❐   171

Optimal design of symmetric switching CMOS inverter using symbiotic organisms search algorithm

Nihad Dib[1], Umar Al-Sammarraie[2]
[1]Department of Electrical Engineering, Jordan University of Science and Technology, Jordan
[2]Department of Social Sciences, Technische Universität Kaiserslautern, Germany
[2]Department of Electrical and Communications Engineering, German Jordanian University, Jordan

Article Info	ABSTRACT
	This paper investigates the optimal design of symmetric switching CMOS inverter using the Symbiotic Organisms Search (SOS) algorithm. SOS has been recently proposed as an effective evolutionary global optimization method that is inspired by the symbiotic interaction strategies between different organisms in an ecosystem. In SOS, the three common types of symbiotic relationships (mutualism, commensalism, and parasitism) are modeled using simple expressions, which are used to find the global minimum of the fitness function. Unlike other optimization methods, SOS has no parameters to be tuned, which makes it an attractive and easy-to-implement optimization method. Here, SOS is used to design a high speed symmetric switching CMOS inverter, which is considered the most fundamental logic gate. SOS results are compared to those obtained using several optimization methods, like particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), and other ones, available in the literature. It is shown that the SOS is a robust straight-forward evolutionary algorithm that can compete with other well-known advanced methods.

Corresponding Author:

Nihad Dib,
Department of Electrical Engineering,
Jordan University of Science and Technology,
P. O. Box 3030, Irbid 22110, Jordan.
Email: nihad@just.edu.jo

1.   INTRODUCTION

Needless to say, the inverter is the most basic logic gate that performs a Boolean operation on a single input. It is considered as the main component in the design of more complex digital integrated circuits (ICs). A CMOS inverter consists of NMOS and PMOS transistors, where the transistors' geometries (channel width and length) have to be carefully chosen. The switching characteristics of the inverter are the fundamental parameters to characterize the integrated circuits technology. Thus, to ensure the reliability and performance of a CMOS circuit, the switching speed of the circuit must be optimized very early in the design phase. The objective of this paper is to optimally design a symmetric switching CMOS inverter, which is of importance to the CMOS IC industry. The right and effective choice of the transistors' channel widths (W) and lengths (L) and the output load capacitance (C_L) is very important and essential in designing a CMOS inverter with symmetric switching characteristics, i.e, symmetric output voltage waveform. Performing such a task by hand turns out to be tedious and time consuming. Recently, to make the design more reliable and flexible, different evolutionary optimization algorithms have been used in the design of CMOS inverter with symmetric switching characteristics [1-8]. In [1, 2], particle swarm optimization (PSO) was used in the design of a CMOS inverter having symmetrical waveform of output voltage with equal rise time (t_r) and fall time (t_f) and equal propagation delay times (t_{pHL} and t_{pLH}). In [3-8], B. P. De *et al* applied

several nature-inspired optimization methods on the design of symmetric switching CMOS inverter. Specifically, PSO with constriction factor and inertia weight approach (PSOCFIWA), differential evolution (DE), craziness based particle swarm optimization (CRPSO), hybrid harmony search with differential evolution (HS-DE), PSO with aging leader and challenger (ALC-PSO), and firefly algorithm (FA) were applied in [3-8], respectively. In each paper, it was shown that the used optimization method gives better results than those presented in [1, 2] and those obtained using the real-coded genetic algorithm (RGA).

In this paper, the optimal transient behavior of CMOS inverter is investigated using a new optimization method, the symbiotic organisms search (SOS) [9]. Specifically, SOS is used to find the optimum values of the widths and lengths of the channels of the NMOS and PMOS transistors along with the load capacitor that give symmetric switching characteristics of the inverter. Since its invention, SOS has been used in several optimization problems in engineering. In [10, 11], SOS has been applied to solve the optimal power flow and the economic emission load dispatch problems. Moreover, SOS has been successfully applied in the optimal design of linear and circular antenna arrays [12, 13], in solving the load frequency control problem [14], in structure optimization problems [15], in solving unconstrained function optimization [16], in the design of an improved 3D Turbo Code [17], in the optimal design of analog active filters [18], in the synthesis of elliptical antenna arrays [19], and other engineering areas [20-24]. The results indicated that SOS gives very good results and is very competitive with the state of the art for the solution of these problems, which motivated this work. The SOS results will be compared with those obtained using different optimization methods available in the literature [3-7]. Moreover, SOS results will be compared with those obtained by the PSPICE simulator.

