Research Article

Automatic Circuit Design and Optimization Using Modified PSO Algorithm

Subhash Patel[1],* and Rajesh A Thakker[2]

[1]*Indus University, Ahmedabad, India.*
[2]*Vishwakarma Government Engineering College, Ahmedabad, India*

_____

Abstract

In this work, we have proposed modified PSO algorithm based optimizer for automatic circuit design. The performance of the modified PSO algorithm is compared with two other evolutionary algorithms namely ABC algorithm and standard PSO algorithm by designing two stage CMOS operational amplifier and bulk driven OTA in 130nm technology. The results show the robustness of the proposed algorithm. With modified PSO algorithm, the average design error for two stage op-amp is only 0.054% in contrast to 3.04% for standard PSO algorithm and 5.45% for ABC algorithm. For bulk driven OTA, average design error is 1.32% with MPSO compared to 4.70% with ABC algorithm and 5.63% with standard PSO algorithm.

Keywords: CMOS, Optimizer, Op-Amp, bulk-driven OTA, Automatic circuit design, Low power, Evolutionary algorithm, ABC algorithm, PSO algorithm, modified PSO algorithm.

_____

1. Introduction

With the advancement in CMOS technology, the size of MOSFET device shrinks. The smaller device size brings more non-linearity in the characteristics of the device. This brings the problem of device sizing in design and optimization of the analog circuits. Under such conditions, the traditional design approach based on the analytical calculations followed by the simulation fails to provide time efficient ASIC development cycle. On the other hand, with increasing power of the modern CPU, it is possible to use optimization algorithms effectively for circuit design.

The optimization methods such as linear programming, integer programming, non-linear programming and quadratic programming are examples of deterministic programming techniques. These techniques require having differentiable objective function in order to obtain global solution of the optimization problem [1]. For the circuit design problem, the formulation of the accurate objective function is extremely difficult. The dynamic programming is another optimization technique based on stochastic programming models that guarantees global solution of optimization problem. However the computational efforts required to solve the problem increase exponentially as the size of problem increases [2]. Another class of the optimization algorithm includes computational intelligence based techniques such as Genetic Algorithm (GA), particle swarm optimization (PSO), artificial bee colony algorithm (ABC). They do not guarantee the global solution of the given problem. However, they do not require complex mathematical calculations and hence easy to implement using programming languages. In many multi-objective problems such as CMOS circuit design, we depend upon the simulation result and it is very difficult to form exact objective function mathematically. Under such

situations, the evolutionary algorithms become obvious choice.

The use of the A-NSGAII algorithm was demonstrated in [3] to design RF low noise amplifier, leapfrog filter and ultra wideband LNA. In [4], single ended telescopic Op-Amp is designed using the genetic algorithm. The PSO algorithm and its variants are used for automatic design of low-power low-voltage CMOS circuits [5]. In [6], the performances of the genetic algorithm (GA), PSO algorithm and Simulated Annealing algorithm are compared by designing the LC voltage controlled oscillator. An evolutionary approach is used to design RF low noise amplifier in [7]. The chaotic DE algorithm, standard DE algorithm, ABC algorithm and PSO algorithm are used in [8] to design miller OTA and their performances are compared.
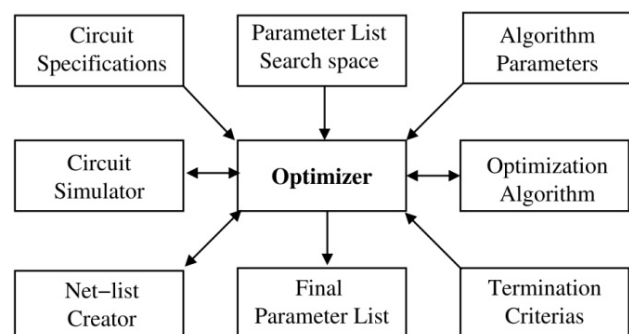

Fig. 1. Conceptual block diagram of optimizer

In this work, we have demonstrated application of ABC algorithm and standard PSO algorithm for automatic circuit design and proposed a modified PSO algorithm. The performances of these three algorithms are compared by designing two-stage CMOS op-amp and bulk driven OTA in 130nm CMOS technology. This paper is organized as follows: In Section 2, the overview of the optimizer is given. Section 3 discusses ABC algorithm, standard PSO algorithm

and modified PSO algorithm. The automatic circuit design examples are illustrated in section 4. Finally conclusions are drawn in section 5.

