

# Symmetrical and High-Speed CMOS Inverter Design Using Hybrid PSO-GSA Optimization Algorithm

V. P. JAY FANTIN\*, G. MOHANBABU

**Abstract:** Complementary Metal-Oxide-Semiconductor (CMOS) inverters are fundamental building blocks in digital integrated circuits (ICs), where achieving high-speed operation and symmetrical switching characteristics remains a significant design challenge. Traditional optimization methods often fall short in balancing the trade-offs between performance parameters such as rise time, fall time, and propagation delay. This paper presents a novel approach using a hybrid metaheuristic algorithm, Particle Swarm Optimization combined with the Gravitational Search Algorithm (PSOGSA), to optimize CMOS inverter design parameters effectively. The proposed method leverages the global search capability of PSO and the local refinement ability of GSA to enhance both convergence speed and solution quality. The research explores eight sets of design parameter constraints under three key optimization cases: (i) minimizing fall time, (ii) achieving symmetrical rise time ( $t_r$ ) and fall times ( $t_f$ ) and (iii) balancing propagation delays like high to low ( $t_{pHL}$ ) and low to high ( $t_{pLH}$ ). The optimized inverter circuit was implemented in the 45 nm technology in cadence EDA tool. The performance of PSOGSA-optimized inverters is benchmarked against results from PSpice simulations and seven conventional optimization techniques. Simulation results reveal that the proposed PSOGSA method achieves propagation delays in the range of 0.54-0.63 ns while maintaining switching symmetry between 97% and 99%. These outcomes highlight the algorithm's effectiveness in delivering robust, high-performance CMOS inverter designs with enhanced switching uniformity. The proposed methodology offers a promising optimization strategy for future low-power and high-speed VLSI applications.

**Keywords:** CMOS inverter design; propagation delay; PSOGSA algorithm; switching symmetry; VLSI optimization

## 1 INTRODUCTION

The VLSI technology has miniaturized the digital devices we use to perform many daily activities and complex tasks. The inverter is the most basic digital circuit whose parameters are used as a reference for designing complex circuits. So, designing a better-performing inverter paves the way to designing efficient digital systems. The major bottlenecks that limit the performance of an inverter are power consumption and symmetric switching. A power-efficient inverter can be designed using CMOS technology. Designing a CMOS inverter with symmetric switching is highly time-consuming as it involves many iterations [1].

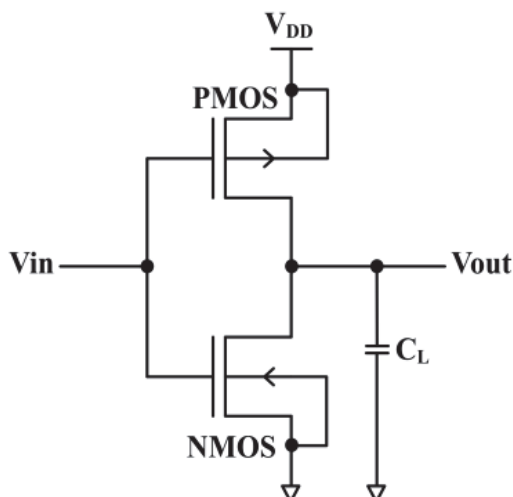


Figure 1 Schematic of CMOS Inverter

The Fig. 1 shows a Complementary MOS technology based inverter which is made up of a PMOS and NMOS transistor, as shown in Fig. 1. The input voltage ( $V_{in}$ ) is applied to the control terminal called as gate terminal of PMOS and NMOS. The supply voltage ( $V_{DD}$ ) is linked to the source node of PMOS, and the source terminal of

N-MOS is connected to the ground. The PMOS and NMOS drain terminals are linked to the output load capacitor ( $C_L$ ). The output ( $V_{out}$ ) is measured across the load capacitor. The fourth terminal, the body or base of NMOS, is connected to the ground alongside its source, and the PMOS base terminal is connected to the supply voltage alongside its source. The CMOS inverter is a circuit that generates an output which is the inversion of the incoming signal. The symmetry among the various sets of delays associated with the output signal can be achieved by minimizing the variation between the propagation delay high value of the signal to the low value of the signal ( $t_{pHL}$ ) and propagation delay low value of the signal to the high value of the signal ( $t_{pLH}$ ) and also the variance among the fall time ( $t_f$ ) and rise time ( $t_r$ ). The timing parameters of the CMOS inverter depend on the length of the channel ( $L$ ), width of the channel ( $W$ ), turn-on voltage also called threshold voltage ( $V_t$ ), carrier mobility ( $\mu$ ), supply voltage ( $V_{DD}$ ) and load capacitance ( $C_L$ ) values [2]. Among these, the three parameters are variable parameters: length, width and load capacitance. A symmetric switching CMOS inverter can be designed by finding the best prime value of the three parameters. But, finding the best prime value is a very complex iterative process requiring a lot of time. Many optimization algorithms and techniques have been employed to find the best possible width length ratio and load capacitance value.

In research papers [3, 4], the authors used the PSO algorithm to improve the CMOS inverter's input-output response. The paper [5, 6] uses an artificial neural network with PSO to improve the CMOS inverter's switching. In [7], the authors used PSO to make the switching symmetrical and checked the results with computer simulations. In [8], PSO combined with a constriction factor and inertia weight approach (PSOCFIWA) is examined to determine switching characteristics. They compared it with results from a standard genetic algorithm called real genetic algorithm (RGA). In [9], the differential evolution (DE) algorithm and RGA is used separately to

make an optimized inverter, looking at different cost functions. They found DE had a lower cost function than RGA. In [10], the authors suggested a new PSO version called craziness-based particle swarm optimization (CRPSO) for an optimal inverter circuit and it has better-switching symmetry than RGA. In [11], they combined harmony search (HS) algorithm and DE algorithm (HS-DE) and compared it with PSO and PSpice results. In [10], they used the aging leader and challenger PSO (ALC-PSO) algorithm to design an optimal CMOS inverter, comparing it with RGA. In [12, 13], they used a cuckoo search algorithm (CSA) to reduce differences in rise and fall times. They compared the timing parameters with PSO, PSO-CIFWA, and RGA results.