The paper is divided as follows: SOS algorithm is briefly described in Section 2. The interested reader may consult the above references for more details on the SOS algorithm. In Section 3, the formulation of the problem is presented, while numerical results and discussion are presented in Section 4. Section 5 concludes the paper.

2.   SOS ALGORITHM

Symbiotic Organisms Search (SOS) has been recently proposed in [9] as an effective evolutionary global optimization method. It is based on the symbiotic interaction strategies between different organisms in an ecosystem. Symbiosis describes the interaction between different species living together. In SOS, the three main types of symbiotic relationships (mutualism, commensalism, and parasitism) are modeled using simple mathematical expressions. An example of mutualism, in which both organisms benefit from such a relationship, is the interaction between bees and flowers. An example of commensalism, where one species benefits from the interaction, while the other does not gain or lose anything, is the relationship between remora fish and sharks. An example of parasitism, where the relationship benefits one species while the other species is harmed, is the interaction between the plasmodium parasites and humans. One of the main advantages in using SOS is the fact that it is free from any tuning parameters. In its implementation, SOS requires only common controlling parameters like population size (number of organisms) and maximum number of iterations.

Similar to most evolutionary optimization methods, the first step in SOS is to randomly generate an initial population of organisms (which is called the ecosystem). The number of generated organisms N is called the population size. Thus, the ecosystem is an NxD matrix, where D is the dimension of the problem (i.e., the number of design variables). An organism (individual) X_i ($i=1, ..., N$) within the ecosystem (population) is a real-valued vector with D elements which represents a single possible solution to a specific optimization problem. The second step is to associate each organism X_i of the ecosystem (i.e., each row in the ecosystem matrix) with a certain value of the fitness function which imitates the degree of adaptation to the desired objective. Afterwards, new solutions are created by simulating the above mentioned three symbiotic interactions between the organisms in the ecosystem. Each organism in the ecosystem randomly interacts with other organisms through all these three phases and new organisms are created which have better fitness function value. This process is repeated until a specific termination criterion (e.g., maximum number of iterations) is fulfilled. In each iteration, every organism X_i ($i=1, ..., N$) goes through the following three phases:

2.1. Mutualism phase

Firstly, another organism X_j is randomly selected to "mutually" interact with X_i. Then, X_i and X_j are updated according to the following expressions [9]:

$$X_{i\,(new)} = X_{i\,(old)} + rand(0,1) \times [X_{best} - (Mutual\ Vector) \times (BF_1)] \tag{1}$$

$$X_{j\ (new)} = X_{j\ (old)} + rand(0,1) \times [X_{best} - (Mutual\ Vector) \times (BF_2)] \tag{2}$$

where the *(Mutual Vector)* is the average between the two interacting organisms, i.e., *(Mutual Vector)* = $(X_i + X_j) / 2$.

In the above equations, X_{best} is the best organism (that is the one that resulted in the minimum value of the fitness function), and *rand(0,1)* is a random number in the range [0, 1]. The parameters BF_1 and BF_2 are called benefit factors which represent the level of benefit between the organisms as the organisms may fully or partially benefit from such a relationship. They are randomly determined as either 1 or 2. At the end of the mutualism phase, the new organisms are accepted, and taken to the next phase, if they give better (that is less) fitness function values than their pre-interaction fitness values.

2.2. Commensalism phase

Similar to the first phase, an organism X_j is randomly selected to interact with X_i. However, in this phase, X_i tries to benefit from the interaction, while X_j itself is not affected at all. Then, X_i is updated using the following expression [9]:

$$X_{i\ (new)} = X_{i\ (old)} + rand(-1,1) \times [X_{best} - X_j] \tag{3}$$

The term $(X_{best} - X_j)$ reflects the benefit provided by X_j *to* X_i to increase its degree of adaptation (to the highest degree represented by X_{best}) such that it can better survive in the ecosystem. The new organism $X_{i\ (new)}$ is adopted, and taken to the third phase, if it results in a better fitness function value. If not, then X_i remains the same, which means that it didn't gain anything from such an interaction with X_j.

2.3. Parasitism phase

In this last third phase, a new vector called "*Parasite Vector*" is created by mutating X_i in a random dimension using a random number. Then, an organism X_j is randomly selected to serve as a host to the parasite vector, and the fitness function for both organisms is evaluated. If the parasite vector has a better fitness, then X_j is removed and its position is taken over by the parasite vector. On the other hand, if X_j has a better fitness value, then it will be immune from the parasite vector, and will not be affected by its existence. In this case, the parasite vector is simply removed from the ecosystem. A pseudo-code and a detailed description of the SOS algorithm can be found in [9].

