Abbas
El Dor[1,*],
Mourad
Fakhfakh[2],
Patrick Siarry[3]

J. Electrical Systems 12-3 (2016): 612-622

JES Journal of Electrical Systems

Regular paper

Multiobjective Differential Evolution Algorithm using Crowding Distance for the Optimal Design of Analog Circuits

This paper details the Multiobjective Differential Evolution algorithm (MODE) using crowding distance for the sizing of analog circuits. MODE is used to compute the Pareto front of a bi-objective optimization problem, namely maximizing the high current cut-off frequency and minimizing the parasitic input resistance of a second generation current conveyor. To highlight performances of MODE, comparisons with the non-sorting genetic algorithm (NSGA-II) were performed. These comparisons show that MODE outperforms NSGA-II in terms of quality of the optimal solutions, diversity of those solutions along the Pareto front, and computing time.

Keywords: Metaheuristics; Multiobjective optimization; MODE; NSGA-II; CMOS; Second generation current conveyor.

Article history: Received 18 April 2016, Accepted 18 August 2016

1. Introduction

The issue of analog electronic circuits' design, sizing and performance optimization continues to attract substantial attention of analog designers. This is mainly due to the ceaseless demand for high performance electronic circuits: lower power consumption, smaller occupied area, larger number of implemented options, etc. [1]. In general, analog circuits' sizing is a time-consuming, tedious and iterative manual process which relies on designer intuition and experience. In this regard, and because of their design complexity, automating analog components' sizing procedure is an important issue towards being able to rapidly design true high performance circuits [2]. Nowadays, metaheuristics are widely used for these purposes, such as Simulated Annealing (SA), Particle Swarm Optimization (PSO), Genetic Algorithms (GA), see for instance [3].

In this paper, we propose to use the Multiobjective Differential Evolution algorithm (MODE) for the optimal design of analog circuits, more precisely the CMOS second generation current conveyor considered in [4], and we present a comparison with the non-sorting genetic algorithm, namely NSGA-II [5]. The same objective functions are considered (minimizing the parasitic input R_X resistance and maximizing the current high cut-off frequency f_{chi}), the same set of constraints to be satisfied, the same set of parameters (and their respective variation range) to be handled, the same technology, etc, as in [3].

The remainder of the paper is structured as follows. Section 2 briefly presents the multi-objective optimization problem. Section 3 details the MODE technique. Section 4 deals

[1,*] Corresponding author: Ecole des Mines de Nantes, TASC INRIA (CNRS UMR 6241), 4 rue Alfred Kastler, 44300 Nantes, France. E-mail: abbas.eldor@mines-nantes.fr
[2] ENET' Com, University of Sfax, B.P. 1163, Sfax 3018, Tunisia. E-mail: fakhfakhmourad@gmail.com
[3] Université de Paris-Est Créteil, LiSSi (E.A. 3956), 122 rue Paul Armangot, 94400 Vitry-sur-Seine, France. E-mail: siarry@u-pec.fr

with current conveyors and gives the mathematical model of the constrained problem to be optimized. Comparison results between MODE and NSGA-II are reported in Section 5. Finally, Section 6 gives some concluding remarks.

2. Multi-objective optimization problem formulation

A multi-objective optimization problem (MOP) can be defined as a problem requiring the optimization of more than one objective function simultaneously. Optimal solution (or optimal decisions) of the problem need to be computed with respect to trade-offs between two or more conflicting objectives. The MOP can be formally defined as the problem of finding the decision vectors $X(x_1, x_2, \dots, x_n)$ in the decision space D, which satisfy the m inequality constraints and p equality constraints, and optimize the objective function vector $F((f_1(X), f_2(X), \dots, f_k(X)))$. In the following, the definition is given with respect to a minimization problem, and it can be used for a maximization problem as every maximization problem can be converted into minimization problem by multiplying it by (-1), and vice-versa. MOP can be defined in the following format:

$$\mathit{Minimize}\quad \left(f_1(\mathrm{X}), f_2(\mathrm{X}), \dots, f_k(\mathrm{X})\right)\quad \text{such that:} g_i\left(X\right) \le 0 \text{ and } h_l\left(X\right) = 0 \tag{1}$$

where k is the number of functions, $i = 1, \dots, m$, $l = 1, \dots, p$, g_i and h_l represent the inequality and the equality constraints respectively.