2. Optimizer for automatic analog circuit design

The optimizer utilizes the circuit simulator and optimization algorithm to design a circuit with desired specifications. The optimizer provides proper coordination between the circuit simulator and optimization algorithm by generating circuit net-list according to the parameters generated by optimization algorithm, initializing the circuit simulation, analyzing the simulator output and providing necessary data to the optimization algorithm to generate new set of parameters. The conceptual block diagram of the optimizer is illustrated in Fig 1. First various circuit specifications are decided. The various circuit parameters with their upper and lower bounds are estimated. Generally for the CMOS circuits, the circuit parameters are width and length of various MOS transistors. With this information optimization algorithm generates the set of circuit parameters and according to this, the circuit for simulator is generated and simulated against pre-determined test cases. The simulation results are analyzed and error is calculated. According to calculated error, new set of parameters is generated by the optimization algorithm. The aim of the optimizer is to reduce the error. Here, we have used root mean square (RMS) error. The RMS error in percentage f_e can be given by,

$$RMS\ error\ fe(\%) = \sqrt{\sum_{i=1}^{N} E_i} \times 100 \qquad (1)$$

$$E_i = \begin{cases} 0 & , if\ i^{th}\ specification\ is\ satisfied \\ \left(\frac{OS_i - DS_i}{DS_i}\right)^2 & , otherwise \end{cases} \qquad (2)$$

where, N is total number of specifications, DS_i is i^{th} desired specification and OS_i is i^{th} obtained specification from simulation. The RMS error gives equal weight to all the specifications. Thus, optimizer tries to satisfy all the specifications equally.

3. Optimization algorithms

3.1 Artificial Bee Colony (ABC) algorithm
The artificial bee colony algorithm is based on swarm optimization technique. It simulates the intelligent behavior of the artificial bees searching food for finding the global solution of the given optimization problem [9]. The performance of the ABC algorithm is compared with other evolutionary algorithms in [9] and [1] by solving different benchmark functions.

In ABC algorithm, the swarm of the artificial bees is divided in two groups: Employee bees and Onlooker bees. The Employee bees are later converted in to Scout bees. Consider an optimization problem with N dimensions and take swarm size *2M*. The number of the employee bees and onlooker bees are same. This leads to M number of Employee bees and same number of Onlooker bees. The algorithm starts with the random initialization. Initially M numbers of food sources are picked up randomly and each one is evaluated for its fitness. Each food source represents potential solution to problem. Mathematically, each food source is model by N-dimensional vector. Thus i^{th} food source can be represented by,

$$X_i = \{X_{i1}, X_{i2}, X_{i3}, \dots \dots, X_{iN}\} \qquad (3)$$

Each artificial bee, tries to improve its food source by sharing the information with other bees. During the process of food source improvement, only one dimension of food source X_i i.e. X_{ij} is selected randomly and updated at a time. The single iteration of the algorithm can be divided in three phases: Employee bee phase, Onlooker bee phase and Scout bee phase. During Employee bee phase, each employee bee tries to find new food source V_i around assigned food source X_i, by updating single dimension of X_i, as follows,

$$V_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \qquad (4)$$

with, $j \in \{1,2,3,....,N\}$ and selected randomly. X_{kj} is j^{th} dimension of neighbor food source and $k \in \{1,2,3,....,M\}$ and selected randomly. Φ_{ij} is uniformly distributed random number between -1 and 1. The new food source V_i is evaluated for its fitness and greedy selection is applied between X_i and V_i.

Onlooker bees select certain food sources where the probability of finding better food source is higher and try to improve only these selected food sources same as Employee bees. Such probability associated with i^{th} food source is given by,

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \qquad (5)$$

where, f_i is a fitness associated with i^{th} food source and can be calculated by,

$$f_i = \begin{cases} \frac{1}{1 + fe_i}, & fe_i > 0 \\ 1 + |fe_i|, & fe_i < 0 \end{cases} \qquad (6)$$

Thus, in Employee bee phase each food source undergoes improvement process, while only selected food sources are improved during Onlooker bee phase. In Scout bee phase, the food sources which failed to improve after certain predetermined trials are abandoned and instead of that new food sources are picked up randomly from the search space. The ABC algorithm is good at exploration but poor at exploitation [10].