3.   PROBLEM FORMULATION

One of the most important issues in the CMOS inverters is the symmetric switching speed. In CMOS inverters circuits, the switching speed is affected by the output resistance of the high state of the P-MOSFET, the capacitive load, the length and width of the PMOS and the NMOS channels and on both the input voltage V_{in} and the power supply. The switching speed can be improved by decreasing the output resistance and this can be done by increasing the power supply. However, the larger power supply will result in increased power dissipation. In this paper, the CMOS inverter is optimized to operate at 2.5 V, and the switching operation is analyzed to determine the rise and fall times (t_r and t_f) and propagation delays (t_{pHL} and t_{pLH}). The PSPICE schematic diagram of the CMOS inverter is shown in Figure 1, while the input and the output waveforms are shown in Figure 2.

As well known, the fall time (t_f), is the time required for the output voltage to drop from $V_{90\%}$ of the output voltage level down to $V_{10\%}$, and the rise time (t_r) is defined as the time required for the output voltage to rise from $V_{10\%}$ to $V_{90\%}$. They can be computed as follows [3-8]:

$$t_f = \frac{C_L}{\mu_n C_{OX}(\frac{W}{L})_n(V_{DD} - V_{tn})}\left[\frac{2(V_{tn} - 0.1V_{DD})}{(V_{DD} - V_{tn})} + \ln(\frac{\left(2(V_{DD} - V_{tn})\right) - 0.1V_{DD}}{0.1V_{DD}})\right] \tag{4}$$

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

On the other hand, the high-to-low propagation delay time (t_{pHL}) is defined as the time delay between the $V_{50\%}$ transition of the rising input voltage and the $V_{50\%}$ transition of the falling output voltage,

and can be computed as [3-8]:

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

Lastly, the low-to-high propagation delay time (t_{pLH}) is defined as the time delay between the $V_{50\%}$ transition of the falling input voltage and the $V_{50\%}$ transition of the rising output voltage, and can be computed as [3-8]:

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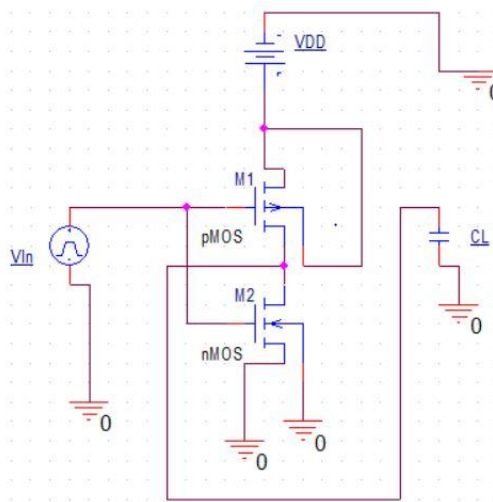

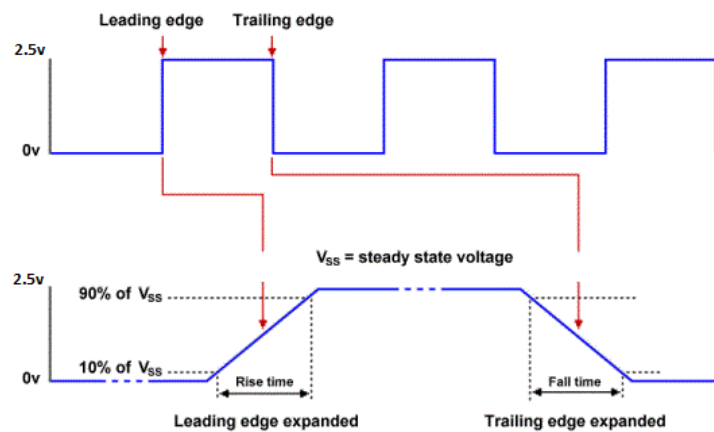Figure 1. CMOS inverter circuit (as drawn in PSIPICE)

Figure 2. Input and output voltage waveforms of the CMOS inverter

The cost (fitness) function to be minimized can be written as [3-8]:

$$J = 10 \, log_{10}(CF) \quad (8)$$

where

$$CF = \left| \left(t_f \left(C_L, \left(\frac{W}{L} \right)_n \right) \right) - \left(t_r (C_L, (\frac{W}{L})_p) \right) \right| + \left| \left(t_{pHL} (C_L, (\frac{W}{L})_n) \right) - \left(t_{pLH} (C_L, (\frac{W}{L})_p) \right) \right| \tag{9}$$

