t_{pLH}) of output voltage using PSO is reported in [25, 27].

In this paper, the optimal switching characteristics of CMOS inverter are explored using real code.gnetic algorithm (RGA) and CRPSO algorithm, independently. It has been realized that RGA is incompetent for local searching [28] in a multidimensional search space and also suffers from premature convergence and gets easily trapped to suboptimal solution [29]. Simulation results achieved from RGA and CRPSO algorithms are compared with those of the recently reported PSO based results [25, 26] to demonstrate the effectiveness and superiority of the performance of CRPSO in achieving the near-global optimal solutions.

The rest of the paper is arranged as follows: In Section 2, the evolutionary techniques under consideration namely RGA, conventional PSO and CRPSO are discussed briefly. Switching characteristics of CMOS inverter are described in Section 3. In Section 4, the objective functions used in this paper are formulated and RGA, CRPSO based inverter design examples are discussed in detail. Discussion of results, validation of results by PSPICE simulator and comparison with PSO based reported results [25, 26] are given in Section 5. Finally, Section 6 concludes the paper.

2 Evolutionary algorithms employed

2.1 Real Coded Genetic Algorithm (RGA)

Standard Genetic Algorithm (also known as real coded GA) is a population-based, robust optimization method, based on the ideas of natural selection and evolution built upon the Darwin's "Survival of the Fittest" strategy [9]. Each generation consists of a population of character strings that are analogous to the chromosome. Each chromosome i.e.aluated by the cost function of the corresponding optimization problem. Each chromosome has a probability of selection and has to take part in the genetic operations such as crossover and mutation based upon the Roulette's wheel strategy to evolve to a global optimal or near-global optimal solution to the problem at hand. The details of RGA are given in [30]. The algorithmic steps of RGA for the problem under consideration are as follows:

Step 1: Initialize the real coded chromosome strings ($\boldsymbol{\omega}$) of n_P(=10) population, each consisting of the number of parameters (dimension of the optimization problem (D=3)) need to be optimized. Each parameter has a maximum boundary and a minimum boundary and is randomly generated within this range. Maximum iteration/genetic cycles (=250 or 500 depending on the case study) is defined. Mutation probability=0.003; Crossover ratio=0.8; Selection probability=1/3.

Step 2: Decoding of the strings and evaluation of cost error function (CF).

Step 3: Selection of elite strings in order of increasing cost function values from the minimum value.

Step 4: Copying the elite strings over the non-selected strings.

Step 5: Crossover and mutation to generate offsprings.

Step 6: Genetic cycle updating.

Step 7: The genetic cycle stops when the termination criteria of maximum genetic cycles is satisfied. The grand minimum CF and its corresponding chromosome string or the desired solution are finally obtained.

2.2 Particle Swarm Optimization (PSO)

PSO is a flexible, robust population-based stochastic search/optimization technique with implicit parallelism, which can easily handle non-differential objective functions, unlike traditional optimization methods. The details of PSO are given in [3].

Mathematically, velocities of the particle vectors are modified according to the following equation:

$$\begin{aligned} V_i^{(k+1)} &= w * + C_1 * rand_1 * \left(pbest_i^{(k)} - S^{(k)}\right) \\ &+ C_2 * rand_2 * \left(gbest^{(k)} - S_i^{(k)}\right), \end{aligned} \quad (1)$$

where $V_i^{(k)}$ is the velocity of ith particle at kth iteration; w is the weighting function; C_1 and C_2 are the positive weighting factors; $rand_1$ and $rand_2$ are the random numbers between 0 and 1; $S_i^{(k)}$ is the current position of ith particle vector at kth iteration; $pbest_i^{(k)}$ is the personal best of ith particle vector at kth iteration; $gbest^{(k)}$ is the group best of the group at kth iteration. The searching point in the solution space may be modified by the following equation:

$$S_i^{(k+1)} = S_i^{(k)} + V_i^{(k)}. \quad (2)$$

The first term of (1) is the previous velocity of the particle vector. The second and third terms are used to change the velocity of the particle. Without the second and third terms, the particle will keep on "flying" in the same direction until it hits the boundary. Namely, it corresponds to a kind of inertia represented by the inertia constant, and tries to explore new areas.

2.3 Craziness based Particle Swarm Optimization (CRPSO)

In order to get rid of the limitations of classical PSO [22, 23] already mentioned and because in birds' flocking or fish schooling, a bird or a fish often changes directions suddenly, conventional PSO is modified by introducing an entirely new velocity expression (3) associated with many random numbers and a "craziness velocity" having a predefined probability of craziness. This modified PSO is termed as CRPSO.

The velocity in this case can be expressed as follows [3, 7]:

$$\begin{aligned} V_i^{(k+1)} &= r_2 * sign(r_3) * V_i^{(k)} \\ &+ (1 - r_2) * C_1 * r_1 * \left\{pbest_i^{(k)} - S_i^{(k)}\right\} \\ &+ (1 - r_2) * C_2 \times (1 - r_1) * \left\{gbest^{(k)} - S_i^{(k)}\right\}, \end{aligned} \quad (3)$$

where r_1, r_2 and r_3 are the random parameters uniformly taken from the interval [0, 1] and $sign(r_3)$ is a function defined as:

$$\begin{aligned} sign(r_3) &= -1 \text{ where } r_3 \le 0.05 \\ &= 1 \text{ where } r_3 > 0.05. \end{aligned} \quad (4)$$

A craziness operator is introduced in the proposed technique to ensure that the particle would have a predefined craziness probability to maintain the diversity of the particles. Consequently, before updating its position the velocity of the particle is crazed by,

$$V_i^{(k+1)} = V_i^{(k+1)} + P(r_4) * sign(r_4) * v_i^{craziness}, \quad (5)$$

where r_4 is a random parameter which is chosen uniformly within the interval [0, 1]; $v_i^{craziness}$ is a random parameter which is uniformly chosen from the interval [$v_i^{\min}$, $v_i^{\max}$]; and $P(r_4)$ and $sign(r_4)$ are defined, respectively, as:

$$\begin{aligned} P(r_4) &= 1 \text{ when } r_4 \le P_{cr} \\ &= 0 \text{ when } r_4 > P_{cr} \end{aligned} \quad (6)$$

$$\begin{aligned} sign(r_4) &= -1 \text{ when } r_4 \ge 0.5 \\ &= 1 \text{ when } r_4 < 0.5, \end{aligned} \quad (7)$$

where is a predefined probability of craziness.

The details of CRPSO regarding the parameters used in [3] are given in [31, 32]. The updated searching point in the solution space is given by (2).

The algorithmic steps of CRPSO algorithm as implemented for the problem under consideration are as follows:

Step 1: Initialization: Population (swarm size) of particle vectors, $n_P = 10$; dimension of the optimization problem, D=3; maximum iteration cycles; target error; fixing algorithm's control parameters, C_1, C_2, P_{cr},$v^{craziness}$; define limits of design parameters; initialization of the velocities of all the particle vectors, $v_{initial} = 0.001$.

Step 2: Generate initial particle vectors randomly within limits; computation of cost function (CF) for each particle vector.

Step 3: Computation of population based minimum cost function value and computation of the personal best solution vectors ($hpbest$), group best solution vector ($hgbest$).

Step 4: Updating the velocities as per (3) and (5); updating the particle vectors as per (2) and checking against the limits of the design parameters; finally, computation of the updated cost function values of the particle vectors and population based minimum cost function value.

Step 5: Updating the $hpbest$ vectors, the hgbest vector; replace the updated particle vectors as initial particle vectors for step 4.

Step 6: Iteration continues from step 4 till the maximum iteration cycles is met; finally, *hgbest* is the vector, representing the optimal design parameters of CMOS inverter.

