Automatic Design of a Telescopic Amplifier Based on Particle Swarm Optimization

Luiz Antônio da Silva Junior, Robson André Domanski, Anderson de Paula Fortes and Alessandro Girardi

***Abstract*—A fast and efficient design methodology for CMOS operational amplifiers is mandatory when there is a reduced time-to-market. In this case, the automation of this process must be considered. However, the n-dimensional design space is highly nonlinear and it is difficult to find an optimized solution. In this context, we developed a tool for the automatic sizing of analog integrated circuits, called UCAF. This tool uses artificial intelligence (AI) techniques to explore solutions according to design requirements. This paper presents the analysis of test sets based on Particle Swarm Optimization (PSO) technique, applied to the design of a telescopic operation amplifier in 0.18μm technology, as well as a brief comparison with the Simulated Annealing (SA) technique. The generated circuit presented better performance when sized with PSO in terms of area and dissipated power.**

***Index Terms*—Optimization, Analog design, PSO.**

I. Introduction

THE fast development of microelectronics in recent decades enabled electronic circuits even more integrated. Although most of the functions in integrated circuits are for processing in the digital domain, the analog circuits are required at the interface between the electronic system and the ”real world” [1].

Both the input of a circuit and its output are usually analog, then the signal is transmitted and / or originated by sensors, antennas and other means of transmission. The analog circuits are widely used in systems applications, such as telecommunications and robotics [1], [2].

With the increased integration of these circuits, the analog circuit design becomes more increasingly complex, making it a challenging and time-consuming part. As it is required to size each device that compose the circuit, and the design space is highly non-linear, this process is considered a bottleneck of the project in a mixed-signal circuit [3]–[5].

Roughly, the analog circuit design is usually composed by two main steps. The first step is the choice of one the several possible circuit architectures and topologies. The second step is to determine the values of circuit parameters (e.g., resistor and capacitor values, geometries and devices, such as gate length of the width MOS transistors) [4], [6].

Luiz Antônio da Silva Junior is an Electrical Engineering undergraduate student at UNIPAMPA campus Alegrete. (e-mail:luizantoniojr.ladsj@gmail.com)

Robson André Domanski is a graduate student by the Electrical Engineering Graduate Program at UNIPAMPA campus Alegrete. (e-mail: robsondomanski@gmail.com)

Anderson de Paula Fortes is a graduate student by the Electrical Engineering Graduate Program at UNIPAMPA campus Alegrete. (e-mail: fortes.cc@gmail.com

Alessandro Girardi is PhD in Microelectronics at UFRGS (2007). Currently he is with UNIPAMPA in Alegrete-RS, Brazil, where he is associate professor. (email: alessandro.girardi@unipampa.edu.br)

Thus, it is clear that the development of CAD (computer aided design) tools is required to handle the complexity of analog circuit design scale in an acceptable time, making automatic scaling analog circuits an important research topic for two decades [2], [4]–[6].

Usually the design of an analog circuits needs several iterations, mostly based on trial and error and electrical simulations. However, combinatorial computational optimization methods may be used to reduce design time and optimize circuit parameters [6].

Algorithms based on metaheuristics and populations have been widely used for solving highly-dimensional non-linear searching problems. It is very difficult to find optimal solutions in these problems, because the design space is very large and requires high computational effort. Examples of optimization heuristic algorithms are Particle Swarm Optimization (PSO), Simulated Annealing (SA) and Genetic Algorithms (GA), which can be easily adapted for modeling different applications.

The analog circuit sizing can be modeled as an optimization problem by defining a cost function to be minimized. This cost function can be a weighted sum of circuit performance metrics (for example, voltage gain, dissipated power, etc). The design free variables are the gate length (L) and width (W) of the transistors that compose the circuit.

In analog circuits, each transistor must be sized separately, although individual performance affects the overall response of the circuit. An automatic design methodology must consider the characteristics of the entire circuit in order to guarantee the achievement of performance requirements [3].

Some previous works describe strategies and methodologies for dealing with this problem [7], [8], [9], [10], [11].

In this context, we propose an automatic synthesis procedure for basic analog building blocks which can size transistors dimensions W and L in a short design time using ordinary computational resources.