Subject to the following constraints:

$$(t_f)_{min} \leq t_f \leq (t_f)_{max}$$
$$(t_r)_{min} \leq t_r \leq (t_r)_{max}$$
$$(t_{pHL})_{min} \leq t_{pHL} \leq (t_{pHL})_{max}$$
$$(t_{pLH})_{min} \leq t_{pLH} \leq (t_{pLH})_{max}$$
$$(C_L)_{min} \leq C_L \leq (C_L)_{max}$$
$$((\frac{W}{L})_n)_{min} \leq (\frac{W}{L})_n \leq ((\frac{W}{L})_n)_{max}$$
$$((\frac{W}{L})_p)_{min} \leq (\frac{W}{L})_p \leq ((\frac{W}{L})_p)_{max}$$

Thus, the objective of the optimization is to get, as much as possible, a symmetric response of the inverter, such that the fall and rise times are equal, and the propagation delays are equal. There are three variables in the optimization problem, namely, the load capacitance C_L, and the ratio between the widths and the lengths of the channels, $(\frac{W}{L})_n$ and $(\frac{W}{L})_p$. Thus, the dimension of the optimization problem is 3. The minimum and maximum values of the above constraints are shown in Table 1, where similar to [3-8], 8 different design sets are considered in this paper. Moreover, similar to [1-8], TSMC 0.25 micron fabrication technology is considered, where: V_{DD}= 2.5 V, V_{tn} = 0.3655 V, V_{tp} = 0.5466 V, $\mu_n C_{OX}$ = 243.6 μA/V^2, and $\mu_p C_{OX}$ = 51.6 μA/V^2.

Table 1. Constraints values for the 8 design sets [3-8]

Design set no.	C_L (pF)	$(W/L)_n$	$(W/L)_p$	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
2	0.1–5.1	1.6–7.1	1.8–18	1.1–15	1.1–15	0.5–8	0.5–8
3	0.47–2	1.4–6.7	3.2–38	0.5–12	0.5–12	0.2–9	0.2–9
4	0.1-1.1	1.2–7	1.5-17.5	0.5–5	0.5–5	0.2–4	0.2–4
5	0.2–14	1.9–5.0	2.7–17	0.7–6	0.7–6	0.4–5	0.4–5
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
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
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

4. NUMERICAL RESULTS

For each design set in Table 1, SOS code with population size=10 and maximum number of iterations=500 is run for 50 independent times. Each run took around 8 seconds on an HP ProBook 6550b Laptop with "Intel Core i7 CPU M 620 @ 2.67 GHz" and "4 GB RAM". Table 2 shows the best results obtained using SOS for each design set, where it can be seen that a symmetric output waveform is obtained with equal fall and rise times and almost equal propagation delays. In Table 3, SOS results are compared with the best-known results found in the literature. Namely, SOS results are compared with those obtained using particle swarm optimization with constriction factor and inertia weight approach (PSOCFIWA) [3], differential evolution (DE) [4], craziness based particle swarm optimization (CRPSO) [5], hybrid harmony search with differential evolution (HS-DE) [6], and particle swarm optimization with aging leader and challenger (ALC-PSO) [7]. It should be mentioned that we took the values of the input parameters (C_L, $(\frac{W}{L})_n$, $(\frac{W}{L})_p$) from the tables in [3-7] and computed the corresponding CF values ourselves, as it is believed that the reported CF values in [3-7] are not right. Even though SOS does not get the least error in all design sets, it does compete very well with other well-developed methods. It should be also mentioned that, unlike other methods, SOS is free of tuning parameters; one has to set the values of the population size and the maximum number of iterations only. Moreover, the basic SOS algorithm is used here without any modifications, in contrary to [3, 5, 7] where PSO was modified to overcome some of its inherent drawbacks. Table 4 shows some statistical measures of the SOS results. It can be clearly seen that the SOS is a robust optimization method as it obtains a very small value for the standard deviation (compared, for example, with the one reported in [5] using CRPSO). Figure 3 shows the convergence curve for the design set number 8 which

obtains the least value for CF. Very good convergence is reached within only 100 iterations. Figure 4 shows box-and-whisker plot for the same design set. In this figure, median (middle of dataset) is represented by the red line, i.e., 50% of the data are larger than this value. Upper and lower ends of the box represent the 75th and 25th percentiles, i.e., 25% of the data are larger than the upper end, while 25% of the data are less than the lower end. The difference between the upper and lower ends of the box (i.e., the length of the box) is known as the "interquartile range (IQR)". The whiskers are lines extending from the ends of the box to show the extent of the rest of the data. The end of the upper whisker indicates the largest value in the dataset (excluding upper outliers), while the end of the lower whisker indicates the smallest value in the dataset (excluding lower outliers). Outliers are data with values beyond the ends of the whiskers, and are indicated with red "+" sign. An outlier is any value that lies more than one and a half times the length of the box from either end of the box.