2.4 Statistical comparison of accuracy between two algorithms

Two sample *t*-test is a hypothesis testing method for determining the statistical significance of the difference between two independent samples of an equal sample size [33]. The *t*-test value will be positive if the second algorithm is better than the first, and it is negative if it is poorer. The *t*-value is defined as given in (8).

$$t = \frac{\overline{\alpha}_1 - \overline{\alpha}_2}{\sqrt{\left(\frac{\sigma_1^2}{\beta+1} + \frac{\sigma_2^2}{\beta+1}\right)}}, \tag{8}$$

where $\overline{\alpha}_1$ and $\overline{\alpha}_2$ are the mean values of the first and the second methods, respectively; σ_1 and σ_2 are the standard deviations of the first and the second methods, respectively; and β is the value of the degree of freedom. When the *t*-value is higher than 1.645 ($\beta = 49$), there is a significant difference between the two algorithms with a 95% confidence level. The *t*-value is larger than 2.15 (degree of freedom = 49), which means that there is a significant difference between the two algorithms with a 98% confidence level. In all the Case studies considered in this work, RGA and CRPSO are considered as algorithm 1 and algorithm 2, respectively.

3 Switching characteristics of CMOS inverter

The operating speed of a digital system is determined by the switching characteristics of the logic gates used to build the system. As the inverter is the basic logic gate of any digital IC technology, the switching characteristics of the inverter are the fundamental parameters to characterize the technology. Therefore, the switching speed of the circuit must be approximated and optimized very early in the design phase to ensure circuit reliability and performance. Here, the optimal switching characteristics of CMOS inverter are investigated using RGA and the proposed CRPSO and also compared with those determined by conventional PSO as reported in [25, 26]. The schematic diagram of a CMOS inverter is shown in Figure 1. Rise time (t_r) and fall time (t_f) of the output voltage are shown in

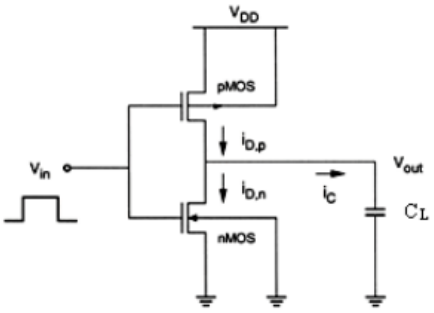

Figure 1: Schematic diagram of a CMOS inverter.

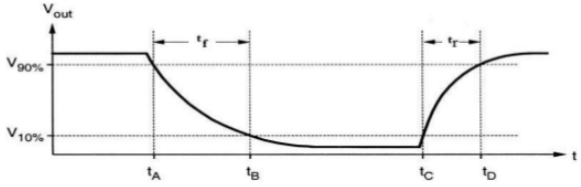

Figure 2: Output voltage rise time (t_r) and fall time (t_f).

Figure 2. The input and output voltage waveforms of CMOS inverter circuit are shown in Figure 3.

The switching operation of the CMOS inverter is analyzed to determine its fall time (t_f), rise time (t_r) and propagation delay times (t_{pHL}, t_{pLH}) It is presumed that a pulse waveform is applied to the input of the inverter. The fall time (t_f) is the time required for the output voltage to drop from V_{90}% level to V_{10}% level. Similarly, the rise time (t_r) is defined as the time required for the output voltage to rise from V_{10}% level to V_{90}% level. The propagation delay times t_{pHL} and t_{pLH} establish the input to output signal delays during high-to-low and low-to-high transitions of the output, respectively. The high-to-low propagation delay (t_{pHL}) is defined as the time delay between the V_{50}% transition of the rising input voltage and V_{50}% transition of the

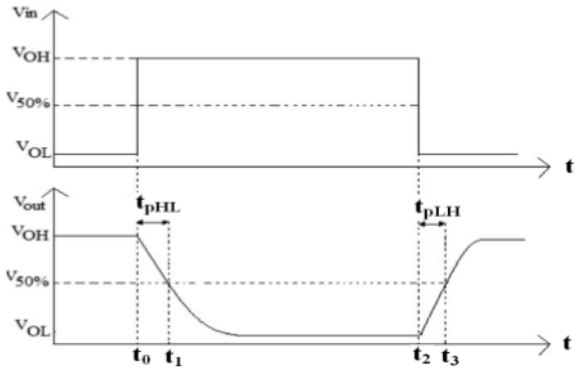

Figure 3: Input and output voltage waveforms of CMOS inverter and definitions of propagation delay times.

falling output voltage. Similarly, the low-to-high propagation delay (t_{pLH}) is the time delay between the V_{50}% transition of the falling input voltage and V_{50}% transition of the rising output voltage.

To compute fall time (t_f) of the output voltage, output load capacitance (C_L) should be discharged through the active NMOS transistor, considering PMOS transistor is in cut-off region. The fall time is given as follows [34]:

$$t_f = \frac{C_L}{\mu_n C_{OX}\left(\frac{W}{L}\right)_n (V_{DD}-V_{tn})}\left[\frac{2(V_{tn}-0.1V_{DD}}{(V_{DD}-V_{tn})} + \ln\left(\frac{(2(V_{DD}-V_{tn}))-0.1V_{DD}}{0.1V_{DD}}\right)\right]. \quad (9)$$

To calculate rise time (t_r) of the output voltage, output load capacitance (C_L) should be charged through the active PMOS transistor, considering NMOS transistor is in cut-off region. The rise time is given in (10) [34].

$$t_r = \frac{C_L}{\mu_p C_{OX}\left(\frac{W}{L}\right)_p (V_{DD}-\|V_{tp}\|)}\left[\frac{2(|V_{tp}|-0.1V_{DD})}{(V_{DD}-|V_{tp}|)} + \ln\left(\frac{(2(V_{DD}-|V_{tp}|))-0.1V_{DD}}{0.1V_{DD}}\right)\right]. \quad (10)$$

Study of propagation delay times t_{pHL} and t_{pLH} involves discharging output load capacitance (C_L) through active NMOS transistor and charging the same C_L through active PMOS transistor, respectively. To simplify the analysis and the derivation of delay expressions, the input voltage waveform is usually assumed to be an ideal step pulse with zero rise and fall times. Under this consideration, t_{pHL} becomes the time required for the output voltage to fall from V_{OH} to the V_{50}% level, and t_{pLH} becomes the time required for the output voltage to rise from V_{OL} to the V_{50}% level. For the CMOS inverter, V_{OH}=V_{DD} and V_{OL}=0 are considered. The propagation delay times are given as follows [35]:

$$t_{pHL} = \frac{C_L}{\mu_n C_{OX}\left(\frac{W}{L}\right)_n (V_{DD}-V_{tn})}\left[\frac{2V_{tn}}{(V_{DD}-V_{tn})} + \ln\left(\frac{(4(V_{DD}-V_{tn}))}{V_{DD}} - 1\right)\right] \quad (11)$$

$$t_{pLH} = \frac{C_L}{\mu_p C_{OX}\left(\frac{W}{L}\right)_p (V_{DD}-|V_{tp}|)}\left[\frac{2|V_{tp}|}{(V_{DD}-|V_{tp}|} + \ln\left(\frac{(4(V_{DD}-|V_{tp}|))}{V_{DD}} - 1\right)\right]. \quad (12)$$

4 Problem formulation

This paper describes three different case studies to characterize the CMOS inverter switching. In Case study-1, the fall time (t_f) of output voltage for the CMOS inverter i.e.plored. Case study-2 is for the design of CMOS inverter having symmetric waveform of output voltage with equal rise time (t_r) and fall time (t_f). In Case study-3, the CMOS inverter is designed with improved symmetry for the output voltage with equal rise time (t_r) and fall time (t_f) and equal high-to-low propagation delay (t_{pHL}) and low-to-high propagation delay (t_{pLH}).