We developed a tool, called UCAF, for automatic sizing of analog circuits considering yield optimization [3]. This tool transforms the analog sizing problem into an optimization problem, exploring the design space through meta-heuristics such as Simualted Annealing (SA) and Genetic Algorithms (GA) and the technique discussed in this work - Particle Swarm Optimization (PSO). PSO is a population-based global optimization algorithm based on the simulation of the social behavior of birds within a flock [12]. This algorithm belongs to the family of Swarm Intelligence (SI) based optimization techniques. The methodology in PSO is to use a swarm of m particles that move through the design space in the direction of the global minimum. Some previous works described strategies

and methodologies for dealing with the PSO [13]–[16]. The cost function evaluation is performed by SPICE electrical simulation. The cost function includes a set of specifications to be optimized and a set of constraints to be reached.

This paper is organized as follows: Section II presents the circuit design methodology; Section III presents the PSO heuristics; Section IV shows the design results for a Telescopic operational amplifier; finally, Section V presents some concluding remarks.

II. Design Optimization Methodology

The developed tool is based on a methodology for the design of analog integrated circuits via optimization techniques using electrical simulation for evaluation. This methodology is based on the definition of a cost function that models the sizing problem of an specific analog block in a generic optimization problem. Fig. 1 demonstrates the implemented automatic design flow. The optimization method receives random values as initial input, the design requirements and the process technology parameters used for the design. The optimization algorithm assigns the values for the circuit variables on each iteration. The cost function is evaluated by electrical simulations over different testbenches for extracting circuit performances. The optimization methodology performs the process until satisfying an arbitrary stop criterion.

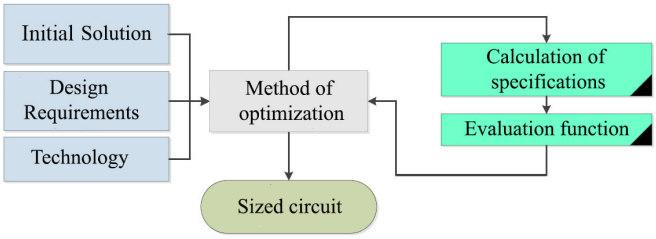

Fig. 1. Basic UCAF automatic design flow

The circuit electrical characteristics are modeled in a cost function, which is implemented as an equation in terms of circuit variables. The electrical characteristics may be the power consumption, the circuit gate area, the voltage gain, etc., or even a combination of these. The minimization of the cost function is performed by heuristics with values provided by electrical simulations through an external electrical simulator.

The cost function used in this work is given by Eq. 1. In this equation, $O_n(X)$ is the circuit *n-th* specification, $C_m(X)$ is the *m-th* constraint, and w_n and w_m are the weights (weighting factor) for the objectives and constraints, respectively. To choose which specifications are optimization targets and which one are constraints, the tool receives the configuration of each of the specifications given by the designer.

$$f_c = \sum_{n=1}^{N} w_n.O_n(X) + \sum_{m=1}^{M} w_m.C_m(X) \qquad (1)$$

Figure 2 shows some modular functions which represent the way that UCAF tool is implemented and organized. According to [3], using modularity we can have a high degree of configurability because each function can be substituted by a similar one without loosing functionality. Any module can be changed independently of the remaining functions.