In [14], they used the symbiotic organisms search (SOS) algorithm to find optimal width-length ratio ( $W/L$ ) and load capacitor ( $CL$ ) for a symmetrical switching CMOS inverter. In [15], they combined the population based sine-cosine algorithm (SCA) which finds the best solution by manipulating the sine and cosine waves with the animal behaviour based modified grey-wolf optimization (mGWO) algorithm which finds the better solution by inclusion of wolf pack hierarchy to optimize a CMOS voltage reference circuit. In [16], they used a mayfly algorithm (MA) to improve switching characteristics of a CMOS inverter. The PSO is one of the best swarm-based optimization algorithms, but PSO is affected by the premature convergence. This premature convergence restricts the PSO from finding optimized solution. In order to overcome this issue, many improvised PSO and its hybrid version were proposed [17]. The hybrid algorithm combines the best strength of both algorithms, PSO's exploitation feature and GSA's exploration feature to find the optimized value. The hybrid PSO-GSA algorithm's ability to find the best optimization value is proven with comparison of the results with PSO and GSA algorithm results. The author used the hybrid PSO-GSA algorithm to find the optimized design values of a differential amplifier with active load. The results were compared with the five other optimization algorithm results. The PSO-GSA was used to find the best possible size of capacitor and the location where it fits properly. The PSO-GSA was employed in [18] in cognitive radio communication to find the best energy saving technique that can be employed in spectrum sensing. In [17] the PSO-GSA algorithm was used to classify the data available in the social network efficiently.

The remaining sections of this paper are ordered as follows: Section 2 specifies an outline of the PSO, GSA and the hybrid model of both the algorithms. Section 3 elucidates the functionality of the CMOS inverter and the equations governing its switching velocity. The objective functions, simulation outcomes, and Spice results are consolidated in Section 4. A comparative analysis of the PSO-GSA results with prior studies is presented in Section 5. Lastly, Section 6 summarises the best achieved results and the concluding observations of this paper.

## 2 HYBRID PSO-GSA ALGORITHM

### 2.1 Particle Swarm Optimization (PSO)

The PSO is a problem-solving method inspired by how birds fly together and fish swim in schools. It is used to find

the most suited solution in a boundary where many solutions are possible. Unlike genetic algorithms and simulated annealing, PSO is good at avoiding getting stuck in places where the solution is not the best. In PSO, there is a group of possible solutions called particles. These particles move around the space, with each particle having a position that could be a good solution. Also, each particle has a speed that helps it move through the space. The location and speed of each particle change over time based on its past performance and the past performance of nearby particles. The update equations [18] for the position ( $x_{id}$ ) and velocity ( $v_{id}$ ) of each particle are as follows, the velocity update expression is

$$v_i^{(k+1)} = wv_i^k + c_1r_1(p_i^k - x_i^k) + c_2r_2(p_g^k - x_i^k) \quad (1)$$

The position update expression is

$$x_i^{(k+1)} = x_i^k + v_i^{(k+1)} \quad (2)$$

In Eq. (1) and Eq. (2),  $w$  is the inertia weight, which governs the variation in current velocity compared to the previous velocity.  $v_i^k$  represents the velocity of particle  $i$  at the  $k^{th}$  iteration.  $c_1$  and  $c_2$  are acceleration coefficients that regulate each of the particle's logical behaviour and also the location based social behaviour.  $r_1$  and  $r_2$  are random values that fall from 0 up to 1.  $p_i^k$  denotes the primelast location of particle  $i$  that was obtained in the  $k^{th}$  iteration (personal best), while  $p_g^k$  stands for the primespot among all particles that was obtained in the  $k^{th}$  reiteration (global best). Lastly,  $x_i^k$  represents the present position of particle  $i$  at the  $k^{th}$  iteration. The algorithm continues over multiple iterations, adjusting the speeds and locations of the particles until a stopping condition is satisfied, like reaching a set number of iterations or finding an acceptable solution. PSO aims to move closer to the best solution by using the particles to explore the search space and use the best solutions found by each particle and the entire group.

### 2.2 Gravitational Search Algorithm (GSA)

The space body interaction based GS Algorithm is an optimization method based on the law of gravity and how celestial bodies interact. It addresses optimization issues by imitating how masses in the universe exert gravitational forces on each other. In GSA, potential solutions to the optimization problem are depicted as masses, with each solution acting like a mass attracting or repelling others depending on its quality (fitness) [18]. The gravitational force between two masses is computed using Newton's law of universal gravitation.

$$F = \frac{Gm_1m_2}{r^2} \quad (3)$$

The masses (solutions) move towards each other based on the gravitational force. Higher fitness solutions attract other masses more strongly. The positions of masses are updated based on the gravitational force, allowing them to

converge towards better solutions over iterations. The algorithm's efficiency depends on parameters like the gravitational constant, how the gravitational force is calculated, and how the masses positions are updated. GSA is appreciated for being easy to understand, needing fewer adjustments, and effectively exploring the entire search area. However, it can take a while to find the best solution, especially as it gets closer and it requires significant computation, particularly with a large number of potential solutions, because every solution's influence on every other.

### 2.3 Particle Swarm Optimization Hybridized with Gravitational Search Algorithm (PSOGSA)

PSOGSA effectively balances GSA's exploration with PSO's exploitation in CMOS optimization, typically outperforming its standalone parent algorithms. It reportedly improves CMOS area, performance, and convergence over non-hybrid algorithms like DE, HS, and ABC [19]. PSOGSA shows superior local optima avoidance and faster convergence than base PSO/GSA on general benchmarks [20]. Comparisons with other hybrid algorithms yield varied, problem-specific outcomes. For a CMOS op-amp, a Ts-CPD hybrid achieved better area minimization than PSOGSA [21]. In the same study, however, PSOGSA offered lower power consumption and superior DC gain/phase margin [21]. PSOGSA's fundamental advantage is its intrinsic mechanism for a balanced global and local search strategy.