4.1 Case study-1

In this case, the main objective is to estimate the fall time of output voltage of a CMOS inverter, as given in (9), with the least error value. During the design process, values of design parameters such as fall time (t_f), output load capacitance (C_L) and aspect ratio (W/L) of MOS structures (equal sized NMOS and PMOS) should be reserved in certain ranges. RGA and CRPSO algorithms are individually utilized to find out the optimal solution set that consists of the exact values of t_f, C_L and (W/L) ratio of NMOS transistors within the given ranges. The cost/error function (CF) is defined as:

$$CF = \left|\mu_n C_{OX}\left(\frac{W}{L}\right)_n t_f - \frac{C_L}{(V_{DD}-V_{tn})}\left[\frac{2(V_{tn}-0.1V_{DD})}{(V_{DD}-V_{tn})} + \ln\left(\frac{(2(V_{DD}-V_{tn}))-0.1V_{DD}}{01.V_{DD}}\right)\right]\right|. \quad (13)$$

The error fitness function is given as

$$J = 10\log_{10}(CF). \quad (14)$$

In order to obtain the exact values of the design parameters, CF is set to a value very close to zero. Equating CF to very closely zero means that the error is nearly equal to zero and t_f is successfully estimated depending on the design parameters. Here, TSMC 0.25 micron fabrication technology parameters[1] used for this design are as follows: V_{DD} =2.5 V, V_{tn}=0.3655 V and $\mu_n C_{OX}$= 243.6 μA/V^2. All optimization programs were run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

The parameters of RGA and CRPSO are given in Table 1. For both the RGA and CRPSO, the initial population matrix size for the particle vectors is 10×3 . Rows specify the number of particle vectors in the population and columns specify the dimensions of each particle vector, defined as $x = [C_L, (W/L)_n, t_f]$. So, the dimension of the optimization problem is 3.

1 http://www.mosis.com/pages/Technical/Testdata/tsmc-025-prm

Table 1: RGA and CRPSO parameters for different Case studies.

Parameters	RGA	CRPSO
Population Size	10	10
Dimension of the optimization problem	3	3
Iteration Cycle	250(Case study-1 & Case study-2) 500(Case study-3)	250 (Case study-1 & Case study-2) 500 (Case study-3)
Crossover rate	0.8	-
Crossover	Two Point Crossover	-
Mutation Probability	0.003	-
Mutation	Gaussian Mutation	-
Selection	Roulette	-
Selection Probability	01/03/15	-
C_1	-	2 (Case study-1 & Case study-3) 1.7 (Case study-2)
C_2	-	2 (Case study-1 & Case study-3) 1.7 (Case study-2)
P_{cr}	-	0.3
$v^{craziness}$	-	0.0001

Table 2: Delay limits and design parameters bound for the Case study-1.

Design Set no.	Specified ranges		
	C_L (pF)	(W/L)	t_f (ns)
1	2.4	0.3 - 3.3	0.5 - 6.7
2	0.2-5.6	0.4 - 2.3	0.3 - 6.0
3	0.6 - 3.4	0.9 - 5.0	0.6 - 8.6
4	0.5 - 3.6	1.2 - 4.1	0.9 - 11.0
5	0.7 - 1.8	0.7 - 4.9	1.2 - 15.0
6	0.3 - 2.4	2.2 - 3.2	1.4 - 12.0
7	0.7 - 2.3	0.7 - 3.0	1.6 - 5.7
8	0.6 - 1.9	1.5 - 3.5	1.0 - 8.15

For CRPSO, the control parameters, $C_1, C_2, C_3\ v^{craziness}$ are taken as 2, 2, 0.3, and 0.0001, respectively. The algorithms were run for 250 iterations i.e.ch trial run (total 50 trial runs) with all the design sets, individually and the best results are reported in Table 3. In this case study, estimation of output voltage fall time (t_f) is performed for eight different ranges of design parameters ($C_L, (W/L)_n$) and design criterion (t_f). Specified ranges and RGA, CRPSO and PSO [25] based results are shown in Table 2 and Table 3, respectively.

4.2 Case study-2

In order to achieve a symmetrical switching response, it i.e.pected to have equal rise time (t_r) and fall time (t_f) of output voltage. Due to some second order effects, a definite error between t_r and t_f is always observed. In this case, the main aim is to estimate the design parameters which minimize the difference between t_r and t_f. The design problem can be specified as follows:

Minimize error cost function

$$CF = \left| \left(t_f \left(C_L, \left(\frac{W}{L} \right)_n \right) - t_r \left(C_L, \left(\frac{W}{L} \right)_p \right) - \right) \right| \quad (15)$$

subject to

$$(t_f)_{\min} \le t_f \le (t_f)_{\max} \text{ and } (t_r)_{\min} \le t_r \le (t_r)_{\max},$$

where $(C_L)_{\min} \le C_L \le (C_L)_{\max}$;

$$\left(\left(\frac{W}{L} \right)_n \right)_{\min} \le \left(\frac{W}{L} \right)_n \le \left(\left(\frac{W}{L} \right)_n \right)_{\max};$$

$$\left(\left(\frac{W}{L} \right)_p \right)_{\min} \le \left(\frac{W}{L} \right)_p \le \left(\left(\frac{W}{L} \right)_p \right)_{\max}.$$

The error fitness function is given as

$$J = 10 \log_{10}(CF). \quad (16)$$

The same TSMC 0.25 micron fabrication technology parameters[1] are also used. For both the RGA and CRPSO, the initial size of population matrix for the particle vectors is taken as 10×3 . The number of particles vectors in the population is defined as rows and each column indicates the dimensions denoted as: $x = \left[C_L, \left(\frac{W}{L}\right)_n, \left(\frac{W}{L}\right)_p \right]$. So, the number of optimizing variables is 3.

Table 3: RGA, CRPSO and PSO [25] based results for the Case study-1.

Design Set no.	RGA based results				CRPSO based results				PSO based reported results [25]
	C_L(pF)	(W/L)	t_f (ns)	CF	C_L(pF)	(W/L)	t_f (ns)	CF	t_f (ns)
1	1.2165	1.5458	4.368	2.1652×10^{-15}	0.2322	1.8299	0.7013	0.87138×10^{-15}	1.777
2	1.084	1.1057	5.4413	1.8396×10^{-15}	0.2523	0.8545	1.6935	0.54942×10^{-15}	3.536
3	3.3646	2.3197	8.0324	4.4488×10^{-15}	0.7934	1.9993	2.1987	0.51438×10^{-15}	3.885
4	3.3416	1.712	10.7982	8.9002×10^{-15}	1.1769	2.0055	3.2527	0.19151×10^{-15}	4.627
5	1.6772	1.0914	8.5011	4.6743×10^{-15}	0.9977	2.6862	2.0596	0.50521x 10-15	2.805
6	2.188	2.2381	5.4335	7.8466×10^{-15}	1.0158	2.6909	2.0936	0.73321×10^{-15}	2.719
7	2.2418	2.8412	4.365	6.1363×10^{-15}	0.8947	2.7031	1.8345	0.13551×10^{-15}	2.028
8	1.3954	1.5224	5.062	6.9919×10^{-15}	0.6528	2.2216	1.6282	0.38524×10^{-15}	3.709

For CRPSO, the control parameters C_1, C_2, P_{cr}, $v^{craziness}$ and are taken as 1.7, 1.7, 0.3, and 0.0001, respectively. The algorithms have been run for 250 iteration cycles in each trial run (total 50 trial runs) with all the design sets, individually and the best results are given in Table 5. Delay limits and bounds of design parameters are shown in Table 4. RGA, CRPSO based results and PSO based reported [25, 26] results for each specified range are given in Table 5.

4.3 Case study-3

In Case study-3, the main aim is to achieve a better symmetrical waveform of output voltage for the CMOS inverter, having equal t_r and t_f and equal t_{pHL} and t_{pLH}. RGA and CRPSO algorithms are individually and independently employed to obtain the optimal design parameters which minimize simultaneously the error between t_f and t_r and the error between propagation delay times (tpHL, t_{pLH}) of the output voltage. The design problem can be summarized as follows:

Minimize error cost function

$$CF = \left| \left(t_f \left(C_L, \left(\frac{W}{L} \right)_n \right) \right) - t_r \left(C_L, \left(\frac{W}{L} \right)_p \right) \right| + \left| \left(t_{pHL} \left(C_L, \left(\frac{W}{L} \right)_n \right) \right) - t_{pLH} \left(C_L, \left(\frac{W}{L} \right)_p \right) \right| \quad (17)$$

subject to

$$(t_f)_{\min} \le t_f \le (t_f)_{\max}; (t_r)_{\min} \le t_r \le (t_r)_{\max};$$

$$(t_{pHL})_{\min} \le t_{pHL} \le (t_{pHL})_{\max} \text{ and}$$

$$(t_{pLH})_{\min} \le t_{pLH} \le (t_{pLH})_{\max},$$

where $C_L)_{\min} \le C_L \le (C_L)_{\max}$;

$$\left(\left(\frac{W}{L} \right)_n \right)_{\min} \le \left(\frac{W}{L} \right)_n \le \left(\left(\frac{W}{L} \right)_n \right)_{\max};$$

$$\left(\left(\frac{W}{L} \right)_p \right)_{\min} \le \left(\frac{W}{L} \right)_p \le \left(\left(\frac{W}{L} \right)_p \right)_{\max};$$

The error fitness function is given as

$$J = 10 \log_{10}(CF). \quad (18)$$

Fabrication technology parameters are the same as used in the previous case studies. The dimension of each particle vector is denoted as: $x = \left[C_L, \left(\frac{W}{L}\right)_n, \left(\frac{W}{L}\right)_p \right]$. So, the number of optimizing variables considered here is 3.