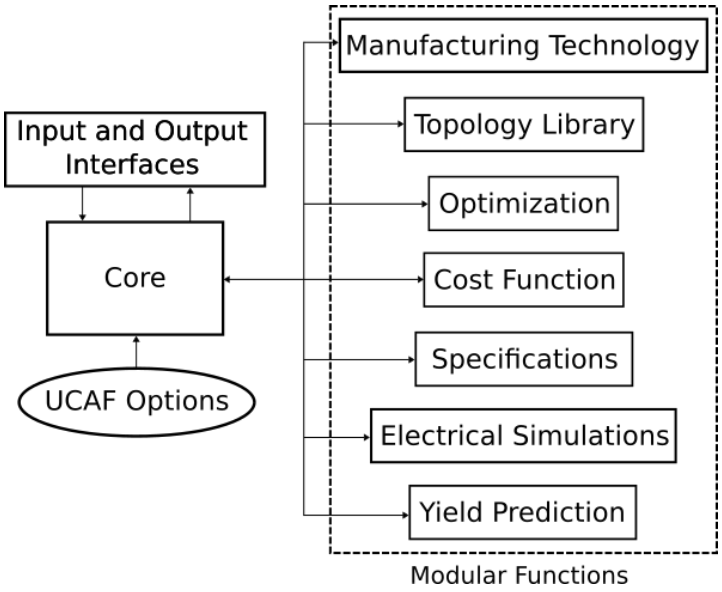

Fig. 2. UCAF Modular Functions

The main function is the ”Core” module, and it is responsible for creating and organizing a new design, creating design folders, setting the modular functions, writing the simulation file and performing other important tasks.

The interface between the design and the fabrication technology is made by the function ”Manufacturing Technology”. The goal of this function is to read and configure the parameters of simulation models from the design kit provided by the foundry.

The ”Topology Library” function contains all the analog blocks previously inserted in the tool which are saved in a cell library. These cells can be reused for different design specifications.

An optimization algorithm that will guide the design space exploration is implemented in the ”Optimization” function. More details about the optimization algorithm used in this work will be given in Section III.

The Cost Function already mentioned and described by Eq. 1 is implemented in the ”Cost Function” which has the goal to represent the design as a single-objective minimization problem.

Finally, the ”Electrical Simulation” and ”Specification” functions are both necessary to measure the values of the circuit specifications, thereby the cost function can be evaluated.

The choice of the optimization algorithm is an important factor for the quality of the optimal solution in an optimization problem. SA was implemented previously in the UCAF tool [3]. This work describes the implementation of the same problem using PSO.

III. Particle Swarm Optimization

The Particle Swarm Optimization algorithm is an evolutionary technique, and was inspired by the social behavior of

birds [17]. PSO is based on studies in artificial life and social psychology, as well as in engineering and computer science. It utilizes a population of individuals, referred to as particles, flying through the hyperspace with some initial velocity. In each iteration, the velocities of the particles are adjusted taking into account the best historical position (global best) of the particles and their best neighborhood position (local best). These positions are determined according to a predefined function fitness [18]. Then, the movement of each particle evolves naturally to an optimal or near-optimal solution. Therefore, PSO is an heuristic proper for the searching of global and local optimization of non-linear systems.

Generally, in population-based search optimization methods, it is necessary a high number of particles in the first search step, thus allowing the exploration of the full range of the search space. On the other hand, during the latter part of the search, when the algorithm converges to an optimum solution, the fine adjustment of the solutions is important to set the global optimum.

Therefore in optimization methods, a proper control of global exploration and local level is crucial to find the best solution more efficiently. Shi and Eberhart [19] introduced the concept of inertial weight in the original version of PSO in order and balance local and global search for the optimization process. The mathematical representation of this concept is given by Eq. 2.

$$w = (w_{initial} - w_{final}) * \frac{(MAXITER - iter)}{MAXITER} + w_{final} \quad (2)$$

Where:

w	inertia coefficient
$w_{initial}$	initial value of the inertia.
w_{final}	final value of the inertia.
$MAXITER$	maximum number of iterations.
$iter$	current iteration number.

Through empirical studies, Shi and Eberhart [20] found that the optimal solution can be improved by varying the value of 0.9 at baseline to 0.4 at the end of the study for most problems.

This modification of the original PSO brought changes in the formula so that the calculated speed is given by Eq. 3 [20].

$$vel_x(i+1) = w_x * vel_x(i) + C1 * rand * (Pbest_x - present_x) + C2 * rand * (Gbest_x - present_x) \quad (3)$$

Were:

vel_x	velocity of the particle x.
w_x	inertia coefficient of particle x.
$C1$ and $C2$	cognitive and social acceleration constants.
$rand$	random number in the range of [0, 1].
$Pbest_x$	best position of particle x.
$Gbest$	best overall position.
$present_x$	current position of particle x.