3.2 Standard PSO algorithm
The Particle Swarm Optimization (PSO) algorithm finds a solution of the given optimization problem by simulating the social behavior of species [11]. In PSO algorithm, each particle of the swarm represents the solution candidate. The particles of the swarm are assumed to move in the search space with the velocity associated with them. Each particle of swarm remembers the best location ever visited by it and the overall best position visited by the all particles. For the optimization problem with N dimensions, the position of the particle and velocity associated with them can be modeled by N dimensional vectors. Let us consider the current position of the ith particle is $X_i(t)$ and velocity associated with is $V_i(t)$. The new position of the ith particle, $X_i(t + 1)$ can be calculated by,

$$\begin{aligned} V(t + 1) = &\ w \cdot V_i(t) \\ &+ C_1 \cdot R_1 \cdot (P_i(t) - X_i(t)) \\ &+ C_2 \cdot R_2 \cdot (P_g(t) - X_i(t)) \end{aligned} \qquad (7)$$

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \qquad (8)$$

where, w is an inertia weight factor, R_1 and R_2 are the uniformly distributed random numbers between 0 and 1, C_1 and C_2 are acceleration constants, P_i represents the best position ever visited by i^{th} particle and P_g is an overall best position. C_1 and C_2 are also called cognitive and social parameter respectively. Sometimes, instead of choosing a constant value of w, it is varied linearly between w_{up} and w_{low} with iterations as follows,

$$w(t) = w_{up} - (w_{up} - w_{low}) \cdot \frac{t}{t_{max}} \qquad (9)$$

where, t is a current iteration and t_{max} is maximum allowed iterations and $w_{up} > w_{low}$. Such variations in value of w, promotes exploration in early stage of the optimization. Another concept is used widely with the standard PSO algorithm is velocity clamping. When absolute velocity of the particle in any dimension crosses predetermined upper limit V_{max}, it is clamped at V_{max}. The parameters of the standard PSO algorithms are w, C_1 and C_2. According to nature of optimization problem, values of these parameters can be tuned [12].

3.3 Modified PSO algorithm

In standard PSO algorithm, the new position of the particle depends upon the current position of the particle, local best position P_i, global best position P_g and the velocity of the particle. Initially, when algorithm starts with random initialization, the swarm of the particle is diverse and as the algorithm progresses, the swarm loses its diversity. The swarm diversity is a very important factor for the PSO algorithm [13]. As long as swarm is diverse, the new solutions will be generated. As the swarm loses its diversity, the movement of the particle in the search space also reduces. When swarm diversity is lost completely, the velocity of particles become zero and their position remain unchanged. Thus, no new solutions will be generated.
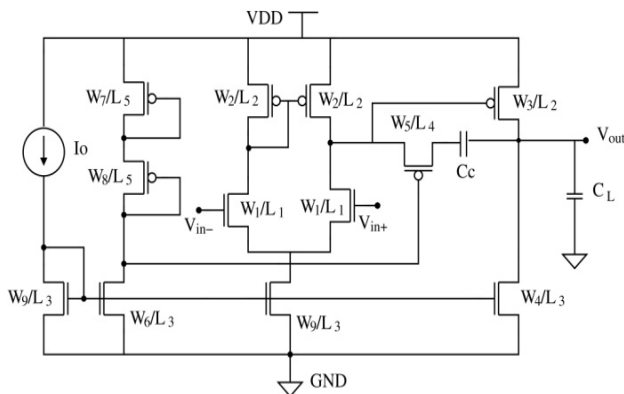

Fig. 2. Two-stage operational amplifier : circuit diagram

In modified PSO algorithm, we emphasize on maintaining swarm diversity. The movement of the particle is decided by Equ. 7 and 8. When the global solution does not improve after the algorithmic iteration, velocity of each particle is examined. If the absolute value of particular velocity component for all the particles fall bellow predetermined value V_{min}, only that component for all particles is re-initialized along with corresponding velocity component. When there is need to re-initialize more than one component, one is selected randomly and re-initialization is carried out only for that dimension along with corresponding velocity

component. After such partial re-initialization of swarm, next partial re-initialization, if required, is carried out only after the fixed number of iterations (I_s). The partial re-initialization of the swarm is not carried out over the complete search space. The partial re-initialization area in percentage for corresponding dimension around the global best position P_g is determined as follows,

$$A = S + (1 - S) \cdot fgb \times 100 \qquad (10)$$

Where, S is constant value between 0 and 1, fgb is a RMS error for current global best position P_g. Such re-initialization scheme helps to maintain swarm diversity and improves the exploration capacity of the algorithm. Moreover, it also helps to avoid trapping of algorithm in local minima.