For CRPSO, the values of C_1, C_2, P_{cr} and $v^{craziness}$ are taken as 2, 2, 0.3, and 0.0001, respectively. Each algorithm was run with an upper bound of 500 iterations for 50 trial runs with all the design sets, individually and the best results are reported in Table 7. Delay limits, bounds of design parameters and PSO based reported results are shown in Table 6. RGA and CRPSO based results are shown in Table 7.

5 Summary of results and discussion

In this work, two evolutionary optimization algorithms called RGA and CRPSO are employed to achieve the near-global optimal solutions for the switching characteristics of CMOS inverter circuit. PSO based reported results [25, 26] have been taken for the sake of comparison. Three different design cases are considered. For all the case studies, eight different ranges of design parameters and design criteria are considered.

5.1 Discussion on Case study-1

Table 3 shows CRPSO based t_f values are the grand least as compared with RGA and the PSO [25] based t_f values for all the design sets of Case study-1. RGA based t_f values are more than PSO based t_f values for all the design sets of Case study-1. Also the errors (CFs) achieved using the CRPSO algorithm for different design sets are the lowest as compared with those of the RGA based results. Thus, the proposed CRPSO has proven to be the best near-global optimizer in this case study.

Figure 4 shows the plot of J versus iteration cycle for the seventh design set of Case study-1. For this seventh design set of Case study-1, RGA results in $J = -142.1209$ dB and execution time required by RGA is 3.767 s in 61 iteration cycles, whereas, CRPSO results in $J = -158.6803$ dB and execution time taken by CRPSO is 2.594 s in 51 iteration cycles. So, CRPSO proves to be better and faster than RGA. For the same design set, CRPSO yields optimal $C_L = 0.8947$ pF and $(W/L)_n = 2.7031$ after final convergence.

5.2 Discussion on Case study-2

CRPSO based t_f and t_r values are the least as compared with RGA and PSO based results [25, 26] for all the design sets of Case study-2. RGA based t_f and t_r values are more than PSO based reported results for all the design sets of Case study-2. The CF values obtained using the CRPSO algorithm for different design sets are the lowest as com-

Table 4: Delay limits and design parameters bound for the Case study-2.

Design set no.	Specified ranges				
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)
1	0.33-2.3	1-3	2-18	1-12	1-12
2	0.6-1.5	0.5-2.5	1.6-19.3	0.5-7.6	0.5-7.6
3	0.3-3	0.3-1.9	1.76-7.65	0.56-8.7	0.56-8.7
4	0.11-1.34	1.5-3.5	2.65-18.9	0.77-7.89	0.77-7.89
5	0.5-1.5	1-2.5	2-13.75	0.1-15	0.1-15
6	0.5-1.5	1-3	2-21	0.1-15	0.1-15
7	1.0-3.0	1.5-3.5	3.75-21	0.1-15	0.1-15
8	1.5-3.5	1.5-3	3-19.2	0.1-10	0.1-10

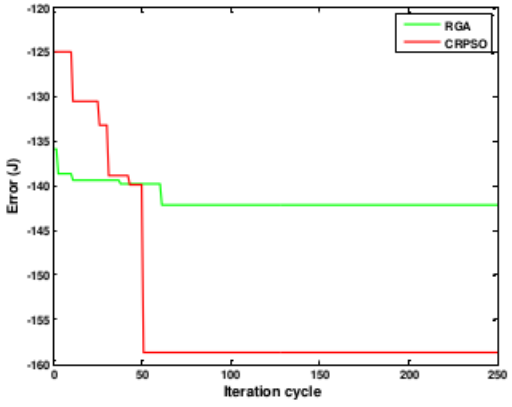

Figure 4: Plot of J versus iteration cycle for the seventh design set of Case study-1.

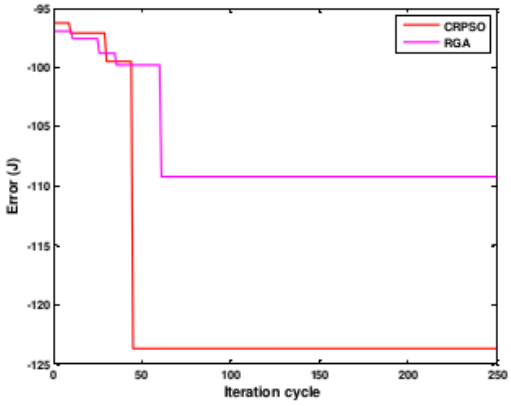

Figure 5: Plot of J versus iteration cycle for the fourth design set of Case study-2.

pared with those of RGA based results. Thus, the proposed CRPSO has again proven to be the best near-global optimizer in this case study also. Table 5 shows all these results of Case sudy-2.

Figure 5 shows the plot of J versus iteration cycle for the fourth design set of Case study-2. For this fourth design set of Case study-2, RGA yields $J = -109.2442$ dB with an execution time 3.871 s in 61 iteration cycles, whereas, CRPSO results in $J = -123.7221$ dB with an execution time =2.719 s in 45 iteration cycles. So, CRPSO is better and faster than RGA. For the same design set, CRPSO yields final optimal convergent values of C_L, $\left(\frac{W}{L}\right)_n$, as 0.3702 pF, 2.4674, and 13.4464, respectively.

5.3 Discussion on Case study-3

As shown by Tables 6 and 7, CRPSO produces the grand best symmetric output waveforms with the least values of t_r, t_f, t_{pHL}, t_{pLH} for the CMOS inverter as compared with RGA and PSO [25] for all the design sets of Case study-3. RGA based t_f, t_r and t_{pHL}, t_{pLH} values are more than PSO based reported results [25] for all the design sets of Case study-3. Also, the CF values obtained using the CRPSO algorithm for different design sets are the lowest compared with those of RGA based results. Thus, the proposed CRPSO proves to be the best near-global optimizer for this Case study-3.

Figure 6 shows the plot of J versus iteration cycle for the eighth design set of Case study-3. For thi.e.ghth design set, RGA results in $J = -103.2900$ dB with an execution time = 4.691 s in 41 iteration cycles, whereas, CRPSO results in $J = -112.7906$ dB with an execution time as 3.313 s in 30 iteration cycles.

Figure 7 shows the plot of C_L versus iteration cycle for the eighth design set of CRPSO in Case study-3. For the first six iteration cycles, C_L remains same at 0.5094 pF. Then, C_L increases slightly. At the 10th iteration cycle the value of C_L changes to 2.0933 pF and remains the same up to 19th iteration cycle. After that, C_L decreases slightly. At the 30th iteration cycle C_L becomes 0.3483 pF and it remains fixed up to 500 iteration cycles.

Table 5: RGA, CRPSO and PSO [25, 26] based results for the Case study-2.