Table 2. SOS results

Design set no.	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)	t_{pHL} (ns)	t_{pLH} (ns)	CF (ps)
1	0.7428	3.49783	18.67556	1.1785	1.1785	0.5	0.51835	18.35
2	0.71593	3.3713	18	1.1785	1.1785	0.5	0.51835	18.35
3	0.60366	6.7	35.77255	0.5	0.5	0.21213	0.21992	7.79
4	0.10812	1.2	6.40702	0.5	0.5	0.21213	0.21993	7.79
5	0.32279	1.9	10.14445	0.9428	0.9428	0.4	0.41469	14.68
6	0.3866	3.03417	16.2	0.7071	0.7071	0.3	0.31101	11.01
7	0.42687	4.73782	25.29611	0.5	0.5	0.21213	0.21992	7.79
8	0.39487	7.30448	39	0.3	0.3	0.12728	0.13195	4.67

Table 3. Comparison of SOS-obtained *CF* values (in ps) with other methods in the literature.
The least *CF* value is in boldface

Design set no.	SOS	CRPSO [5]	RGA [5]	HS-DE [6]	PSO-CFIWA [3]	DE [4]	ALC-PSO [7]
1	18.35	31.84	65.77	17.74	18.26	37.26	17.83
2	18.35	35.06	75.09	17.92	18.17	39.5	17.85
3	7.79	15.37	77.54	7.81	7.79	16.01	7.8
4	7.79	15.12	36.83	7.79	7.79	12.81	13.68
5	14.68	30.04	69.28	14.19	14.46	42.13	14.19
6	11.01	20.59	70.92	22.95	11.53	23.58	10.96
7	7.79	8.84	69.25	7.81	7.9	15.97	7.79
8	4.67	7.91	68.78	4.67	4.67	9.35	4.67

Table 4. Statistics of SOS CF values out of 50 runs for each set

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 8 (CRPSO) [5]
Minimum (ps)	18.3538	18.3538	7.78695	7.78695	14.683	11.01227	7.78695	4.67217	5.2594
Maximum (ps)	18.537	18.8628	7.847	7.88833	14.8548	11.04983	7.97586	4.76056	15.3432
Mean (ps)	18.3853	18.3815	7.7943	7.7983	14.6958	11.01903	7.81208	4.68196	10.0572
Standard deviation (x 10^{-14}) (s)	4.1548	7.9977	1.3174	1.9595	2.66413	0.89952	3.684	1.53946	108.15

The 0.25 micron CMOS inverter circuit is simulated using the new version of PSPICE (V 17.7). The results of eight different design set are obtained and shown in Table 5, considering the same SOS optimized values as shown in Table 2 of the important parameters, output load capacitor and the geometry of the transistor, as inputs for PSPICE simulation. Figures 5 and 6 shows the timing diagrams for the rise and fall times and the propagation delay times, respectively, for design set number 8 using PSPICE simulation. A reasonable similarity between the SOS results in Table 2 and PSPICE results in Table 5 for the rise and fall times can be observed. However, larger group delays are predicted using PSPICE. This is due to the fact that more complex expressions are used in PSPICE to evaluate the rise and fall times and the propagation delays, while simple approximate expressions are used here assuming long channel transistors and an ideal step input voltage. Similar observation has been noted in [3-7]. Clearly, these approximate expressions underestimate the propagation delays, and thus, we are now exploring the use of more complex expressions, like the improved ones very recently proposed in [25].