At each iteration, the position of the particle is updated as follows:

$$present_x(i+1) = present_x + vel_x(i+1) \quad (4)$$

Here, i is the current iteration and $i+1$ is the next one.

The Algorithm 1 shows the pseudocode of the PSO algorithm that we propose for the sizing of analog circuits.

Algorithm 1 Pseudocode of the PSO algorithm.

```
1: Generate initial population.
2: Generate initial fitness
3: Obtain initial values for Pbest and Gbest.
4: function PSO
5:     for x < MAXITE do
6:         For each particle, compute objective function
7:         Update Pbest and Gbest.
8:         Evaluate the fitness of each particle. ▷ Eq. 3
9:     end for
10: end function
```

It is important to remember that global search algorithms, such as Simulated Annealing [21] are of great importance for solving problems effectively and efficiently. However, the PSO is implemented without such hybridization techniques, in order to demonstrate that the mechanisms of social learning of PSO alone may result in obtaining good solutions for nonlinear optimization problems.

IV. Application Example

As an design example, we performed the automatic design of a Telescopic Operational Amplifier [22] in XFAB CMOS 0.18 μm technology with V_{DD} = $|V_{SS}|$ equal to 0.9V. The technology also defines the minimum values of the circuit dimensions, which are $L = 0.8\mu m$ and $W = 0.22\mu m$. The schematics of the amplifier is shown in Fig. 3.

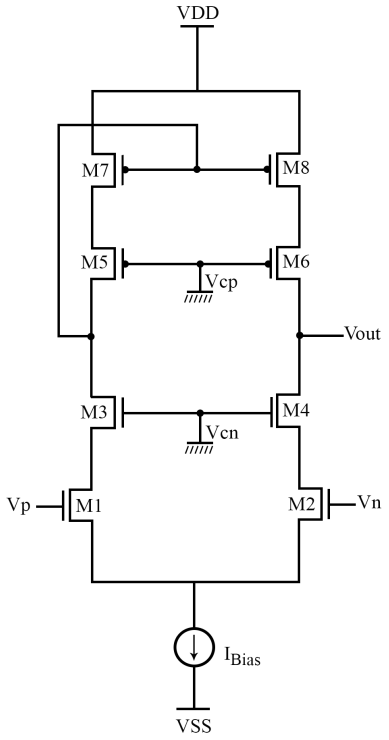

Fig. 3. Schematics of the Telescopic Amplifier.

Eight transistors of the circuit must be sized, so that parameters like gate width (W) and length (L) of are obtained. Due to the constructive form of the circuit, some equalities between the transistors can be assigned. Thereby, the amplifier has the following equalities: $M1 = M2$, $M3 = M4$, $M5 = M6$, $M7 = M8$. In addition to the sizing of the transistors to the circuit design, the current source I_{Bias} , the Vp and Vn input voltages must be scaled. So, this design consists of 11 free variables: $W1$, $L1$, $W3$, $L3$, $W5$, $L5$, $W7$, $L7$, I_{Bias}, Vn and Vp.

The main specifications of this circuit are low-frequency voltage gain (Av0), Gain-Bandwidth Product (GBW), slew-rate (SR), phase margin (PM), common mode input range (ICMR), silicon area and dissipated power.

In this methodology the optimization procedure has a cost function according Eq. 1. The objectives of the cost function are the minimization of power consumption and gate area. The constraints are defined as Av0, GBW, SR, PM and ICMR.

In order to check the circuit performance, the optimization process uses an interface with the external electrical simulator Synopsys HSPICE®.

Through an AC, DC or transient analysis it is possible to estimate the value of each specification of the circuit simulating determined circuit testbench.

To measure the specifications of Av0, GBW, and the PM it is necessary to perform an AC analysis using the testbench presented in Fig. 4. The results of this analysis can be plotted as a Bode diagram. The Av0 and GBW specifications can be extracted through the gain curve. In the same way, from phase curve we can obtain the phase margin specification.