Design set no.	RGA based results						CRPSO based results						PSO based reported results [25, 26]	
	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)	Error CF (ps)	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)	Error CF(ps)	t_f (ns)	t_r (ns)
1	1.2874	1.5557	8.4922	4.5367	4.5267	10.031	0.7337	2.8958	15.7513	1.389	1.3909	1.9023	1.86	1.86
2	1.2577	1.0886	5.9107	6.3333	6.3535	20.231	0.7309	2.085	11.3273	1.9217	1.9266	4.9709	2.31	2.31
3	1.9176	1.2087	6.6391	8.6975	8.6244	73.11	0.6209	0.6056	3.296	5.6207	5.6253	4.6023	7.07	7.07
4	1.3297	2.6852	14.6899	2.7148	2.7029	11.901	0.3702	2.4674	13.4464	0.82261	0.82218	0.42441	0.87	0.87
5	1.4851	1.6222	8.8055	10.038	10.072	34.404	0.6424	1.0541	5.7484	3.3409	3.3368	4.136	3.77	3.78
6	1.4934	1.0106	5.4471	8.1016	8.1866	85.044	0.826	1.1319	6.1694	4.0006	3.9981	2.5929	6.18	6.19
7	2.9905	1.5011	8.1406	10.921	10.969	47.963	1.3703	1.59	8.6485	4.7247	4.7311	6.4191	5.54	5.54
8	3.4164	1.8942	10.306	9.8874	9.8983	10.896	1.6099	1.5722	8.5579	5.6134	5.617	3.6085	7.79	7.79

Table 6: Delay limits, design parameters bound and PSO [25] reported results for the Case study-3.

Design set no.	Specified ranges							PSO based reported results [25]			
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)	t_{pHL} (ns)	t_{pLH} (ns)	t_f(ns)	t_r(ns)	t_{pHL} (ns)	t_{pLH} (ns)
1	0.2- 4	1.1- 6.1	2.8 -19.3	1.1 - 13	1.1 - 13	0.5 - 10	0.5 - 10	1.18	1.18	0.55	0.5
2	0.1 -5.1	1.6 - 7.1	1.8 - 18	1.1 - 15	1.1- 15	0.5 - 8	0.5 - 8	1.18	1.18	0.55	0.5
3	0.47 - 2	1.4 - 6.7	3.2 - 38	0.5 - 12	0.5 - 12	0.2 - 9	0.2 - 9	0.5	0.5	0.23	0.21
4	0.1-1.1	1.2 - 7	1.5-17.5	0.5- 5	0.5- 5	0.2- 4	0.2 - 4	0.5	0.5	0.23	0.21
5	0.2 - 14	1.9 - 5.0	2.7 - 17	0.7 - 6	0.7 - 6	0.4 - 5	0.4 - 5	0.94	0.94	0.44	0.4
6	0.3 -3.6	1.3 - 3.5	3.5 - 16.2	0.25 - 7	0.25 - 7	0.3 - 4.5	0.3 - 4.5	0.78	0.78	0.36	0.33
7	0.2 -4.9	1.1 - 5.8	2.2 -25.3	0.5 - 6.6	0.5 - 6.6	0.2 - 7.7	0.2 - 7.7	0.53	0.53	0.25	0.23
8	0.2 -3.5	0.3 - 7.6	1.3 - 39	0.3 - 6.6	0.3 - 6.6	0.1 - 4.4	0.1 - 4.4	0.32	0.32	0.15	0.14

Table 7: RGA and CRPSO based results for the Case study-3.

Design set no.	RGA based results								CRPSO based results							
	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	t_{pHL} (ns)	t_{pLH} (ns)	Error CF (ps)	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	t_{pHL} (ns)	t_{pLH} (ns)	Error CF (ps)
1	1.608	3.4587	18.7995	2.5487	2.554	1.0814	1.1235	47.524	0.7347	3.4142	18.5973	1.1796	1.1796	0.50049	0.5189	18.41
2	1.2625	2.7157	14.8586	2.5485	2.537	1.0813	1.1161	46.239	0.4042	1.8798	10.2903	1.1788	1.1729	0.50014	0.51596	21.74
3	1.2872	6.0924	34.4854	1.1583	1.1146	0.49144	0.49031	44.836	0.615	6.6992	36.7252	0.50326	0.50002	0.21352	0.21996	9.6801
4	0.6901	2.3398	12.649	1.6169	1.6291	0.68601	0.71666	42.862	0.1924	2.0854	11.4188	0.50577	0.5031	0.21459	0.22132	9.3983
5	0.7487	2.2411	12.4224	1.8316	1.7997	0.77709	0.79172	46.441	0.3719	2.1614	11.877	0.9432	0.93491	0.40018	0.41127	19.386
6	0.8194	2.213	12.2291	2.0298	2.0007	0.8612	0.88011	48.033	0.3655	2.6324	14.3402	0.76112	0.76102	0.32293	0.33478	11.952
7	1.229	4.1733	23.2164	1.6144	1.588	0.68495	0.69539	44.063	0.4053	4.4372	23.6604	0.5111	0.51149	0.21685	0.22501	8.5498
8	2.2245	6.2761	34.7111	1.9431	1.9136	0.8244	0.8418	46.881	0.3483	6.3391	34.484	0.3012	0.30158	0.12779	0.13267	5.2594

Table 8: *t*-values between RGA and CRPSO for different Case studies over 50 runs.

CF values	Case study-1		Case study-2		Case study-3	
	RGA (2nd design set)	CRPSO (7th design set)	RGA (1st design set)	CRPSO (4th design set)	RGA (4th design set)	CRPSO (8th design set)
Minimum	1.8396×10^{-15}	0.13551×10^{-15}	10.031×10^{-12}	0.42441×10^{-12}	42.862×10^{-12}	5.2594×10^{-12}
Maximum	6.1976×10^{-15}	0.99514×10^{-15}	40.926×10^{-12}	2.9132×10^{-12}	85.147×10^{-12}	15.3432×10^{-12}
Mean	3.0904×10^{-15}	0.57005×10^{-15}	25.044×10^{-12}	1.6015×10^{-12}	64.201×10^{-12}	10.0572×10^{-12}
Standard deviation	2.2598×10^{-15}	0.10178×10^{-15}	19.591×10^{-12}	0.35412×10^{-12}	49.963×10^{-12}	1.0815×10^{-12}
t-value	7.8784 for CRPSO		8.4598 for CRPSO		7.6610 for CRPSO	

Table 9: PSPICE based results versus RGA based results for the Case study-1.

Design set no.	PSPICE inputs		PSPICE results	RGA based result
	C_L(pF)	(W/L)	t_f (ns)	t_f (ns)
1	1.2165	1.5458	7.1999	4.368
2	1.084	1.1057	8.567	5.4413
3	3.3646	2.3197	13.871	8.0324
4	3.3416	1.712	17.745	10.7982
5	1.6772	1.0914	13.385	8.5011
6	2.188	2.2381	9.3984	5.4335
7	2.2418	2.8412	7.9902	4.365
8	1.3954	1.5224	8.3109	5.062

Table 10: PSPICE based results versus CRPSO based results for the Case study-1.

Design set no.	PSPICE inputs		PSPICE results	CRPSO based results
	C_L(pF)	(W/L)	t_f (ns)	t_f (ns)
1	0.2322	1.8299	1.2572	0.7013
2	0.2523	0.8545	2.6021	1.6935
3	0.7934	1.9993	3.7531	2.1987
4	1.1769	2.0055	5.5341	3.2527
5	0.9977	2.6862	3.7109	2.0596
6	1.0158	2.6909	3.7691	2.0936
7	0.8947	2.7031	3.3395	1.8345
8	0.6528	2.2216	2.8454	1.6282

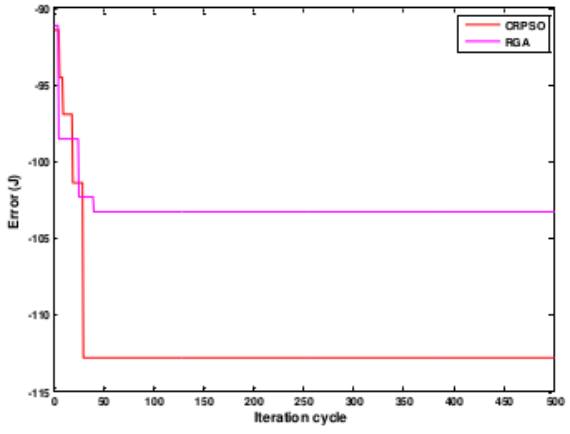

Figure 6: Plot of J versus iteration cycle for the eighth design set for Case study-3.