4. Circuit design examples

To compare the performances of ABC, PSO and MPSO algorithms, two analog CMOS circuits namely two-stage operational amplifier (Op-Amp) and bulk driven operational trans-conductance amplifier (OTA) are designed in 130nm CMOS technology. For MPSO algorithm, the values of parameters are set as follows: $w_{up} = 0.9$, $w_{low} = 0.2$, $C_1 = 0.49$, $C_2 = 1.99$, $I_s = 3$ and $S = 0.25$. For PSO algorithm, the designs of Op-Amp and OTA are carried out using two different set of parameters. For first case, the value of different parameters are set as suggested in [5] and they are $w_{up} = 0.9$, $w_{low} = 0.4$, $C_1 = 1.49$ and $C_2 = 1.49$. We call this case of PSO algorithm PSO1. For second case, we set the values of algorithmic parameters same as MPSO algorithm: $w_{up} = 0.9$, $w_{low} = 0.2$, $C_1 = 0.49$ and $C_2 = 1.99$. This second case of PSO is called PSO2. Each circuit is designed independently 25 times with MPSO, ABC, PSO1 and PSO2 algorithm. For the Op-Amp design, swarm size is set to 15. For the algorithm termination criteria, maximum numbers of circuit evaluations are used. This limit is set to 5000 for Op-Amp design case. For OTA design, swarm size is 12 and maximum circuit evaluations are 10000. For the circuit simulation, NG-SPICE simulator is utilized. Whole optimizer along with algorithm is implemented with help of C++. The experiment is conducted on computer having following major specifications: Processor - AMD-8350, CPU clock rate - 4GHz, RAM - 4GB, OS - Ubuntu 12.04, Kernel - 3.13.0.66-generic.

4.1 Two stage Op-Amp

The two stage CMOS operational amplifier (Op-Amp) is one of the most widely used analog circuit. It is a building block of ADC, amplifiers, mixer and signal conditioning circuits. The circuit of op-amp is illustrated in Fig 2 [14]. The set of desired specification is described in Table 1. The design parameters are width and length of the transistors, value of the current source I_0 and capacitor C_C. The circuit is designed in 130nm to drive load of 0.05pF. The search space i.e. upper and lower bounds on width and length of transistor, value of I_0 and value on C_C, is illustrated in Table 2. The circuit is designed twenty-five times. Table 3 shows the average of obtained specifications and RMS error f_e over 25 independent design runs. Table 4 and 5 show the best and worst obtained specifications, respectively. The variations in the RMS error f_e with the circuit evaluations is shown in Fig. 3. The Table 6 shows the number of times zero RMS error is obtained or in other words all the specifications are satisfied
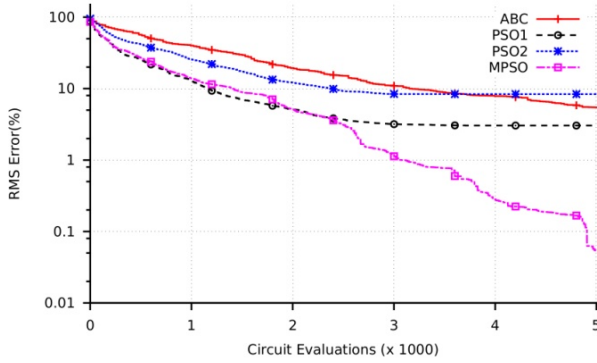
Fig. 3. Two stage Op-Amp : Average RMS error vs circuit evaluations

Table 1. Op-Amp : Desired Specifications

Specification	Desired value
Gain	> 80 dB
UGB	> 100 MHz
Phase Margin (PM)	> 62°
Power consumption (PC)	< 20 μW
Rise Slew Rate (RSR)	≥60 V/μS
Fall Slew Rate (FSR)	≥60 V/μS
PSRR	≥80 dB
CMRR	≥75 dB

Table 2. Op-Amp : Search space for design variables

Parameter	Search Space
W_1 to W_9 (μm)	0.2 to 10
L_1 to L_5 (μm)	0.2 to 1
I_0 (μm)	0.5 to 10
C_C (pF)	0.001 to 1
VDD (V)	1.2

.