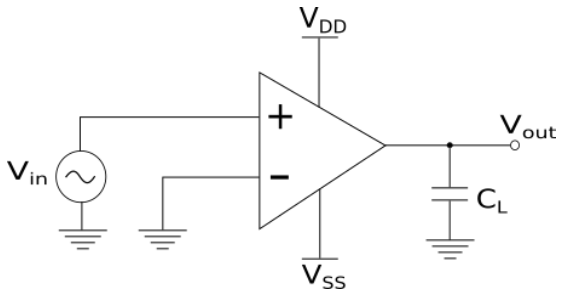

Fig. 4. AC testbench configuration.

Fig. 5 shows the amplifier connected in a unity gain configuration, which is used to obtain the ICMR specification. In this case a DC analysis is performed and the input voltage is varied from a minimum to a maximum level. Positive and negative values of ICMR are obtained from the output when the gain is linear.

Using the same ICMR circuit configuration we can measure the response speed of an amplifier (Slew Rate). As shown in Fig. 6, a step voltage is used as input in this simulation, and through a transient analysis we can obtain the value of SR by verifying the raise and fall rate of the output voltage level.

We conducted some tests to find the best configuration of the PSO algorithm applied to this problem, as shown below.

First, we choose a value of 100 for the number of iterations and we vary the population size from 10 to 240. The results are shown in Fig. 7, Fig. 8 and Fig. 9.

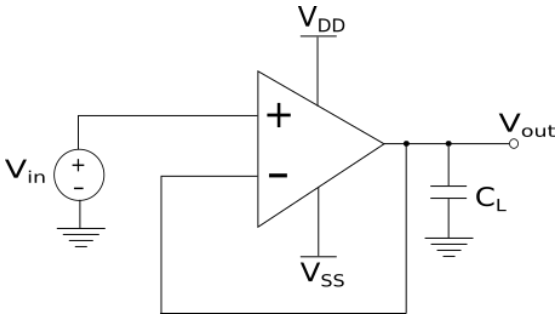

Fig. 5. ICMR testbench configuration.

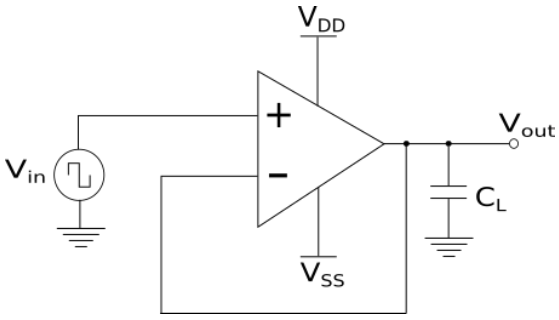

Fig. 6. SR testbench configuration.

For testing purposes we set a fixed value for the seed of the random number generator (RNG) used by RAND function in order to have the same random sequence for all test cases.

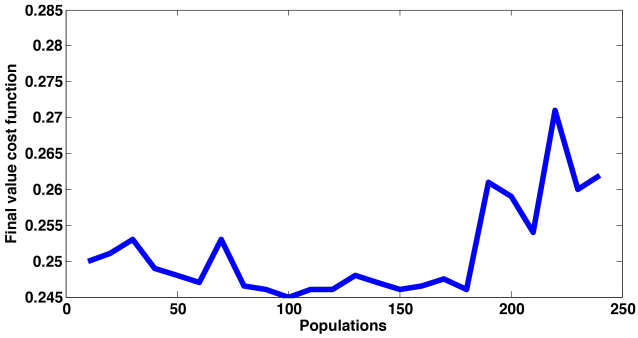

Fig. 7. Values for cost function after 100 iterations according to population size.

Analyzing the set of figures, where the size of the iteration was chosen as 100, the best result is found when the population size reaches the value of 100. At this point we find the lowest value for the cost function, power dissipation and gate area.

Knowing the best population size, we can evaluate the optimum number of iterations, which is the stop criterium of the algorithm. Fig. 10, Fig. 11 and Fig. 12 demonstrate the results for the optimized cost function value for different iteration numbers. The number of iterations was varied from 10 to 240 for a fixed population size of 100. We consider again the same random sequence for all test cases.