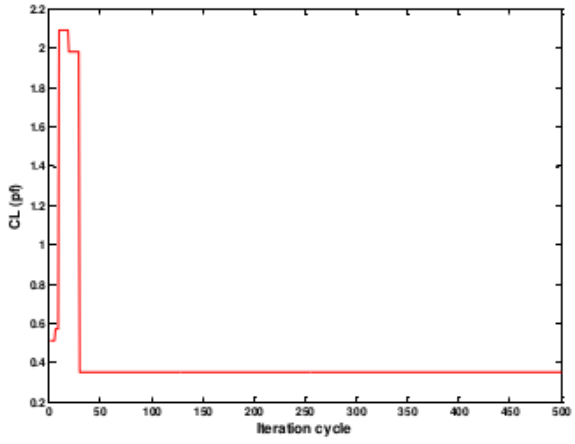

Figure 7: Plot of C_L versus iteration cycle for the eighth design set of CRPSO for Case study-3.

Table 11: PSPICE based results versus RGA based results for the Case study-2.

Design set no.	PSPICE inputs			PSPICE results			RGA based results		
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	Error(ns)	t_f(ns)	t_r(ns)	Error (ps)
1	1.2874	1.5557	8.4922	7.5624	6.6399	0.9225	4.5367	4.5267	10.031
2	1.2577	1.0886	5.9107	10.095	9.006	1.089	6.3333	6.3535	20.231
3	1.9176	1.2087	6.6391	13.87	12.375	1.495	8.6975	8.6244	73.11
4	1.3297	2.6852	14.6899	4.9882	3.9594	1.0288	2.7148	2.7029	11.901
5	1.4851	1.6222	8.8055	16.603	14.513	2.09	10.038	10.072	34.404
6	1.4934	1.0106	5.4471	12.827	11.668	1.159	8.1016	8.1866	85.044
7	2.9905	1.5011	8.1406	17.817	15.745	2.072	10.921	10.969	47.963
8	3.4164	1.8942	10.306	16.722	14.234	2.488	9.8874	9.8983	10.896

Table 12: PSPICE based results versus CRPSO based results for the Case study-2.

Design set no.	PSPICE inputs			PSPICE results			CRPSO based results		
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	Error(ns)	t_f(ns)	t_r(ns)	Error (ps)
1	0.7337	2.8958	15.7513	2.4694	2.1351	0.3343	1.389	1.3909	1.9023
2	0.7309	2.085	11.3273	3.2602	2.8769	0.3833	1.9217	1.9266	4.9709
3	0.6209	0.6056	3.296	8.7738	7.9554	0.8184	5.6207	5.6253	4.6023
4	0.3702	2.4674	13.4464	1.5339	1.3959	0.138	0.82261	0.82218	0.42441
5	0.6424	1.0541	5.7484	5.3739	4.8052	0.5687	3.3409	3.3368	4.136
6	0.826	1.1319	6.1694	6.4509	5.7812	0.6697	4.0006	3.9981	2.5929
7	1.3703	1.59	8.6485	7.841	6.8851	0.9559	4.7247	4.7311	6.4191
8	1.6099	1.5722	8.5579	8.7224	7.9136	0.808	5.6134	5.617	3.6085

Table 13: PSPICE based results versus RGA based results for the Case study-3.

Design set no.	PSPICE inputs			PSPICE results					RGA based result				
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	t_{pHL}(ns)	t_{pLH}(ns)	Error (ns)	t_f(ns)	t_r(ns)	"t_{pHL} (ns)"	"t_{pLH} (ns)"	Error (ps)
1	1.608	3.4587	18.7995	5.0402	3.842	2.6045	1.5905	2.2122	2.5487	2.554	1.0814	1.1235	47.524
2	1.2625	2.7157	14.8586	4.6814	3.6916	2.4359	1.6852	1.7405	2.5485	2.537	1.0813	1.1161	46.239
3	1.2872	6.0924	34.4854	3.3885	1.7639	1.9868	0.89849	2.71291	1.1583	1.1146	0.49144	0.49031	44.836
4	0.6901	2.3398	12.649	2.8627	2.4943	1.5716	1.1932	0.7468	1.6169	1.6291	0.68601	0.71666	42.862
5	0.7487	2.2411	12.4224	3.2725	2.7016	1.7663	1.3098	1.0274	1.8316	1.7997	0.77709	0.79172	46.441
6	0.8194	2.213	12.2291	3.6087	2.9753	1.926	1.4444	1.115	2.0298	2.0007	0.8612	0.88011	48.033
7	1.229	4.1733	23.2164	4.6131	2.4104	2.6348	1.2041	3.6334	1.6144	1.58808	0.68495	0.69539	44.063
8	2.2245	6.2761	34.7111	5.6578	2.8724	3.1773	1.4272	4.5355	1.9431	1.9136	0.8244	0.8418	46.881

Table 14: PSPICE based results versus CRPSO based results for the Case study-3.

Design set no.	PSPICE inputs			PSPICE results					CRPSO based result				
	C_L(pF)	$(W/L)_n$	$(W/L)_p$	t_f(ns)	t_r(ns)	t_{pHL} (ns)	t_{pLH} (ns)	Error (ns)	t_f(ns)	t_r(ns)	t_{pHL} (ns)	t_{pLH} (ns)	Error (ps)
1	0.7347	3.4142	18.5973	2.2711	1.9555	1.2873	0.7992	0.8037	1.1796	1.1796	0.50049	0.5189	18.41
2	0.4042	1.8798	10.2903	2.0744	1.793	1.1801	0.97329	0.48821	1.1788	1.1729	0.50014	0.51596	21.74
3	0.615	6.6992	36.7252	1.3011	0.97988	0.83292	0.48345	0.67069	0.50326	0.50002	0.21352	0.21996	9.6801
4	0.1924	2.0854	11.4188	1.0023	0.96108	0.64678	0.48286	0.20514	0.50577	0.5031	0.21459	0.22132	9.3983
5	0.3719	2.1614	11.877	1.7178	1.5243	1.0063	0.7546	0.4452	0.9432	0.93491	0.40018	0.41127	19.386
6	0.3655	2.6324	14.3402	1.4414	1.3426	0.88577	0.59809	0.38648	0.76112	0.76102	0.32293	0.33478	11.952
7	0.4053	4.4372	23.6604	1.1333	0.98507	0.72991	0.49755	0.38059	0.5111	0.51149	0.21685	0.22501	8.5498
8	0.3483	6.3391	34.484	0.93346	0.87646	0.6304	0.37537	0.31203	0.3012	0.30158	0.12779	0.13267	5.2594

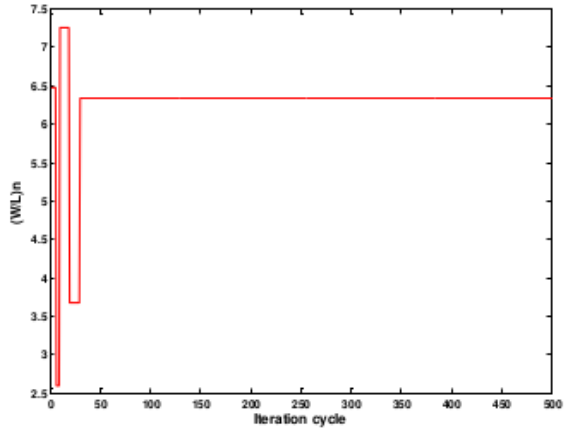

Figure 8: Plot of $(W/L)_n$ versus iteration cycle for the eighth design set of CRPSO for Case study-3.

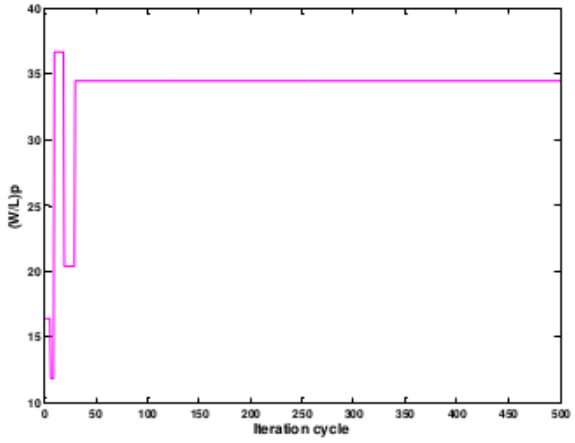

Figure 9: Plot of $(W/L)_p$ versus iteration cycle for the eighth design set of CRPSO for Case study-3.