In the fusion PSOGSA, the PSO and GSA components work in tandem, influencing each other in the search for an optimal solution. PSO focuses on manipulation, utilizing historical information and best solutions. Each of the elements or particles changes its velocity on the basis of two factors. One is the cognitive component which is controlled by the best known location of the particle and the other is the social component which is influenced by the best known position of an element or particle amount of all the particles. This methodology makes the particles narrow down to the best solution among the overall searching space. GSA introduces randomness and exploration into the search process. Gravitational forces are applied randomly to the particle's velocities after each PSO iteration. These random forces encourage exploration by introducing perturbations in particle movement. GSA helps prevent stagnation in local optima by allowing particles to explore new regions of the search space. The PSO component adjusts particle velocities primarily based on historical information and best solutions. Meanwhile, the GSA component adds random forces to the velocities, aiming to promote exploration. PSO tends to exploit known promising regions by steering particles towards good solutions. GSA complements PSO by injecting randomness, allowing particles to explore unknown regions even if they might not seem promising initially. The initial parameters considered for this work is tabulated in Tab. 1.

The pseudocode of PSOGSA is as follows:

- Step 1. Adjust PSO-GSA Parameters
- Adjust num\_particles, num\_iterations,  $V_{DD}$
- Define Constants:  $CL_{range}$ ,  $WLn_{range}$ ,  $WLP_{range}$
- Set PSO Parameters:  $W$ ,  $c_1$ ,  $c_2$ ,  $G$

- Step 2. Initialize Particle Variables
  - Initialize particles\_\_position arbitrarily within specified ranges
  - Initialize particles\_\_velocity arbitrarily
  - Initialize particles\_\_best\_position and particles\_\_best\_value
  - Set global\_\_best\_position and global\_\_best\_value
- Step 3. Main Loop
  - For each iteration from 1 to num\_iterations:
    - For each particle from 1 to num\_particles:
      - Evaluate Cost Function for Particle
      - Calculate current\_value for the particle using  $CF(CL, WLn, WLP, V_{DD})$
      - Update particles\_\_best\_position and particles\_\_best\_value if better solution found
      - Update global\_\_best\_position and global\_\_best\_value if better solution found
      - Display Iteration and Best Values
      - Update Particle Velocities and Positions (PSO)
        - For each particle:
          - Revise particles\_\_velocity based on PSO formulas
          - Revise particles\_\_position based on velocities
          - Update Gravitational Forces (GSA)
            - For each particle:
              - Add random gravitational forces to particle velocities
      - End of Main Loop
    - Step 4. Display Final Optimized Values
      - Display ( $CL$ ,  $WLn$ ,  $WLP$ )
      - Calculate and Display Optimized Transfer Functions ( $T_f$ ,  $T_r$ ,  $T_{pHL}$ ,  $T_{pLH}$ )
      - Display Final Optimized Cost Function Value

Table 1 PSOGSA initial parameter values

Parameter	Value Considered
num_particles	50
num_iterations	500
$w$	0,7
$c_1$	2
$c_2$	2
$G$	100

### 3 VOLTAGE TRANSFER CHARACTERISTICS OF CMOS INVERTER

The voltage transfer characteristics of a CMOS inverter determine how the output interacts with the input and also the output changes in response to input alterations, crucial for circuit design. The Fall time ( $t_f$ ) as expressed by expression (4) is the time band needed for the output voltage to shift from a higher level value to the lowest level during the dropping transition of the input signal. It is usually measured between the 9/10<sup>th</sup> and 1/10<sup>th</sup> point of the output VTC. Rise time ( $t_r$ ) is the time gap needed for the output voltage to move on from a lower value to a higher level during the input signal's rising transition. It is typically measured between the 1/10<sup>th</sup> and 9/10<sup>th</sup> points of the output voltage waveform. These values are computed using Eq. (4) and Eq. (5) which is mentioned in [16], respectively.

$$t_f = \frac{CL}{\mu_n C_{OX} \left(\frac{W}{L}\right)_n (V_{DD} - V_m)} \cdot \left[ \frac{2(V_m - 0.1V_{DD})}{(V_{DD} - V_m)} + \ln \left( \frac{2(V_{DD} - V_m) - 0.1V_{DD}}{0.1V_{DD}} \right) \right] \quad (4)$$

$$t_r = \frac{C_L}{\mu_p C_{OX} \left(\frac{W}{L}\right)_p (V_{DD} - |V_{tp}|)} \cdot \left[ \frac{2(|V_{tp}| - 0.1V_{DD})}{(V_{DD} - |V_{tp}|)} + \ln \left( \frac{2(V_{DD} - |V_{tp}|) - 0.1V_{DD}}{0.1V_{DD}} \right) \right] \quad (5)$$

$$t_{pHL} = \frac{C_L}{\mu_n C_{OX} \left(\frac{W}{L}\right)_n (V_{DD} - V_{tn})} \cdot \left[ \frac{2V_{tn}}{(V_{DD} - V_{tn})} + \ln \left( \frac{4(V_{DD} - V_{tn})}{V_{DD}} - 1 \right) \right] \quad (6)$$

$$t_{pLH} = \frac{C_L}{\mu_p C_{OX} \left(\frac{W}{L}\right)_p (V_{DD} - |V_{tp}|)} \cdot \left[ \frac{2|V_{tp}|}{(V_{DD} - |V_{tp}|)} + \ln \left( \frac{4(V_{DD} - |V_{tp}|)}{V_{DD}} - 1 \right) \right] \quad (7)$$

The Eq. (6) as mentioned in [16] is used to compute the high to low propagation delay ( $t_{pHL}$ ). The  $t_{pHL}$  represents the time band between the change of input signal from a lower voltage point to higher voltage point and the output signal's movement from a higher point to lower point in the graph. Simply, it represents the time band between the inputs achieving a 1/2<sup>th</sup> of its final value to that of the output achieving the same. In the same way the expression (7) as mentioned in [16] is used to compute the low to high propagation delay ( $t_{pLH}$ ). The mentioned the time gap between the inputs moving from a higher point to a lower point in the graph, to the output swinging from a lower point to a higher point. It short it is the time band between the input achieving a 1/2<sup>th</sup> of its final value to that of the output achieving the same.