Table 3. Op-Amp : Average of obtained specifications, RMS error and CPU time for design over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	80.2	78.9	78.5	76.8
UGB (MHz)	105.5	98.2	102.3	99.08
PM (^{o})	63.2	61.6	61.8	60.9
PC (μW)	18.9	19.6	19.1	19.2
RSR (V/μS)	76.7	72.0	71.5	74.0
FSR (V/μS)	63.3	65.2	62.7	64.4
PSRR (dB)	87.4	85.9	85.9	86.5
CMRR (dB)	76.8	76.6	77.9	76.4
f_e (%)	0.054	5.45	3.04	8.37
CPU Time (S)	281.6	401.3	354.2	361.9

From the obtained results following observations can be made.

- In 25 independent design runs of Op-Amp, the average RMS error is only 0.054% with MPSO in contrast with 5.45% with ABC, 3.04% with PSO1 and 8.37% with PSO2.
- Out of 25 design runs, MPSO designs Op-Amp 21 times with zero RMS error and thus obtaining all specifications while ABC is successful only 2 times, PS01 and PSO2 are for 7 times.
- The worst obtained design with MPSO RMS error is 0.59% which is much less than 17.2% with ABC, 12.2% with PSO1 and 20.1% with PSO2.
- The average CPU time required to design Op-Amp with MPSO is 281.6 seconds. That is 401.3 seconds for ABC, 354.2 seconds for PSO1 and 361.9 seconds for PSO2.

Table 4. Op-Amp : Best obtained specifications and RMS error over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	81.2	80.3	80.0	80.0
UGB (MHz)	104.9	111.7	100.4	100.1
PM (^{o})	63.3	63.2	63.7	62.5
PC (μW)	19.8	19.5	19.7	19.3
RSR (V/μS)	80.8	76.6	74.6	72.1
SR (V/μS)	60.9	60.6	60.0	62.4
PSRR (dB)	87.3	81.2	104.7	89.2
CMRR (dB)	75.4	80.8	75.4	76.21
f_e (%)	0.0	0.0	0.0	0.0

The obtained results for Op-Amp design show the robustness of the modified PSO algorithm compared to standard PSO algorithm and ABC algorithm
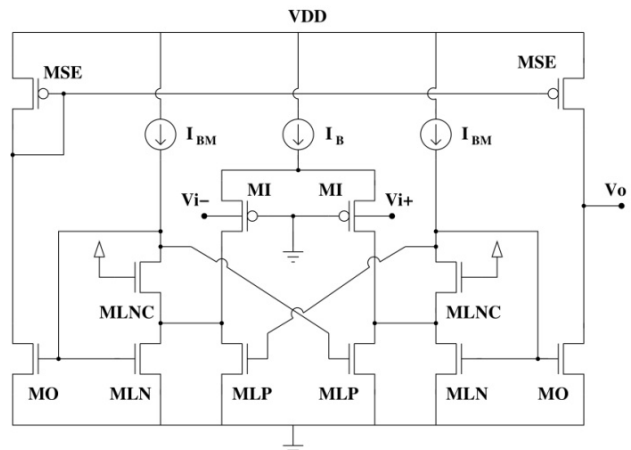

Fig 4. Bulk-driven OTA : circuit diagram

Table 5. Op-Amp : Worst obtained specifications for design over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	79.7	74.5	75.2	64.6
UGB (MHz)	99.5	90.4	93.5	98.3
PM (^{o})	61.8	60.0	59.2	59.7
PC (μW)	19.7	19.9	21.2	20.1
RSR (V/μS)	81.5	53.0	59.4	99.5
FSR (V/μS)	68.6	58.5	59.1	63.6
PSRR (dB)	105.5	77.4	77.6	77.2
CMRR (dB)	74.9	80.1	84.5	72.5
f_e (%)	0.59	17.2	12.2	20.1

Table 6. Op-Amp : Number of times all the specifications are satisfied

| Algorithm	No of times zero RMS error is obtained
MPSO	21
ABC	2
PSO1	7
PSO2	7

4.2 Bulk-driven OTA
In the bulk-driven circuit technique, the voltage signal is applied at the bulk terminal of the MOSFET. In low voltage application, i.e. supply voltage is less than 1V; this technique enhances the performance of the circuit by overcoming the limitations imposed by the threshold voltage. Another advantage of bulk-driven technique for low voltage

application is that, it does not require any modification in the structure of MOSFET [16-18].