Figure 8 shows the plot of $(W/L)_n$ versus iteration cycle for the eighth design set of CRPSO in Case study-3. The value of $(W/L)_n$ is constant at 6.4684 for the first six iteration cycles. Then a sharp fall of $(W/L)_n$ is noticed. From iteration cycle 10 to 19, $(W/L)_n$ remains fixed at 7.2584. After that, a quick fall is observed. At the 30th iteration cycle, $(W/L)_n$ becomes 6.3391 and it remains constant up to 500 iteration cycles.

Figure 9 shows the plot of $(W/L)_p$ versus iteration cycle for the eighth design set of CRPSO in Case study-3. For the first six iteration cycles, $(W/L)_p$ is fixed at 16.3384. Then $(W/L)_p$ is decreased. From iteration cycle 10 to 19, $(W/L)_p$ is constant at 36.6374. After that, a sharp fall is observed. At 30th iteration cycle, $(W/L)_p$ becomes 34.4840 and it is fixed up to maximum iteration cycles.

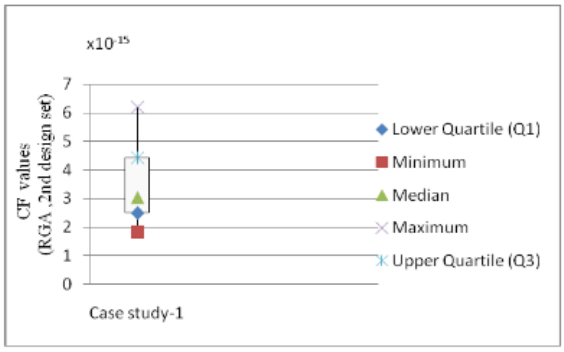

Figure 10: Box and whisker plots of RGA for the second design set of Case study-1 over 50 trial runs.

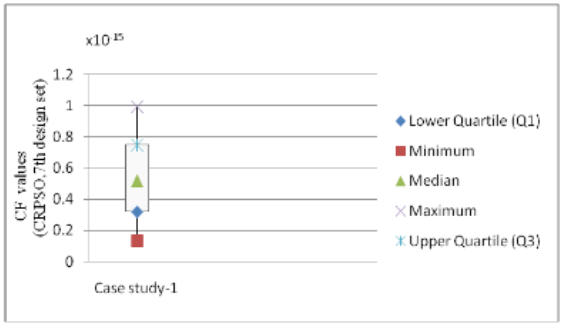

Figure 11: Box and whisker plots of CRPSO for the seventh design set of Case study-1 over 50 trial runs.

5.4 Discussion on compartive Box and Whisker plots

RGA and CRPSO have been individually run for 50 $t_r$ial runs for each design set of all the case studies and the resulting CF values obtained in each run have been used for Box and Whisker plots. Figures 10-15 show the Box and Whisker plots of the best design sets of RGA and CRPSO for all the case studies, respectively. Upper and lower ends of boxes represent the 75th and 25th percentiles. Median is represented by the green colour. The whiskers are the lines extending from each end of the boxes to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers.

From Figures 10-15, it in is evident that the lowest value of CF obtained by CRPSO is lower than the CF obtained using RGA for all the Case studies. The median of the CF values obtained by the CRPSO is lower than that of RGA. So, CRPSO performs more stably.

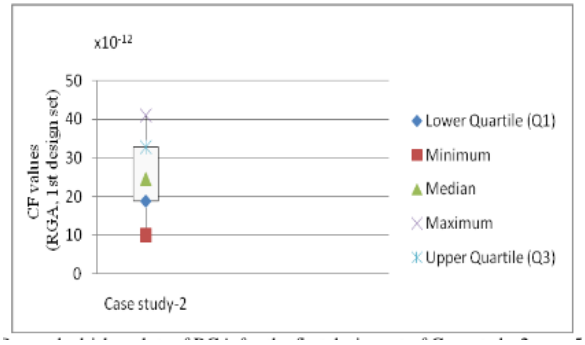

Figure 12: Box and whisker plots of RGA for the first design set of Case study-2 over 50 trial runs.

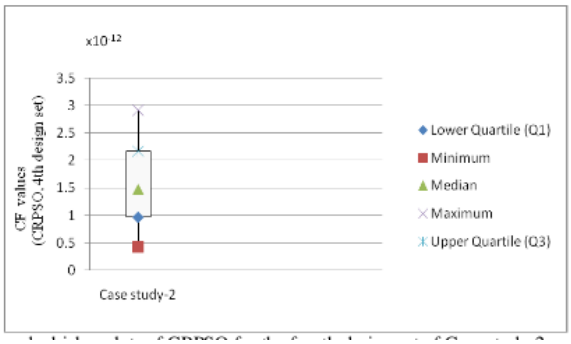

Figure 13: Box and whisker plots of CRPSO for the fourth design set of Case study-2 over 50 trial runs.

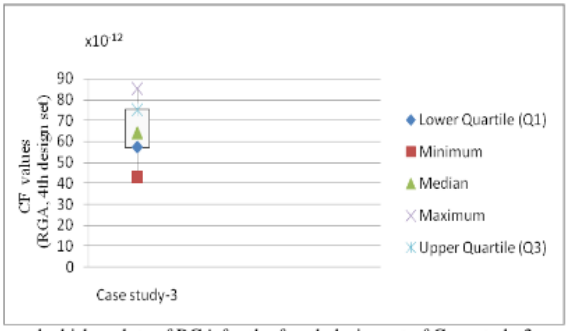

Figure 14: Box and whisker plots of RGA for the fourth design set of Case study-3 over 50 trial runs.

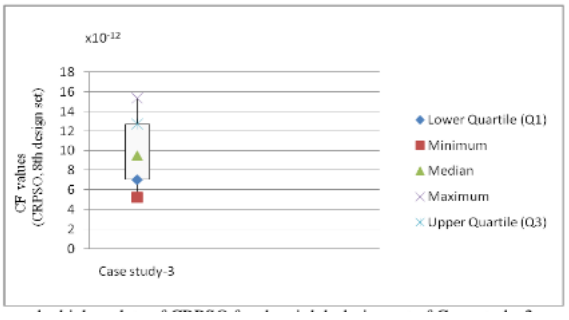

Figure 15: Box and whisker plots of CRPSO for the eighth design set of Case study-3 over 50 trial runs.

5.5 Discussion on compartive *t*-values

The *t*-values between the best design sets of RGA and CRPSO for different Case studies are shown in Table 8. The *t*-values of all the Case studies are larger than 2.15 (degree of freedom = 49), which means that there is a significant difference between RGA and CRPSO with a 98% confidence level. Thus, from statistical analysis, it is clear that the CRPSO based optimization technique is a much better algorithm than RGA; CRPSO offers more robust and promising results.

5.6 Discussion on validation of results by PSPICE

To validate the results obtained through RGA and CRPSO optimizations, the inverters are redesigned using PSPICE for each design set considering the respective optimal values of output load capacitor and transistor dimensions as inputs. PSPICE based results are shown in Tables 9–14, respectively. The dissimilarity between PSPICE based results and RGA/CRPSO based design results specially for Case studies 2 and 3 occur from the fact that PSPICE evaluates the rise time, fall time and propagation delay times using more complex circuit equation sets. Whereas, the delay expressions, used in RGA and CRPSO based designs are very simple and derived from the simple current–voltage relationships of long-channel transistors. So, the effect of channel velocity saturation and small-geometry effects of transistor are not considered. Thus, PSPICE based design result in greater delay times as compared with RGA and CRPSO based inverter designs.

6 Conclusion

In this work the evolutionary algorithms like real code.gnetic algorithm (RGA) and a highly modified version of PSO called craziness based particle swarm optimization (CRPSO) are utilized to achieve the optimal switching characteristics of CMOS inverter. RGA and CRPSO algorithms are executed, individually and independently, to three different inverter design cases with different ranges of design parameters. The PSO based reported results are also given. The proposed CRPSO algorithm has established its efficiency in finding the grand least errors for all the design cases. As compared to RGA based computed results and PSO based reported results, CRPSO yields the best symmetric output waveform of the designed CMOS inverter

with equal rise time and fall time and equal propagation delay times.