As mentioned above, the objective functions may be in conflict. Thus, in contrast with a mono-objective optimization problem the goal of the multi-objective optimization algorithm is to provide a set of Pareto optimal solutions. A solution X of a problem is called Pareto optimal, if and only if there does not exist another solution Y such that $f(Y)$ dominates $f(X)$, i.e., no component of $f(X)$ is smaller than the corresponding component of $f(Y)$ and at least one component is greater [6].

3. Overview of the Differential Evolution metaheuristic

Differential Evolution (DE) is inspired by Genetic Algorithms (GA) and Evolutionary Strategies (ES) [7]. GA changes the structure of individuals using the mutation and crossover operators, while ES achieves self-adaptation by a geometric manipulation of individuals. This combination has been implemented through an operation, simple but powerful, of mutation vectors proposed in 1995 by K. Price and R. Storn [8]. DE is a direct parallel method that uses NP solutions: $X_{i,G}$, $i = 1, \dots, NP$, where the index i denotes the i^{th} solution of the population and G denotes the generation to which the population belongs.

The standard DE uses three main operators (mutation, crossover and selection) for the movement of the agents as well as GA. Firstly, a vector $X_{i,G}$ is randomly selected, which is the current vector of the agent i at generation G. Then, $X_{i,G}$ moves according to the three following operations:

a. **Mutation**: the mutation operation of DE applies the vector difference between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three solutions $\{X_{r1,G}, X_{r2,G}, X_{r3,G}\}$ in the population set of (say) NP elements. For each current vector (target vector) $X_{i,G}$, a mutant vector (perturbed vector) $V_{i,G}$ is generated based on the three chosen solutions, which will be calculated using the following equation:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} + X_{r3,G}) \tag{2}$$

where, $i = 1 \ldots NP$, $r_1, r_2, r_3 \in \{1 \ldots NP\}$ are randomly selected integers such that $r_1 \neq r_2 \neq r_3 \neq i$ and F is a real and constant factor $\in [0, 1]$.

b. **Crossover**: the crossover operation is introduced to increase the diversity of the target vectors. Once the mutant vector $V_{i,G} = (x_{1,i,G}, \ldots, x_{n,i,G})$ is generated, it is subjected to crossover operation with target vector $X_{i,G} = (x_{1,i,G}, \ldots, x_{n,i,G})$, that finally generates the trial solution, $U_{i,G} = (u_{1,i,G}, \ldots, u_{n,i,G})$, as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand \leq C_r \quad \forall j = jj \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{3}$$

where, $j = 1, \ldots, n$, $jj \in \{1, \ldots, n\}$ is a random parameter's index, chosen once for each i. The crossover rate, $C_r \in [0, 1]$, is set by the user.

c. **Selection**: The population for the next generation is selected from the solution in current population (target vector $X_{i,G}$) and its corresponding trial solution (trial vector $U_{i,G}$). To determine which vector should become a member of generation $G+1$, the fitness function values of these two vectors are compared. Indeed, we keep the vector that has the smallest fitness function value, in the case of minimization, according to the following rule:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{4}$$

As a result, all the solutions of the next generation are as good as or better than their counterparts in the current generation. In DE, trial solution is not compared against all the solutions in the current generation, but only against one solution, its counterpart, in the current generation.

DE algorithm is proposed by the authors for solving mono-objective optimization problems. While Multi-Objective Differential Evolution (MODE) is an extension of DE introduced to solve the multi-objective optimization problems. The role of MODE is to maintain the diversity of the population and to generate potential candidate solutions right from the beginning of the algorithm [9]. This algorithm uses Pareto-based ranking assignment and crowding distance metric [10]. In MODE the fitness of a solution is first calculated using Pareto-based ranking and then reduced with respect to the solutions crowding distance value. This fitness value is then used to select the best solutions for the next generation.