It is worth mentioning that PSPICE results are obtained using V17.7 with the following settings:

a. Simulation profile:
   Analysis type = Transient
   TSTOP: 8 seconds

Start saving data after: 0 seconds
Maximum step size: 1ms
SKIPBP: true
All other simulation parameters are left to their default values (RELTOL, VNTOL, ABSTOL, CHGTOL,
GMIN, ITL1, ITL2, ITL4, TNOM, THREADS, Speed level, SKIPBP).
b. CMOS technology used: MbreakP and MbreakN to simulate the 0.25 μm TSMC fabrication technology
with the parameters mentioned in the paper ($\mu_n C_{ox}$, V_{tn}, V_{tp}, $(W/L)_n$, $(W/L)_p$).
c. Input source: (VPULSE) source with the following characteristics:
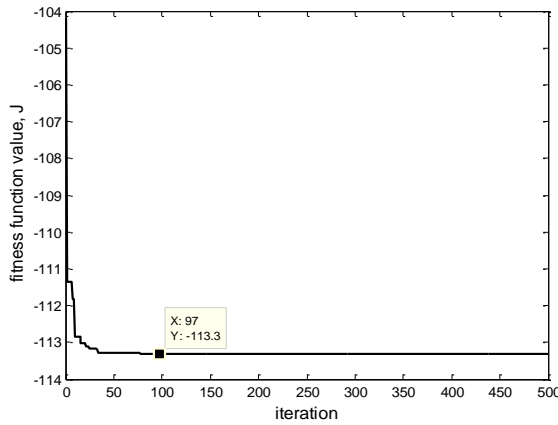V1 = 0; V2 = 2.5 = VDD; TD = 0; TR = 1 n; TF = 1 n; PW = 1; PER = 2

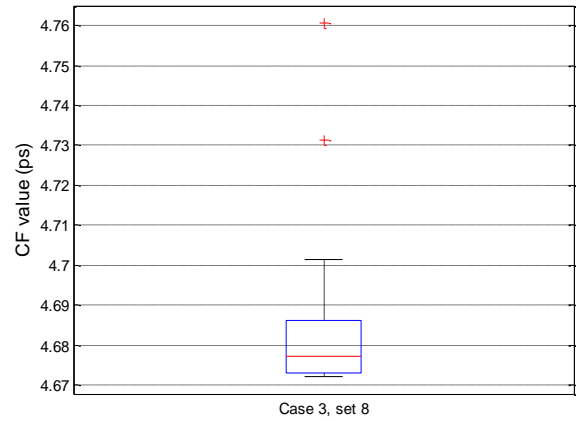Figure 3. Convergence curve for the SOS results for the design set number 8