Nomenclature

CMOS - Complementary Metal Oxide Semiconductor
VLSI - Very Large Scale Integration
RGA - Real coded Genetic Algorithm
PSO - Particle Swarm Optimization
CRPSO- Craziness based Particle Swarm Optimization
NMOS - n-type Metal Oxide Semiconductor
PMOS - p-type Metal Oxide Semiconductor
FIR - Finite Impulse Response
PSPICE - Personal computer Simulation Program with Integrated Circuit Emphasis
TSMC - Taiwan Semiconductor Manufacturing Company Limited
C_L - Output load capacitance
$(W/L)_n$ - Aspect ratio of NMOS transistor
$(W/L)_p$ - Aspect ratio of PMOS transistor
μ_n - Mobility of electron
μ_p - Mobility of hole
V_{tn}-Threshold voltage of NMOS transistor
V_{tp} - Threshold voltage of PMOS transistor
V_{DD} - Supply voltage
C_{OX} - Oxide capacitance per unit area
t_f - Fall time (time required for the output voltage to drop from V_{90}% level to V_{10}% level)
t_r - Rise time (time required for the output voltage to rise from V_{10}% level to V_{90}% level)
t_{pHL} - Propagation delay for HIGH to LOW transition (time delay between the V_{50}% transition of the rising input voltage and V_{50}% transition of the falling output voltage)
t_{pLH} - Propagation delay, for LOW to HIGH transition (time delay between the V_{50}% transition of the falling input voltage and V_{50}% transition of the rising output voltage)
CF - Cost Function

References

[1] Ma Q., Cowan C.F.N., Genetic algorithms applied to the adaptation of IIR filters, Signal Process., 1996, 48, 155–163.
[2] Luitel B., Venayagamoorthy G.K., Particle swarm optimization with quantum infusion for system identification, Eng. Appl. Artif. Intel., 2010, 23, 635–649.
[3] Kar R., Mandal D., Mondal S., Ghoshal S.P., Craziness based Particle Swarm Optimization Algorithm for FIR Band Stop Filter Design, Swarm Evol. Comput., 2012, 7, 58–64.
[4] Hussain Z.M., Sadik A.Z., O'Shea P., Digital Signal Process.-An Introduction with MATLAB Applications, Springer-Verlag, 2011.
[5] Fang W., Sun J., Xu W., A new mutated quantum behaved particle swarm optimizer for digital IIR filter design, EURASIP J. Adv. Signal. Process., DOI:10.1155/2009/367465
[6] Krusienski D.J., Jenkins W.K., Adaptive filtering via particle swarm optimization, Proceedings of 37th Asilomar Conference on Signals, Systems and Computers (9-12 November 2003), 2003, 571–575.
[7] Mandal S., Ghoshal S.P., Kar R., Mandal D., Design of Optimal Linear Phase FIR High Pass Filter using Craziness based Particle Swarm Optimization Technique, Journal of King Saud University-Computer and Information Sciences, 2012, 24, 83–92.
[8] Saha S.K., Kar R., Mandal D., Ghoshal S.P., IIR filter design with craziness based particle swarm optimization technique, World. Acad. Sci. Eng. Technol., 2011, 5, 1052–1059.
[9] Holland J.H., Adaptation in Natural and Artificial Systems, MIT Press Cambridge, MA, USA, 1975.
[10] Mastorakis N.E., Gonos I.F., Swamy M.N.S., Design of Two Dimensional Recursive Filters Using Genetic Algorithms, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 2003, 50, 634–639.
[11] Lu H.C., Tzeng S.T., Design of arbitrary FIR log filters by genetic algorithm approach, Signal Process., 2000, 80, 497–505.
[12] Das A., Vemuri R., An Automated Passive Analog Circuit Synthesis Framework using Genetic Algorithms, IEEE Computer Society Annual Symposium on VLSI 7 (9-11 March 2007), 2007, 145–152.
[13] Gill S.S., Chandel R., Chandel A., Comparative study of Ant Colony and Genetic Algorithms for VLSI circuit partitioning, World. Acad. Sci. Eng. Technol., 2009, 28, 890–894.
[14] Tang M., Lau R.Y.K., A Parallel Genetic Algorithm for Floor plan Area Optimization, 7th International Conference on Intelligent Systems Design and Application (20-24 October 2007, Rio de Janeiro, Brazil), 2007, 801–806.
[15] Kennedy J., Eberhart R., Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks (Nov/Dec 1995, Perth, WA), 1995, 1942–1948.
[16] Eberhart R., Shi Y., Comparison between genetic algorithm and particle swarm optimization, Evolutionary Programming-VII, 1998, 1447, 611–616.
[17] Chen S., Luk B.L., Digital IIR filter design using particle swarm optimization, Int. J. Modelling, Identification and Control, 2010, 9, 327–335.
[18] Gudise V.G., Venayagamoorthy G.K., FPGA placement and routing using particle swarm optimization, IEEE Computer Society Annual Symposium on VLSI (19-20 February 2004), 2004, 307–308.
[19] Lai C., A novel image segmentation approach based on particle swarm optimization, IEICE Trans. Fundamentals, 2006, E89-A, 324–327.
[20] Ulker S., Particle swarm optimization applications to microwave circuits, Microw. Opt. Technol. Lett., 2008, 50, 1333–1336.
[21] Vural R. A., Yildirim T., Analog circuit sizing via swarm intelligence, Int. J. Electron. Commun. (AEÜ), 2012, 66, 732– 740
[22] Ling S.H., Iu H.H.C., Leung F.H.F., Chan K.Y., Improved hybrid particle swarm optimized wavelet neural network for modelling the development of fluid dispensing for electronic packaging, IEEE Trans. Ind. Electron., 2008, 55, 3447–3460.

[23] Biswal B., Dash P.K., Panigrahi B.K., Power quality disturbance classification using fuzzy c-means algorithm and adaptive particle swarm optimization, IEEE Trans. Ind. Electron., 2009, 56, 212–220.

[24] Saha S.K., Kar R., Mandal D., Ghoshal S.P., An Efficient Craziness Based Particle Swarm Optimization Technique for Optimal IIR Filter Design, Transactions on Computational Science XXI, 2013, 8160, 230–252.

[25] Vural R.A., Der O., Yildirim T., Investigation of particle swarm optimization for switching characterization of inverter design, Expert. Syst. Appl., 2011, 38, 5696–5703.

[26] Vural R.A., Der O., Yildirim T., Particle swarm optimization based inverter design considering transient performance, Digital Signal Process., 2010, 20, 1215–1220.

[27] Mukhopadhyay J., Pandit S., Modeling and Design of a Nano Scale CMOS Inverter for Symmetric Switching Characteristics, VLSI Design, DOI:10.1155/2012/505983.

[28] Karaboga N., Digital IIR filter design using differential evolution algorithm, EURASIP J. Adv. Signal. Process., DOI:10.1155/ASP.2005.1269.

[29] Karaboga N., A new design method based on artificial bee colony algorithm for digital IIR filters, J. Franklin Inst., 2009, 346, 328–348.

[30] Hardel R.G., Mandal D., Ghoshal S.P., Kar R., Minimization of side lobe of optimized uniformly spaced and non-uniform excited time modulated linear antenna arrays using genetic algorithm, Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science, 2012, 7677, 451–458.

[31] Mandal D., Ghoshal S. P., Bhattacharjee A. K., Radiation Pattern Optimization for Concentric Circular Antenna Array With Central Element Feeding Using Craziness Based Particle Swarm Optimization,Int. J. RF Microw. C. E. , 2010, 20, 577–586.

[32] De B.P., Kar R., Mandal D., Ghoshal S.P., Optimal analog active filter design using craziness-based particle swarm optimization algorithm, Int. J. Numer. Model., DOI:10.1002/jnm.2040

[33] Walpole R.E., Myers R.H., Probability and statistics for engineers and scientists, Macmillan Publishing Co., Inc., New York, 1978.

[34] DeMassa T.A., Ciccone Z., Digital Integrated Circuits, John Wiley & Sons, New York, 1996.

[35] Kang S.M., Leblebici Y., CMOS Digital Integrated Circuits Analysis and Design, TMH, India, 2003.