The analysis of the figures demonstrate that the cost function value decreases considerably in about in 60 iterations. Beyond this point, the cost function does not have a significant

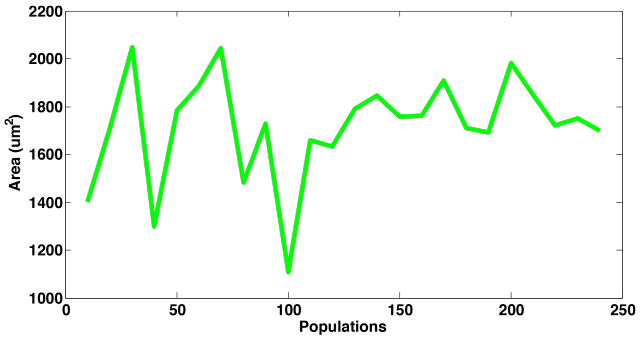

Fig. 8. Values for gate area after 100 iterations according to population size.

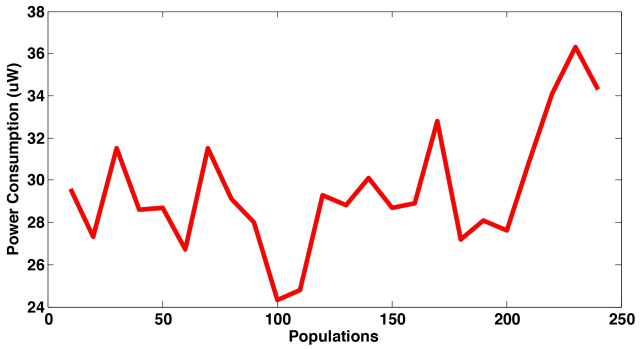

Fig. 9. Values for power consumption after 100 iterations according to population size.

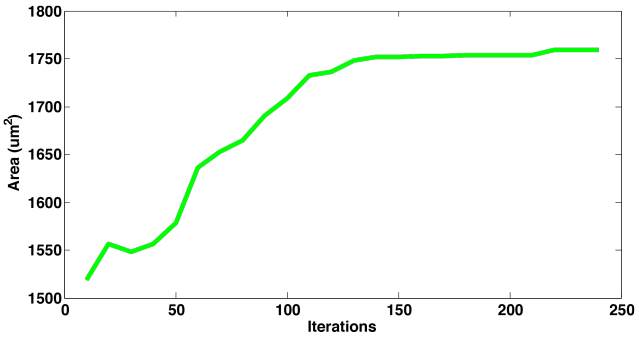

Fig. 11. Values of gate area for a population of 100 individuals and varying the number of iterations.

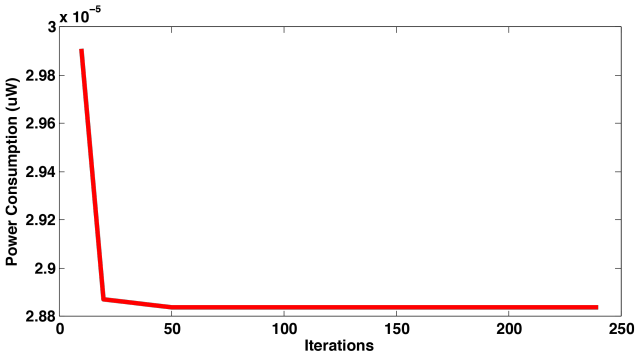

Fig. 12. Values of power consumption for a population of 100 individuals and varying the number of iterations.

minimization. Another important factor is that running more iterations demands more computational time, thus resulting in a significant increase in the cost of the algorithm.

To verify if PSO is dependent on the seed of the random number generator, four new simulation runs were carried out, each with a fixed size of 100 individuals and iterations ranging from 10 to 240. Fig. 13 demonstrates the results.

This figure shows that the PSO algorithm has different behavior for different values of seed. The final results for the cost function is different for every case of tested seeds, but it is possible to analyze which had the best result when the seed value was defined as default (2^0). Thus, the value set in the seed also influence the final result of the cost function.

It is possible to compare the results obtained by PSO with the same optimization procedure performed by SA, which was described in [3]. The values of the required specifications are shown in the second column of Table I. The results obtained by UCAF tool using SA and PSO are shown in the third and fourth columns of this table, respectively.