#### 4 PROBLEM FORMULATION

This paper examines three distinct scenarios to analyse the transient behaviour of the CMOS inverter. Case I focuses on exploring the  $t_f$  of the CMOS inverter's output voltage. Case II involves the schemes to form a CMOS inverter that is able to produce a symmetric output waveform with equal  $t_r$  and  $t_f$ . Finally, Case III proposes the design parameters for improving the balance of the CMOS inverter's output voltage by achieving equality between  $t_r$  and  $t_f$  as well as equal  $t_{pHL}$  and  $t_{pLH}$ . Here, TSMC 45 nm library file is used for this design with the following parameters:  $V_{DD} = 1,7$  V,  $V_{tn} = 0,25$  V,  $V_{tp} = 0,50$  V,  $\mu_n C_{OX} = 183,6 \frac{\mu A}{V^2}$ ,  $\mu_p C_{OX} = 47,6 \frac{\mu A}{V^2}$ . Tab. 2 shows the design

parameter  $\left(\frac{W}{L}\right)_n$ ,  $\left(\frac{W}{L}\right)_p$  and  $C_L$  value range sets used in all the three cases. These sets of values were considered based on the previous CMOS inverter design literatures [5, 7, 11-15]. These combinations of range bound were

considered because they provide optimal result which was determined by iterative process [5].

Table 2 CMOS design parameter bound for each set

Set No.	$\left(\frac{W}{L}\right)_n$	$\left(\frac{W}{L}\right)_p$	$C_L / \text{pF}$
1	1,1-6,1	2,8-19,3	0,2-4,0
2	1,6-7,1	1,8-18,0	0,1-5,1
3	1,4-6,7	3,2-38,0	0,47-2,0
4	1,2-7,0	1,5-17,5	0,1-1,1
5	1,9-5,0	2,7-17,0	0,2-14
6	1,3-3,5	3,5-16,2	0,3-3,6
7	1,1-5,8	2,2-25,3	0,2-4,9
8	0,3-7,6	1,3-39,0	0,2-3,5

#### 4.1 Case I

This case aims to minimize the  $t_f$  by finding the optimum values of load capacitance ( $C_L$ ) and width length ratio  $\left(\frac{W}{L}\right)_n$  of NMOS. During this optimization process the  $C_L$  value and  $\left(\frac{W}{L}\right)_n$  value were restricted to a specified range. The  $t_f$  values were calculated based on the Eq. (4). The PSO-GSA synthesized  $t_f$  value ranges corresponding to the CMOS design parameter bounds mentioned in Tab. 1 are tabulated in Tab. 3.

Table 3 Case I PSO-GSA optimized inverter synthesized results

Set No.	$t_f / \text{ns}$
1	0,189-0,998
2	0,096-0,709
3	0,396-0,754
4	0,081-0,254
5	0,238-2,376
6	0,491-1,622
7	0,200-1,003
8	0,160-0,732

#### 4.2 Case II

To achieve a symmetric switching response, equal  $t_f$  and  $t_r$  of the output voltage are desired. However, due to influence of second order effects, there is a consistent variation between  $t_f$  and  $t_r$ . The primary objective of this case is to determine the design parameters that minimize the cost function (i.e.) difference between  $t_f$  and  $t_r$ . The cost function ( $CF$ ) to be minimized is as follows,

$$CF = \left| \left( t_f \left( C_L, \left( \frac{W}{L} \right)_n \right) - t_r \left( C_L, \left( \frac{W}{L} \right)_p \right) \right) \right| \quad (8)$$

where  $t_f \left( C_L, \left( \frac{W}{L} \right)_n \right)$  means fall time ( $t_f$ ) Eq. (4). The  $t_f$  is the time taken for the load capacitor ( $C_L$ ) to discharge through the pull down NMOS transistor.

The  $t_r \left( C_L, \left( \frac{W}{L} \right)_p \right)$  means rise time ( $t_r$ ) expression (5). The  $t_r$  is the time taken for the load capacitor ( $C_L$ ) to charge through the pull up PMOS transistor. The Eq. (8) is subjected to the following constrains.

**Table 4** Case II PSO-GSA optimized inverter's synthesized results

Set No.	$t_f$ / ns	$t_r$ / ns	$CF$ / ps
1	0,39-18,01	0,41-18,02	1,88-325,43
2	0,20-6,47	0,29-6,49	3,72-584,47
3	0,58-2,69	0,59-2,68	2,91-93,82
4	0,16-2,19	0,24-2,20	2,60-91,98
5	0,27-30,28	0,80-3,021	6,91-620,48
6	1,07-8,17	0,10-8,21	10,13-169,71
7	0,38-6,19	0,37-6,18	4,26-283,43
8	0,23-3,88	0,19-3,91	1,78-146,37

$$(C_L)_{\min} \leq C_L \leq (C_L)_{\max} \quad (9)$$

$$\left( \left( \frac{W}{L} \right)_n \right)_{\min} \leq \left( \frac{W}{L} \right)_n \leq \left( \left( \frac{W}{L} \right)_n \right)_{\max} \quad (10)$$

$$\left( \left( \frac{W}{L} \right)_p \right)_{\min} \leq \left( \frac{W}{L} \right)_p \leq \left( \left( \frac{W}{L} \right)_p \right)_{\max} \quad (11)$$

The PSO-GSA synthesized  $t_f$ ,  $t_r$  and  $CF$  value (the difference between  $t_f$  and  $t_r$ ) corresponding to the CMOS design parameter bounds are listed in Tab. 4.

### 4.3 Case III

For symmetric switching, it is essential to have equal  $t_f$  and  $t_r$  and also  $t_{pLH}$  and  $t_{pHL}$  for the output voltage. The main goal of this scenario is to identify the design parameters that will support to minimize the cost function ( $CF$ ) which further reduces the associated delays. The  $CF$  to be minimized is as described in (4,16).

**Table 5** Case III PSO-GSA optimized inverter's synthesized result

Set No.	$t_f$ / ns	$t_r$ / ns	$t_{pHL}$ / ns	$t_{pLH}$ / ns	$CF$ / ps
1	0,33-4,69	0,34-4,60	0,14-1,99	0,15-2,02	15,74-285,33
2	0,16-8,75	0,20-8,70	0,07-3,71	0,08-3,83	10,57-454,57
3	0,46-2,31	0,44-2,23	0,19-0,98	0,19-0,98	9,14-111,68
4	0,16-2,19	0,24-2,20	0,07-0,93	0,10-0,96	9,09-135,05
5	0,39-1,80	0,44-1,74	0,16-7,63	0,19-7,68	16,44-924,17
6	0,63-5,10	0,62-5,00	0,26-2,16	0,27-2,20	11,60-274,30
7	0,26-5,19	0,28-5,15	0,11-2,20	0,12-2,26	6,06-251,23
8	0,19-2,83	0,21-2,81	0,08-1,20	0,09-1,24	4,67-108,44