The operational trans-conductance amplifier (OTA) is used widely to drive large capacitive load. The circuit diagram of the bulk driven OTA is shown in Fig.4. This circuit is proposed in [15]. The set of desired specifications and simulation results obtained with 350nm technology in [15] are shown in Table 7.The design parameters with their upper and lower bounds are illustrated in Table 8. The circuit is design twenty-five times indecently in 130nm technology to drive load of 15pF. The obtained results are illustrated in Tables 9, 10, 11. The variation in average RMS error with the circuit evaluations is illustrated in Fig. 5.

Table 7. Bulk-driven OTA : Desired Specifications

Specification	Desired value	Results of [15]
Gain (dB)	> 45	41.7
UGB (MHz)	> 15	10
PM (^{o})	> 60°	58
PC (μW)	< 200	200
RSR (V/μS)	≥10	8.9
FSR (V/μS)	≥10	8.3
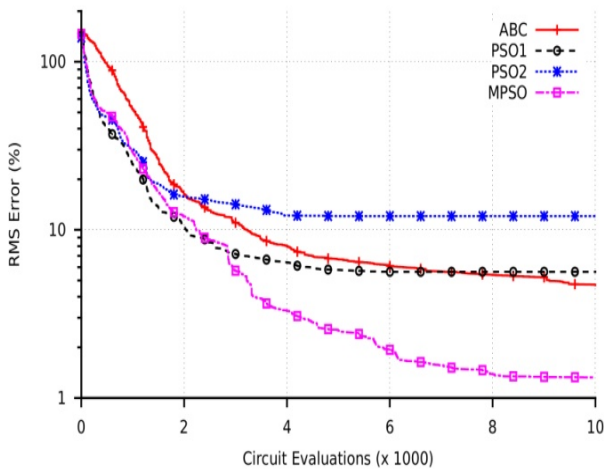
Fig. 5. Bulk-driven OTA : Average RMS error vs circuit evaluations

Table 8. Bulk-driven OTA : Search space for design variables

Parameter	Search Space
Width of all transistors (μm)	1 to 100
Length of all transistors (μm)	0.2 to 5
I_B (μA)	3 to 50
I_{BM} (μA)	3 to 50
VDD (V)	0.6

Table 9. OTA : Average of obtained specifications, RMS error and CPU time for design over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	44.5	42.1	42.6	40.9
UGB (MHz)	18.7	16.6	18	16.4
PM (^{o})	60.5	58.9	59.4	57.3
PC (μW)	124.5	117.9	120.8	124.7
RSR (V/μS)	13.5	12.2	12.5	13.5
FSR (V/μS)	14.2	12.5	12.9	15.4
f_e (%)	1.32	4.70	5.63	120.7
CPU Time (S)	342.1	462.1	405.8	432.6

Table 10. OTA : Best obtained specifications and RMS error for design over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	45.0	44.9	46.6	46.1
UGB (MHz)	15.1	22.1	16.6	17.7
PM (^{o})	61.2	59.9	61.1	60.7
PC (μW)	94.6	90.8	81.1	82.1
RSR (V/μS)	13.5	10.6	10.7	14.4
FSR (V/μS)	13.7	20.5	10.6	15.9
f_e (%)	0.0	0.22	0.0	0.0

In the design of the bulk-driven OTA, the performance of modified PSO is found better than standard PSO and ABC algorithms. The average RMS error is 1.32% for the modified PSO algorithm, 5.63% for the PSO1 algorithm, 12.07% for PSO2 and 4.70% for the ABC algorithm. In twenty-five independent runs, modified PSO algorithm has designed OTA successfully 13 times, while ABC, PSO1 and PSO2 algorithms are only successful zero time, 6 times, 3 times respectively. The average CPU time for designing bulk driven OTA is also less in case of the modified PSO algorithm. With automatic circuit design technique, obtained results are better than obtained with manual design of [15].