The results obtained by using PSO presented a reduction of 41.93% in dissipated power and 13.43% in gate area when compared to the results obtained by using SA. The SA used 8,920 iterations, and, for purposes of comparison, PSO was configured to have 9,000 executions. The final cost function was reduced from 0.86 in the SA to 0.23 in the PSO algorithm. It is possible to notice that both algorithms reached all the required specifications.

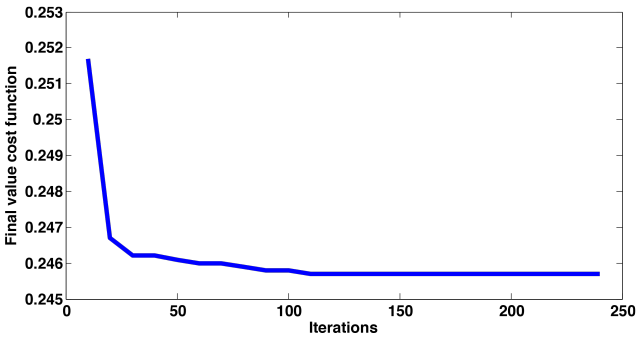

Fig. 10. Values of cost function for a population of 100 individuals and varying the number of iterations.

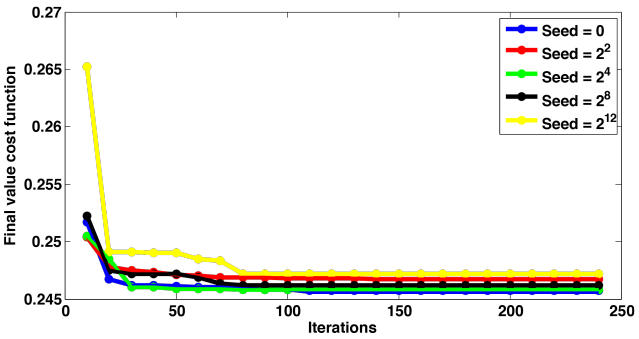

Fig. 13. Cost function value for a population of 100 individuals and varying the number of iterations for different seeds of the random number generator.

TABLE I
SPECIFICATION RESULTS FOR THE DESIGN OF A TELESCOPIC USING PSO AND SA.

Specification	Required Value	PSO (this work)	SA [3]
Av0 (dB)	$\geq$ 60.00	63.90	62.88
GBW (MHz)	$\geq$ 2.00	4.71	3.50
PM (^{0})	$\geq$ 50.00	75.98	50.09
SR (V/μs)	$\geq$ 5.00	5.01	5.76
ICMR+ (V)	$\geq$ 0.40	0.58	0.64
ICMR- (V)	$\leq$ -0.40	-0.71	-0.60
Pdiss (μW)	Minimize	28.84	49.66
Gate area (μm^2)	Minimize	1690	1957
Cost function	Minimize	0.23	0.86

The values for the variables of the circuit are contained in Table II. The second column shows the results of the values for the variables of the circuit where the second column shows the result obtained with the PSO, and the third with the SA.

TABLE II
VALUES FOR THE CIRCUIT VARIABLES FOUND FOR THE TELESCOPIC AMPLIFIER USING PSO AND SA.

Variable	PSO (this work)	SA [3]
W_1	27.75 μm	32.88 μm
W_3	26.68 μm	44.47 μm
W_5	16.78 μm	50.09 μm
W_7	46.99 μm	47.22 μm
L_1	10.00 μm	9.86 μm
L_3	0.50 μm	1.75 μm
L_5	8.58 μm	9.66 μm
L_7	8.72 μm	6.01 μm
I_{Bias}	16.02 μA	29.65 μA
V_n	0.85 V	0.86 V
V_p	-0.85 V	-0.86 V

V. CONCLUSION

The proposed methodology using PSO for the automatic design of analog integrated circuits presented better results when compared to the same procedure using SA meta-heuristic. Both algorithms achieved all the required specifications. However, the circuit sized with PSO has a considerable improvement in terms of dissipated power and gate area, with approximately the same number of iterations. The proposed algorithm is suitable for analog design, due to the fact that the convergence of the final solution is not directly dependent on the initial solution. Also, it requires small interference from the human designer. As future work, we aim to implement a methodology for estimating the process, voltage and temperature (PVT) variations within UCAF, including yield prediction during the optimization procedure.