Figure 4. Box-and-whisker plot for the SOS results for design set number 8

Table 5. PSPICE results using Vin "High" = V_{DD}

| Design set no.	C_L (pF)	$(W/L)_n$	$(W/L)_p$	t_f (ns)	t_r (ns)	t_{pHL} (ns)	t_{pLH} (ns)	CF (ns)
1	0.7428	3.49783	18.67556	1.35524	0.96916786	1.1966	1.1166	0.46607
2	0.71593	3.3713	18	1.35524	0.96916741	1.0971	1.1627	0.45167
3	0.60366	6.7	35.77255	0.76703569	0.64575389	0.887348	0.602467	0.40616
4	0.10812	1.2	6.40702	0.76704227	0.64575767	0.914830	0.627832	0.40828
5	0.32279	1.9	10.14445	1.15790	0.83511820	1.0645	0.715316	0.67197
6	0.3866	3.03417	16.2	1.00256	0.70468631	0.961739	0.673224	0.58639
7	0.42687	4.73782	25.29611	0.7670353	0.64575389	0.878689	0.566099	0.43387
8	0.39487	7.30448	39	0.679782	0.668853	0.795644	0.512026	0.29455
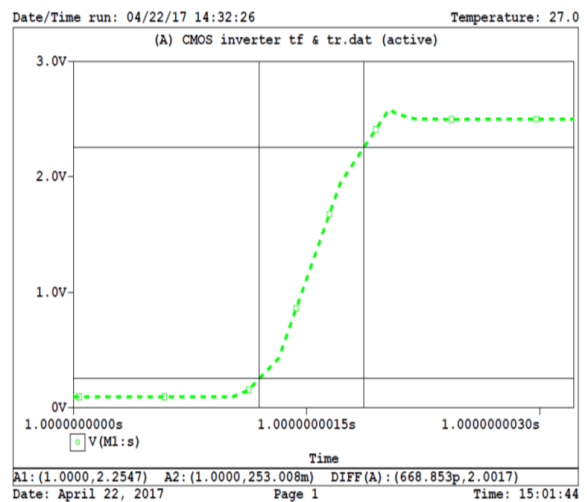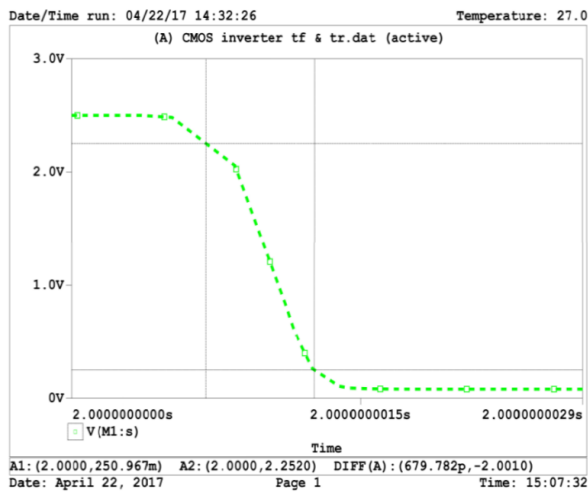
Figure 5. Fall time and rise time timing diagrams obtained using PSPICE for design set 8
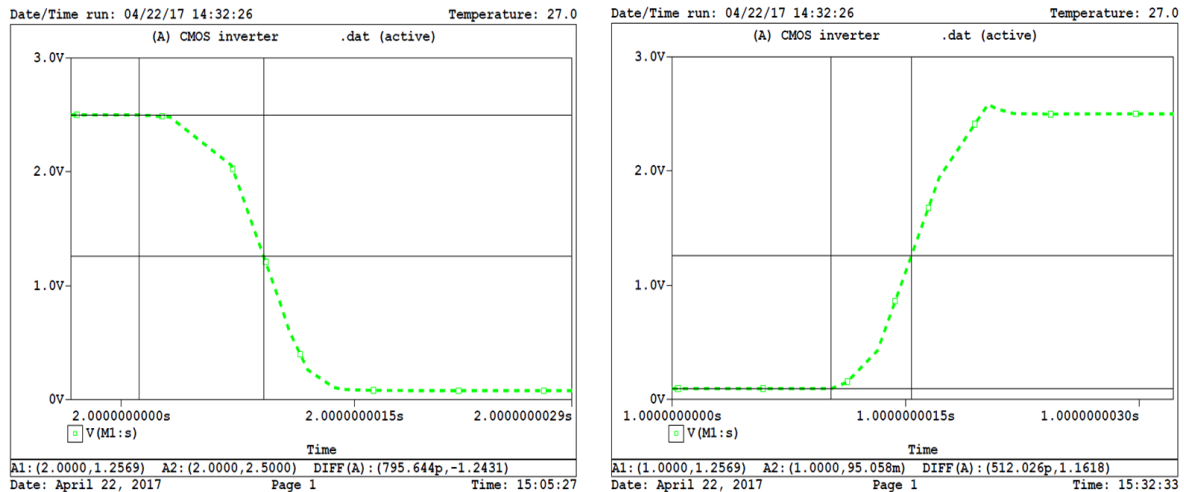
Figure 6. Propagation delays timing diagrams obtained using PSPICE for design set 8

5.   CONCLUSION

In this paper, SOS algorithm was used in the optimal design of a CMOS inverter with symmetric switching characteristics. SOS results were compared with those obtained using other evolutionary optimization methods available in the literature. The comparison showed that SOS is a robust, simple, and fast converging method. For all 8 design sets, SOS method performed very well, which clearly shows the effectiveness of the SOS method. As mentioned before, compared to other methods, SOS has the advantage of being free of tuning parameters; one just needs to set the values of the population size and the maximum number of iterations.