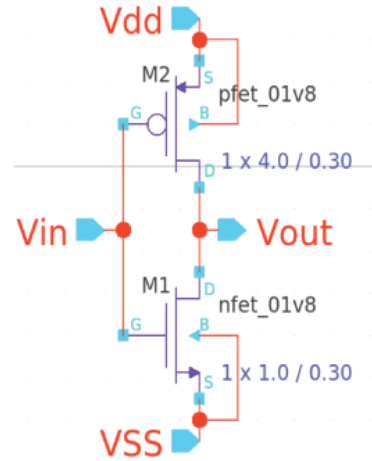
The PSO-GSA synthesized  $t_f$ ,  $t_r$ ,  $t_{pHL}$  and  $t_{pLH}$  and  $CF$  value range corresponding to the CMOS design parameter bounds are tabulated in Tab. 5. The time delays are mentioned in nanoseconds and cost function is mentioned in picoseconds.

$$CF = \left| t_f \left( C_L, \left( \frac{W}{L} \right)_n \right) - t_r \left( C_L, \left( \frac{W}{L} \right)_p \right) \right| + \left| t_{pHL} \left( C_L, \left( \frac{W}{L} \right)_n \right) - t_{pLH} \left( C_L, \left( \frac{W}{L} \right)_p \right) \right| \quad (12)$$

## 5 RESULTS AND DISCUSSION

This section compares the optimal switching characteristics synthesized by PSO-GSA with the corresponding PSpice results and results synthesized by the

various techniques. Fig. 2 shows spice simulated CMOS circuit. The simulations are experimented using Cadence EDA tool.


**Figure 2** The spice schematic CMOS Inverter circuit

### 5.1 Case I

The Case I focuses on comparing the PSO-GSA synthesized CMOS optimal fall time ( $t_f$ ) with PSpice results and also with the results of literature review techniques discussed earlier. The minimalistic synthesized  $t_f$  value is 0,081498 ns and is got for the 4<sup>th</sup> set of Tab. 1 parameter range. The simulated result for the same set of parameters is 0,0808340 ns. The difference between the synthesized and simulated value is very small 0,664 ps which shows that the optimized CMOS design parameters are practically feasible.

**Table 6** Case I PSO-GSA optimized inverter's  $t_f$  (ps) differential value comparison with other algorithm based inverter's  $t_f$  (ps) differential value

Set No.	$t_f$ / ps						
	PSO [4]	DE [9]	CRPSO [10]	HS-DE [11]	ALC-PSO [14]	MA [16]	PSOGSA (Proposed)
1	3069	748,7	555,9	397,61	296,1	1377,96	15,98
2	4480	1816,0	908,6	655,3	508,7	1182,66	8,29
3	6243	1786,6	1554,4	460,1	627,3	1402,79	33,93
4	7050	2333,4	2281,4	736,7	453,1	1649,16	0,66
5	4041	2280,9	1651,3	725,6	700,6	1543,57	15,15
6	4384	1678,7	1675,5	722	343,7	1815,72	28,70
7	1511	1551,7	1505,0	813,6	786,6	2010,54	17,07
8	4741	1667,1	1217,2	593,3	605,6	2075,71	2,31

Each column in Tab. 6 shows the the difference between the algorithm synthesized optimal inverter's  $t_f$  value and the corresponding PSpice simulated results. The Tab. 6 compares this differential  $t_f$  of PSO-GSA based inverter with the differential value obtained for SO [4], DE [9], CRPSO [10], HS-DE [11], ALC-PSO [14], MA [16] based inverters. It has been shown that the proposed optimization technique PSO-GSA is able to attain a minimal difference between the synthesized  $t_f$  and simulated  $t_f$  results.

### 5.2 Case II

Case II focuses on comparing the PSO-GSA synthesized cost function (8) with PSpice results and other optimization techniques. The minimalistic synthesized cost function ( $CF$ ) value is 1,884 ps and is got for the 1<sup>st</sup> set of

Tab. 1 parameter range. The simulated result for the 1<sup>st</sup> set of  $C_L$  and  $(W/L)_n$  is 2,600 ps. The difference between the synthesized and simulated value is very small 0,715 ps which shows that the optimized CMOS design parameters are practically feasible. The optimum value of the cost function ( $CF$ ) is 0,715 ps and it converges after 55 iterations.

Each column in Tab. 7 represents the difference between the algorithm synthesized  $CF$  and the corresponding simulation  $CF$  value of each optimization algorithm. Tab. 7 as a whole compares this differential  $CF$  value (in nanoseconds) of each algorithm's optimized inverter with PSO/GSA optimized inverter. Each row showcases the superiority of PSO/GSA optimized inverter's error difference over other algorithm's differential  $CF$  value. The differential  $CF$  value 0.48 ns obtained for the 7<sup>th</sup> set of parameter bounds mentioned in Tab. 1 is the minimal among all the values present in the Tab. 7.

**Table 7** Case II PSO/GSA optimized Inverter's  $CF$  (ns) differential value comparison with other algorithm optimized inverter's  $CF$  (ns) differential value

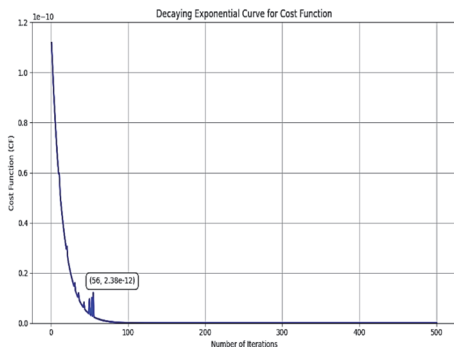
Set No.	Cost function $CF$ / ns							
	PSO [3]	PSO [4]	DE [9]	CRPSO [10]	HS-DE [11]	ALC-PSO [14]	MA [16]	PSO/GSA (Proposed)
1	2360	850	353,8	332,4	119,4	85,8	541,9	0,71
2	1730	1170	402,5	33,36	73,8	79,1	618,4	0,78
3	2950	2150	996,5	813,8	144	103,2	67,9	3,88
4	4270	1500	147,6	137,6	41,1	58,8	504,1	0,52
5	6400	960	642,4	564,6	93,3	145	636,2	6,75
6	2354	1750	775,0	667,1	91,3	95,8	925,5	6,83
7	2000	1370	976,1	949,5	87	934	978,4	0,48
8	2400	60	1260,3	804,4	76,1	83,8	966,6	5,75

**5.3 Case III**

Case III focuses on comparing the PSO/GSA synthesized cost function (17) with PSpice results and other optimization techniques. Tab. 8 shows the PSO/GSA synthesized optimal inverter's design parameters,  $t_f$ ,  $t_r$ ,  $t_{pHL}$  and  $t_{pLH}$  and  $CF$  values. The least  $CF$  value 4.676 ps is achieved for the 8th set of optimal design parameters.