REFERENCES

[1] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec 2000.

[2] R. Harjani, "Oasys: a framework for analog circuit synthesis," in *ASIC Seminar and Exhibit, 1989. Proceedings., Second Annual IEEE*, Sep 1989, pp. P13–1/1–4.

[3] L. C. Severo, A. B. de Oliveira, A. Girardi, F. N. Kepler, and M. C. Cera, "Simulated annealing to improve analog integrated circuit design: Trade-offs and implementation issues," in *Simulated Annealing - Single and Multiple Objective Problems*, 2012, ch. 13, pp. 261–284.

[4] M. Taherzadeh-Sani, R. Lotfi, H. Zare-Hoseini, and O. Shoaei, "Design optimization of analog integrated circuits using simulation-based genetic algorithm," in *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on*, vol. 1, 2003, pp. 73–76 vol.1.

[5] S. Deyati and P. Mandal, "An automated design methodology for yield aware analog circuit synthesis in submicron technology," in *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, March 2011, pp. 1–7.

[6] A. Jafari, M. Zekri, S. Sadri, and A. Mallahzade, "Design of analog integrated circuits by using genetic algorithm," in *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, vol. 1, March 2010, pp. 578–581.

[7] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *INTEGRATION, the VLSI journal*, vol. 43, no. 1, pp. 136–155, 2010.

[8] B. Liu, F. V. Fernandez, and G. G. E. Gielen, "Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 793–805, June 2011.

[9] R. A. L. Moreto, C. E. Thomaz, S. P. Gimenez, and A. L. P. Rotondaro, "From architecture to manufacturing: An accurate framework for optimal ota design," in *2015 Latin America Congress on Computational Intelligence (LA-CCI)*, Oct 2015, pp. 1–6.

[10] F. R. Shahroury and I. Riad, "The design and optimization of low-voltage pseudo differential pair operational transconductance amplifier in 130 nm cmos technology," *18th International Conference on Computer Modelling and Simulation*, 2016.

[11] E. Martens and G. Gielen, "Classification of analog synthesis tools based on their architecture selection mechanisms," *Integration, the {VLSI} Journal*, vol. 41, no. 2, pp. 238 – 252, 2008.

[12] M. Clerc, "Beyond standard particle swarm optimisation," *Int. J. Swarm. Intell. Res.*, vol. 1, no. 4, pp. 46–61, Oct. 2010.

[13] R. Vural, T. Yildirim, T. Kadioglu, and A. Basargan, "Performance evaluation of evolutionary algorithms for optimal filter design," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 1, pp. 135–147, Feb 2012.

[14] L. Min, L. Bing, X. Weiming, and W. Houjun, "Diagnostics of incipient faults in analog circuits," in *Electronic Measurement Instruments (ICEMI), 2013 IEEE 11th International Conference on*, vol. 2, Aug 2013, pp. 833–838.

[15] S. Kamisetty, J. Garg, J. Tripathi, and J. Mukherjee, "Optimization of analog rf circuit parameters using randomness in particle swarm optimization," in *Information and Communication Technologies (WICT), 2011 World Congress on*, Dec 2011, pp. 274–278.

[16] R. Domanski and A. Girardi, "Analysis of optimization algorithms for sizing analog circuits," *6th IEEE Latin American Symposium on Circuits and Systems*, 2015.

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[18] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, Oct 1995, pp. 39–43.

[19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, May 1998, pp. 69–73.

[20] ——, "Empirical study of particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3, 1999, pp. –1950 Vol. 3.

[21] M. G. Resende, "Computing approximate solutions of the maximum covering problem with grasp," *J. of Heuristics*, vol. 4, pp. 161–171, 1998.

[22] P. Allen and D. Holberg, *CMOS Analog Circuit Design*, ser. Oxford series in electrical and computer engineering. Oxford University Press, 2002.