REFERENCES

[1]   R. Vural, O. Der, and T. Yildirim, "Investigation of particle swarm optimization for switching characterization of inverter design," *Expert. Syst. Appl.*, vol. 38, pp. 5696–5703, 2011.
[2]   R. Vural, O. Der, and T. Yildirim, "Particle swarm optimization based inverter design considering transient performance," *Digital Signal Process.*, vol. 20, pp. 1215–1220, 2010.
[3]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Design of symmetric switching CMOS inverter using PSOCFIWA," *2014 International Conference on Communications and Signal Processing (ICCSP)*, pp. 1818-1824, 2014.
[4]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal CMOS inverter design using differential evolution algorithm," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 2, pp. 219-241, 2015.
[5]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal high speed CMOS inverter design using craziness based Particle Swarm Optimization Algorithm," *Open Eng.,* vol. 5, pp. 256–273, 2015.
[6]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal design of high speed symmetric switching CMOS inverter using hybrid harmony search with differential evolution," *Soft Computing*, vol. 20, no. 9, pp. 3699-3717, Sep. 2016.
[7]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "PSO with aging leader and challengers for optimal design of high speed symmetric switching CMOS inverter," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 4, pp. 1403–1422, Aug 2017.
[8]   B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Design of Optimal CMOS Inverter for Symmetric Switching Characteristics Using Firefly Algorithm with Wavelet Mutation," *International Journal of Swarm Intelligence Research (IJSIR),* vol. 5, no. 2, pp. 29-64, 2014.
[9]   M. Cheng and D. Prayogo, "Symbiotic organism search: a new metaheuristic optimization algorithm," *Computers and Structures*, vol. 139, pp. 98-112, Jul. 2014.
[10] S. Duman, "Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones," *Neural Computing and Applications*, vol. 28, no. 11, pp. 3571–3585, Nov. 2017.

[11] M. Dosoglu, U. Guvenc, S. Duman, Y. Sonmez, and H. Kahraman, "Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems," *Neural Computing and Applications*, vol. 29, no. 3, pp. 721–737, Feb. 2018.

[12] N. Dib, "Design of linear antenna arrays with low side lobes level using symbiotic organisms search," *Prog. Electromagn. Res. B*, vol. 68, pp. 55-71, 2016.

[13] N. Dib, "Design of Planar Concentric Circular Antenna Arrays With Reduced Side Lobes Level Using Symbiotic Organisms Search," *Neural Computing and Applications*, vol. 30, no. 12, pp. 3859-3868, Dec. 2018.

[14] D. Guha, P. Roy, and S. Banerjee, "Symbiotic organism search algorithm applied to load frequency control of multi-area power system," *Energy Syst*, vol. 9, no. 2, pp. 439–468, May 2018.

[15] G. Tejani, V. Savsani, and K. Vivek, "Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization," *Journal of Computational Design and Engineering*, vol. 3, pp. 226–249, 2016.

[16] S. Nama, A. Saha, and S. Ghosh, "Improved symbiotic organisms search algorithm for solving unconstrained function optimization," *Decision Science Letters*, vol. 5, pp. 361−380, 2016.

[17] S. Banerjee, and S. Chattopadhyay, "Power Optimization of Three Dimensional Turbo Code Using a Novel Modified Symbiotic Organism Search (MSOS) Algorithm," *Wireless Pers Commun*, vol. 92, pp. 941–968, 2017.

[18] N. Dib and B. El-Asir, "Optimal Design of Analog Active Filters Using Symbiotic Organisms Search," International *Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 31, no. 5, pp. 1-15, Sep. 2018.

[19] N. Dib, A. Amaireh, and A. Al-Zoubi, "On the Optimal Synthesis of Elliptical Antenna Arrays," *Int. J. of Electronics*, vol. 106, no. 1, pp. 121-133, Jan. 2019.

[20] G. Tejani, N. Pholdee, S. Bureerat, D. Prayogo, and A. Gandomi, "Structural optimization using multi-objective modified adaptive symbiotic organisms search," *Expert Systems with Applications*, vol. 125, pp. 425-441, 2019.

[21] D. Prayogo, and C. Kusuma, "Optimization of resource leveling problem under multiple objective criteria using a symbiotic organisms search," *Civil Engineering Dimension*, vol. 21, pp. 43-49, 2019.

[22] G. Tejani, N. Pholdee, S. Bureerat, and D. Prayogo, "Multiobjective adaptive symbiotic organisms search for truss optimization problems," *Knowledge-based systems*, vol. 161, pp. 398-414, 2018.

[23] D. Prayogo, M. Cheng, F. Wong, D. Tjandra, and Duc-Hoc Tran, "Optimization model for construction project resource leveling using a novel modified symbiotic organisms search," *Asian Journal of Civil Engineering*, vol. 19, pp. 625-638, 2018.

[24] D. Prayogo, Min-Yuan Cheng, and H. Prayogo, "A novel implementation of nature-inspired optimization for civil engineering: a comparative study of symbiotic organisms search," *Civil Engineering Dimension*, vol. 19, no. 1, pp. 36-41, 2017.

[25] E. Zafarkhah, M. Maymandi-Nejad, and M. Zare, "Improved accuracy equation for propagation delay of a CMOS inverter in a single ended ring oscillator," *International Journal of Electronics and Communications (AEÜ)*, vol. 71, pp. 110–117, Jan. 2017.