**Table 8** Case III PSO/GSA optimized inverter's optimal time delay values and optimal parameter values

Set No.	$C_L$ / pF	$(\frac{W}{L})_n$	$(\frac{W}{L})_p$	$t_f$ / ns	$t_r$ / ns	$t_{pHL}$ / ns	$t_{pLH}$ / ns	$CF$ / ps
1	0,584	3,269	17,464	0,991	0,990	0,420	0,436	15,745
2	0,213	3,150	17,202	0,376	0,367	0,159	0,162	10,570
3	0,493	5,719	30,729	0,478	0,475	0,203	0,209	9,150
4	0,158	2,794	15,286	0,313	0,306	0,133	0,134	9,092
5	0,408	2,803	15,093	0,807	0,800	0,343	0,352	16,448
6	0,316	2,778	14,908	0,632	0,629	0,268	0,277	11,606
7	0,223	4,258	22,950	0,291	0,288	0,123	0,127	6,060
8	0,253	6,158	33,168	0,228	0,226	0,097	0,099	4,676



**Figure 3** Plot of  $CF$  with iterations for the 8th design set of Case III

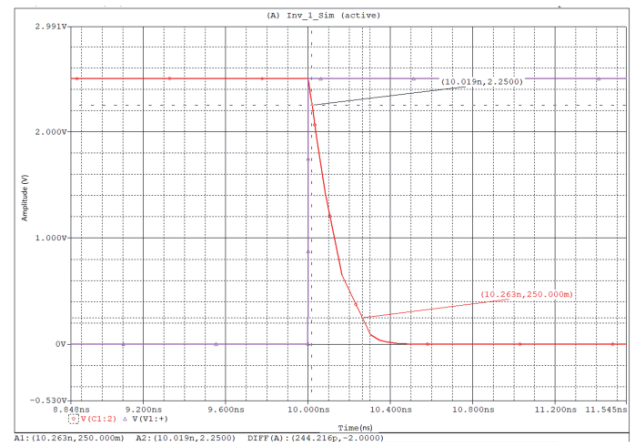
Tab. 9 shows the PSpice simulated  $t_f$ ,  $t_r$ ,  $t_{pHL}$  and  $t_{pLH}$  and  $CF$  for the optimal parameters values mentioned in the 2<sup>nd</sup> to 4<sup>th</sup> column of the Tab. 8. In this table also the least value 5.206 ps is achieved at the 8<sup>th</sup> set of optimal design parameters. The difference in the PSO/GSA synthesized  $CF$  and the pspice simulated  $CF$  is listed out in the last column of Tab. 10. Tab. 10 compares the differential  $CF$  value of various algorithms with PSO/GSA optimized inverter's value mentioned in the last column. The proposed inverter has the least differential value among all the values derived. Among all the PSO/GSA values, the most least difference which is 0.5 ps is achieved for the 8<sup>th</sup> set of optimal design parameters. This shows that the PSO/GSA synthesized optimal inverter has the performance very similar to the practical circuit with only a negligible difference in all three cases. Fig. 3 shows the convergence plot of the cost function ( $CF$ ) for the 8<sup>th</sup> design set. The optimum value of the cost function ( $CF$ ) is 4.6761 ps and it converges after 50 iterations.

**Table 9** Case III PSO/GSA optimized inverter's simulated time delay results

Set No.	$C_L$ / pF	$(\frac{W}{L})_n$	$(\frac{W}{L})_p$	$t_f$ / ns	$t_r$ / ns	$t_{pHL}$ / ns	$t_{pLH}$ / ns	$CF$ / ps
1	0,584	3,269	17,463	1,047	1,058	0,440	0,455	26,029
2	0,213	3,150	17,202	0,408	0,408	0,162	0,165	3,448
3	0,493	5,719	30,729	0,509	0,513	0,212	0,220	11,690
4	0,158	2,794	15,286	0,305	0,300	0,137	0,140	7,411
5	0,408	2,803	15,093	0,876	0,889	0,351	0,364	27,064
6	0,316	2,778	14,908	0,618	0,634	0,276	0,285	24,832
7	0,223	4,258	22,950	0,292	0,291	0,128	0,132	5,114
8	0,253	6,158	33,168	0,244	0,246	0,101	0,104	5,206

**Table 10** Case III PSO/GSA optimized Inverter's  $CF$  (ps) comparison with other algorithm optimized inverter's  $CF$  (ps)

Set No.	Cost function $CF$ / ps								
	PSO [4]	ANN-PSO [6]	DE [9]	CRPSO [10]	HS-ALC-DE [10]	PSO [14]	SOS [15]	MA [16]	PSO/GSA (Proposed)
1	2220	318,4	844,6	785,3	101	110,8	447,7	1058,2	10,3
2	1620	712,9	731,5	466,5	142	115,4	302,1	1025,5	7,1
3	1060	587	1025	574,0	171	94,3	398,4	522,6	2,5
4	570	437,6	387,8	195,7	164	113,6	400,5	338,2	1,6
5	1650	288,2	683,5	425,8	136	116,8	657,3	565,0	10,6
6	1220	448,3	373,9	374,5	108	85,7	575,4	461,7	13,2
7	1020	580,3	526,5	371,0	162	77,3	426,1	464,9	0,9
8	3130	-	501,8	306,8	79	73,2	289,9	308,2	0,5



**Figure 4** Spice simulation for the 8<sup>th</sup> design set of Case III Fall Time ( $t$ )

Fig. 6 and Fig. 7 showcase the propagation delays  $t_{pLH}$  and  $t_{pHL}$  simulated graph for the case III optimal inverter parameters obtained in the 8<sup>th</sup> set of Tab. 10.