Table 11. OTA : Worst obtained specifications and RMS error for design over 25 independent runs

	MPSO	ABC	PSO1	PSO2
Gain (dB)	42.5	40.0	35.9	39.0
UGB (MHz)	18.5	14.8	23.2	12.6
PM (^{o})	59.6	56.3	57.3	44.3
PC (μW)	149.2	100.9	136.1	90.5
RSR (V/μS)	12.8	15.8	12.6	9.9
FSR (V/μS)	9.89	11.1	9.8	84.5
f_e (%)	5.51	12.62	20.66	33.04

Table 12. OTA : Number of times all the specifications are satisfied

Algorithm	No of times zero RMS error is obtained
MPSO	13
ABC	0
PSO1	6
PSO2	3

5. Conclusion

In this work, we have demonstrated application of the computational intelligence based techniques such as standard PSO algorithm and ABC algorithm to solve the multi-dimension and multi-constrained problem of the analog CMOS circuit design and synthesis. We have also proposed modified PSO algorithm for the automatic circuit design. The performances of the standard PSO algorithm, ABC algorithm and modified PSO algorithm are compared by designing the two stage op-amp and bulk driven OTA. The results show that the performance of the modified PSO algorithm is far better than that of standard PSO algorithm and ABC algorithm. For the two-stage Op-Amp, the average RMS error with modified PSO algorithm is 0.054% while that is 5.45% for ABC, 3.04 % for the PSO1 algorithm and 8.37% for the PSO2 algorithm. In case of the bulk driven OTA, the average RMS error with modified PSO algorithm is almost three and half times smaller than that of ABC algorithm.

_____

References

1. P. Civicioglu, E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms", Artificial Intelligence Review 39 (4) (2013) 315-346

2. Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems", Evolutionary Computation, IEEE Transactions on 12 (2) (2008) 171-195.

3. G. Nicosia, S. Rinaudo, E. Sciacca, "An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization", Knowledge-Based Systems 21 (3) (2008) 175-183.

4. M. Barros, J. Guilherme, N. Horta, "Analog circuits optimization based on evolutionary computation techniques", INTEGRATION, the VLSI journal 43 (1) (2010) 136-155.

5. R. A. Thakker, M. S. Baghini, M. B. Patil, "Automatic design of low-power low-voltage analog circuits using particle swarm optimization with re-initialization", Journal of Low Power Electronics 5 (3) (2009) 291-302.

6. P. Pereira, M. Kotti, H. Fino, M. Fakhfakh, "Metaheuristic algorithms comparison for the lc-voltage controlled oscillators optimal design", in: Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference on, IEEE, 2013, pp. 1-6.

7. A. Somani, P. P. Chakrabarti, A. Patra, "An evolutionary algorithm-based approach to automated design ofanalog and rf circuits using adaptive normalized cost functions", Evolutionary Computation, IEEE Transactions on 11 (3) (2007) 336-353.

8. S. L. Sabat, S. K. Kumar, S. K. Udgata, "Differential evolution and swarm intelligence techniques for analog circuit synthesis", in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE, 2009, pp. 469-474.

19.

9. D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm", Journal of global optimization 39 (3) (2007) 459-471.

10. G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization", Applied Mathematics and Computation 217 (7) (2010) 3166-3173.

11. J. Kennedy, "Particle swarm optimization", in: Encyclopedia of Machine Learning, Springer, 2010, pp. 760-766.

12. K. E. Parsopoulos, "Particle Swarm Optimization and Intelligence: Advances and Applications: Advances and Applications", IGI Global, 2010.

13. M. Pant, T. Radha, V. P. Singh, "A simple diversity guided particle swarm optimization", in: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE, 2007, pp. 3294-3299.

14. P. Allen, D. Holberg, "CMOS Analog Circuit Design", OUP USA, 2012.

15. J. Carrillo, G. Torelli, R. Perez-Aloe, J. Duque-Carrillo, "1-v rail-to-rail bulk-driven cmos ota with enhanced gain and gain-bandwidth product", in: Circuit Theory and Design, 2005. Proceedings of the 2005 European Conference on, Vol. 1, 2005, pp. I/261-I/264.

16. B. J. Blalock, P. E. Allen, "A low-voltage, bulk-driven mosfet current mirror for cmos technology", in: Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on, Vol. 3, IEEE, 1995, pp. 1972-1975.

17. B. J. Blalock, P. E. Allen, G. Rincon-Mora, et al., "Designing 1-v op amps using standard digital cmos technology", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on 45 (7) (1998) 769-780.

18. R. He, L. Zhang, "Evaluation of modern mosfet models for bulk-driven applications", in: Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on, IEEE, 2008, pp. 105-108.