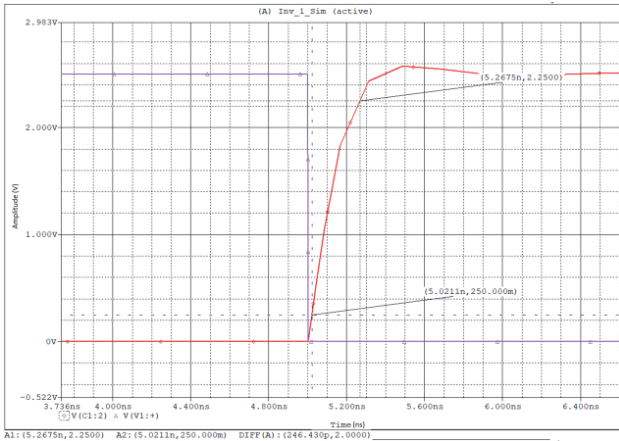


Figure 5 Spice simulation for the 8<sup>th</sup> design set of Case III Rise Time ( $t_r$ ).

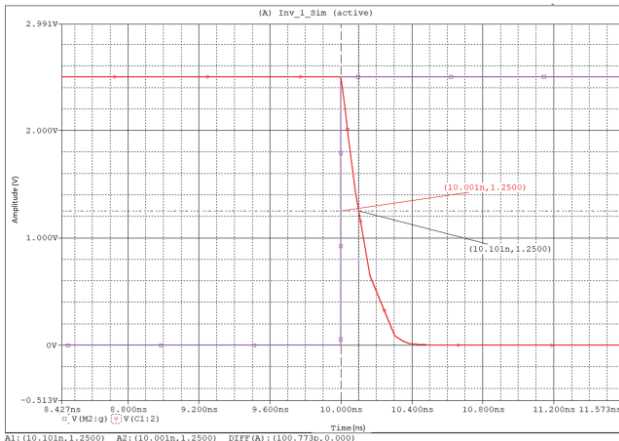


Figure 6 PSpice simulated graph of High to Low propagation delay ( $t_{PHL}$ ) for the 8<sup>th</sup> design set of 3<sup>rd</sup> case

The margin is a crucial factor with which the tolerance level of a circuit to level of noise in the input to generate an error free output. The tolerance level is denoted as the noise immunity. A good noise immunity is formed by greater voltage transition and sharp transfer of voltage values. These factors improve the circuit's tolerance rate to input noise and generate error free output by preventing the influence of noise in the output. For a binary or digital systems, two values governing this noise immunity are

$$NML(\text{NoiseMarginforLow}) = V_{IL} - V_{OL} \quad (13)$$

$$NMH(\text{NoiseMarginforHigh}) = V_{OH} - V_{IH} \quad (14)$$

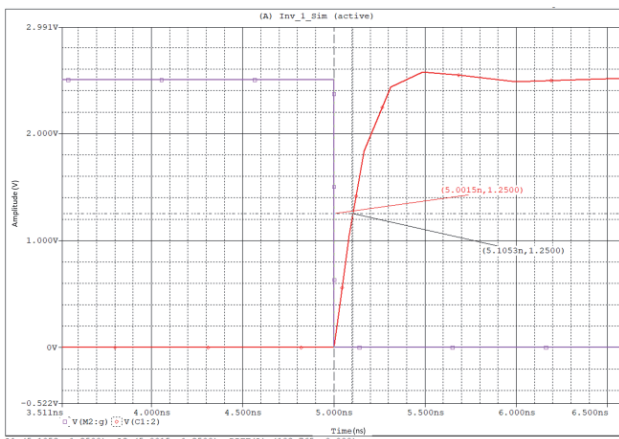


Figure 7 PSpice simulated graph of Low to High propagation delay ( $t_{PLH}$ ) for the 8<sup>th</sup> design set of 3<sup>rd</sup> case

Table 11 Performance comparison of case III PSO-GSA optimized inverter with other algorithm optimized inverter

Optimization Algorithm	PSO	DE	HS-DE	ALC-PSO	MA	PSOGSA
Propagation Delay / ns	0,5	0,7	0,6	0,4	0,5	0,6
Power Consumption / mW	5,235	8,863	6,162	9,034	7,348	6,652
Symmetrical Switching / %	95	92	94	96	92	94
Noise Margin / V	0,1-0,2	0,2-0,3	0,15-0,25	0,1-0,2	0,1-0,2	0,15-0,3
Transistor Sizing / Biasing	B	LT, HB	B	LT, HB	B	B

\*B - Balanced, LT - Large transistors, High biasing.

From the simulated results the value of NML is 0,74 V and NMH is 0,82 V. Here, the NML and NMH are not equal. The NML can be made equal to the NMH by adjusting the value of  $V_{th}$  nearer to half of the supply voltage ( $V_{DD}/2 = 0,9$ ). The PSO-GSA optimization algorithm achieves the 98% symmetrical switching and less noise range of 0,1 V to 0,2 V by trading of the transistor size and biasing voltage. This level of low noise margin is achieved by the reduction of differences in the time delays discussed in the three cases. The sharp transition achieved through the hybrid algorithm made the CMOS inverter perform better than those designed by other algorithm.

## 6 CONCLUSION

This research presents a hybridized algorithm based approach for designing a high-performance CMOS inverters with superior switching symmetry. The optimization outcomes are verified through spice simulations using a TSMC 45 nm MOS technology in cadence EDA tool. In Case I, various design parameters influence the fall time estimation. By exploring eight sets of design criteria, PSO-GSA aligns closely with spice simulation results, with a maximum fall time difference of 33,931 ps. Case II aims for an inverter which achieves a symmetric fall transition and rise transition. In this case the maximum difference between synthesized and simulated cost functions at 6,833 ps. Case III focuses on achieving symmetry in fall, rise transition, and propagation delay. Spice simulations confirm this balance transition, with minimal differences between synthesized and simulated results (ranging from 0,5299 ps to 13,226 ps). Compared to other methods, PSO-GSA exhibits superior performance in matching practical simulations and achieving low power consumption of 5,2 mW and low margin of 0,2 V. The above mentioned symmetrical switching is achieved by compromising the size of the transistor. Even though the performance metric shows a balanced design between the transistor size and biasing, still it is larger in size. The future work is to derive a cost function that integrates the size, power consumption and switching speed and also the inclusion of artificial intelligence along with PSO-GSA for the optimal design of more complex CMOS circuits.

## 7 REFERENCES

- [1] Wu, C. (2024). Reduce energy consumption - causes of CMOS inverter switching delay and its influencing factors. *Applied and Computational Engineering*, 78, 67-75.

- <https://doi.org/10.54254/2755-2721/78/20240466>
- [2] Zihan, L. (2024). On the propagation delay of CMOS inverters. *Applied and Computational Engineering*, 84, 163-172. <https://doi.org/10.54254/2755-2721/84/20240876>
- [3] Acar, V., Der, O., & Yildirim, T. (2010). Particle swarm optimization based inverter design considering transient performance. *Digital Signal Processing*, 20(4), 1215-1220. <https://doi.org/10.1016/j.dsp.2009.10.022>
- [4] Acar, V., Der, O., & Yildirim, T. (2010). Investigation of particle swarm optimization for switching characterization of inverter design. *Expert Systems with Applications*, 38(5), 5696-5703. <https://doi.org/10.1016/j.eswa.2010.10.064>
- [5] Valavala, L. T., Munot, K., & Karri, B. R. T. (2018). Design of CMOS Inverter and Chain of Inverters Using Neural Networks. *2018 IEEE International Symposium on Smart Electronic Systems (iSES)*, 269-274. <https://doi.org/10.1109/iSES.2018.00065>
- [6] Dhabak, D. & Pandit, S. (2011). An artificial neural network-based approach for performance modeling of nano-scale CMOS inverter. *Institute of Engineering and Management Conference*, 165-170. <https://doi.org/10.1155/2012/505983>
- [7] Saxena, V., Sreejeth, N., & Singh, K. (2020). Interior Search Algorithm integrated Matlab SPICE Interface For Optimization of CMOS Inverter Switching Characteristics. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1-6. <https://doi.org/10.1109/INOCON50539.2020.9298226>
- [8] De, B. P., Kar, R., Mandal, D., & Ghoshal, S. P. (2014). Design of symmetric switching CMOS inverter using PSO/FIWA. *2014 International Conference on Communication and Signal Processing*, 1818-1824. <https://doi.org/10.1109/ICCSP.2014.6950160>
- [9] De, B. P., Kar, R., Mandal, D., & Ghoshal, S. P. (2015). Optimal CMOS inverter design using differential evolution algorithm. *Journal of Electrical Systems and Information Technology*, 2(2), 219-241. <https://doi.org/10.1016/j.jesit.2015.03.014>
- [10] Chowdhury, D., De, B. P., Maji, K. B., Ghosh, S., Kar, R., & Mandal, D. (2019). Optimization Algorithm. *Proceedings of the 2nd International Conference on Communication, Devices and Computing: ICCDC 2019*, 602, 203.
- [11] De, B. P., Kar, R., Mandal, D., & Ghoshal, S. P. (2016). Optimal design of high speed symmetric switching CMOS inverter using hybrid harmony search with differential evolution. *Soft Computing*, 20, 3699-3717. <https://doi.org/10.1007/s00500-015-1731-4>
- [12] Kumar, M., Rawat, T. K., & Majhi, A. (2016). Design of symmetric switching CMOS inverter using cuckoo search algorithm. *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 1-6. <https://doi.org/10.1109/ICPEICES.2016.7853659>
- [13] De, B. P., Kar, R., Mandal, D., & Ghoshal, S. P. (2015). Optimal high speed CMOS inverter design using craziness based Particle Swarm Optimization Algorithm. *Open Engineering*, 5(1). <https://doi.org/10.1515/eng-2015-0031>
- [14] De, B. P., Kar, R., Mandal, D., & Ghoshal, S. P. (2017). PSO with aging leader and challengers for optimal design of high speed symmetric switching CMOS inverter. *International Journal of Machine Learning and Cybernetics*, 8, 1403-1422. <https://doi.org/10.1007/s13042-016-0517-z>
- [15] Dib, N. & Al-Sammarräie, U. (2020). Optimal design of symmetric switching CMOS inverter using symbiotic organisms search algorithm. *International Journal of Electrical & Computer Engineering*, 10(1), 171-179. <https://doi.org/10.11591/ijece.v10i1.pp171-179>
- [16] Zghoul, F. N., Altechi, H., & Abuelrub, A. (2023). A mayfly-based approach for CMOS inverter design with symmetrical switching. *Algorithms*, 16(5), 237. <https://doi.org/10.3390/a16050237>
- [17] Ramírez-Ochoa, D. D., Pérez-Domínguez, L. A., Martínez-Gómez, E. A., & Luviano-Cruz, D. (2022). PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: A review. *Symmetry*, 14(3), 455. <https://doi.org/10.3390/sym14030455>
- [18] Kaur, K., Rattan, M., & Patterh, M. S. (2018). Cognitive radio design optimization over fading channels using PSO, GSA and hybrid PSO/GSA. *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1700-1706. <https://doi.org/10.1109/ICCONS.2018.8662949>
- [19] Jayaprakasam, S., Rahim, S. K. A., Chee, Y., & Leow, Y. (2015). PSO/GSA-Explore: A new hybrid metaheuristic approach for beam pattern optimization in collaborative beamforming. *Applied Soft Computing*, 30. <https://doi.org/10.1016/j.asoc.2015.01.051>
- [20] Mallick, S., Kar, R., Mandal, D., & Ghoshal, S. P. (2017). Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization. *International Journal of Machine Learning and Cybernetics*, 8, 309-331. <https://doi.org/10.1007/s13042-014-0324-3>
- [21] Lagos-Eulogio, P., Miranda-Romagnoli, P., Seck-Tuoh-Mora, J. C., & Hernández-Romero, N. (2023). Improvement in sizing constrained analog IC via Ts-CPD Algorithm. *Computation*, 11(11), 230. <https://doi.org/10.3390/computation11110230>

**Contact information:**

**V. P. JAY FANTIN**, Assistant Professor  
(Corresponding Author)  
Department of Electronics and Communication Engineering,  
SSM Institute of Engineering and Technology,  
Dindigul-Palani Highway, Dindigul - 624002, India  
E-mail: jayfantinssm@rediffmail.com

**G. MOHANBABU**, Principal & Professor  
J.N.N Institute of Engineering,  
Chennai, Tamil Nadu - 601102, India  
E-mail: shamyubabu@gmail.com