The non-dominated sorting algorithm divides the R population, a parent population P_i of size NP and an offspring population Q_i of size NP, into several fronts, denoted by $F_1, F_2, \ldots, F_j$. The first front (F_1) is formed by non- dominated solutions from R. Solutions in F_1 are removed from R and the remaining solutions are employed to calculate the next set of non-dominated solutions, denoted by F_2. This process is repeated in order to find F_3, and so on, until R is empty. The rank value of an individual is the index of the front it belongs to.

The solutions from the fronts are copied to the next population P_{i+1}. As P_i and Q_i have size NP, there are $2NP$ solutions which compete for NP slots in P_{i+1}. Solutions from fronts

$F_j=1,\ldots,n$ are copied to P_{i+1} until there are more solutions in front F_n than slots in P_{i+1}. In this case, the individuals from F_n with the highest crowding distance values are copied to P_{i+1} until P_{i+1} is fulfilled. The crowding distance is useful to maintain the population diversity. It reflects the density of solutions around its neighborhood. This value is calculated from a perimeter defined by the nearest neighbors in each objective. Figure 1 illustrates the non-dominated sorting algorithm and crowding distance mechanism implemented in MODE.

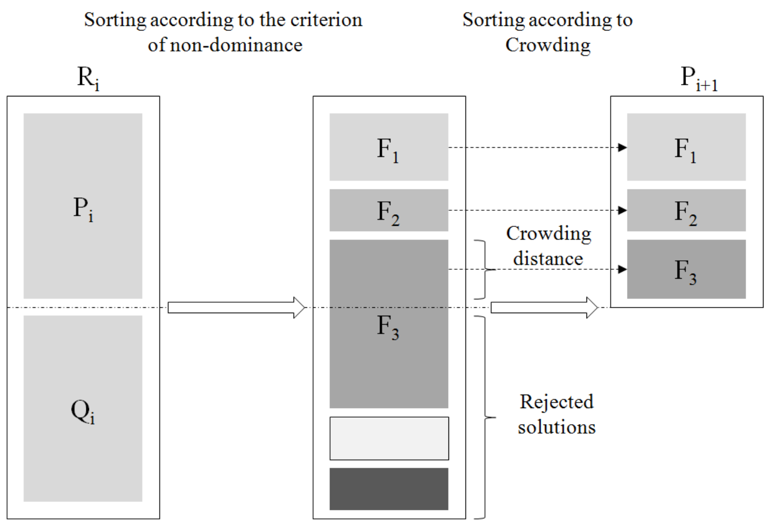

Figure 1: Sorting by non-dominance and crowding distance used in MODE.

MODE uses a tournament selection to choose individuals for reproduction. It randomly picks two individuals from P_i and chooses the best one, which has the lowest rank. If both solutions have the same rank, the solution with the longest crowding distance is preferred.

Pseudo code and flowchart of MODE are given in Algorithm 1 and Figure 2.

Algorithm 1: Pseudo code of MODE

- **Initialize** the values of *n, k, NP, Cr, F, max_fun* (maximum number of function evaluations) and *feval* = 0 (number of function evaluations).
- **Input** lower and upper bounds on decision variables [$xmin_i$, $xmax_i$], where, $i = 1,\ldots,n$.
- **Generate** *NP* random solutions.
- **Generate** *NP* opposite solutions.
- **Evaluate** function values of 2*NP* solutions.
- **Select** *NP* fittest solutions using non dominated and crowding distance and store them in current population pop_1.
- **While** (*feval* < *max_fun*) // main loop

  {

  for ($i: 0 \rightarrow NP$) // iteration loop

  {

  - **Select** randomly three distinct individuals X_{r1}, X_{r2} and X_{r3} and also different from target individual X_i.
  - **Generate** a perturbed individual V_i using mutation equation (eq. 2).
  - **Generate** a trial individual U_i using crossover between V_i and X_i by (eq. 3).
  - **Evaluate** function value at this U_i.

    Evaluate_function(); feval++;
  - Nondomination checking of trial individual U_i with target individual X_i.

    If (U_i dominates X_i)

    Replace X_i by U_i in current population pop_1.

    and add X_i to advanced population pop_2.

Else

Add U_i to advanced population pop_2.

} // end of iteration loop.

Select *NP* fittest solutions using non dominated and crowding distance sorting and store them in pop_1.

pop_1 = nondominated_crowd_sort (pop_1, pop_2).

}// end main loop.

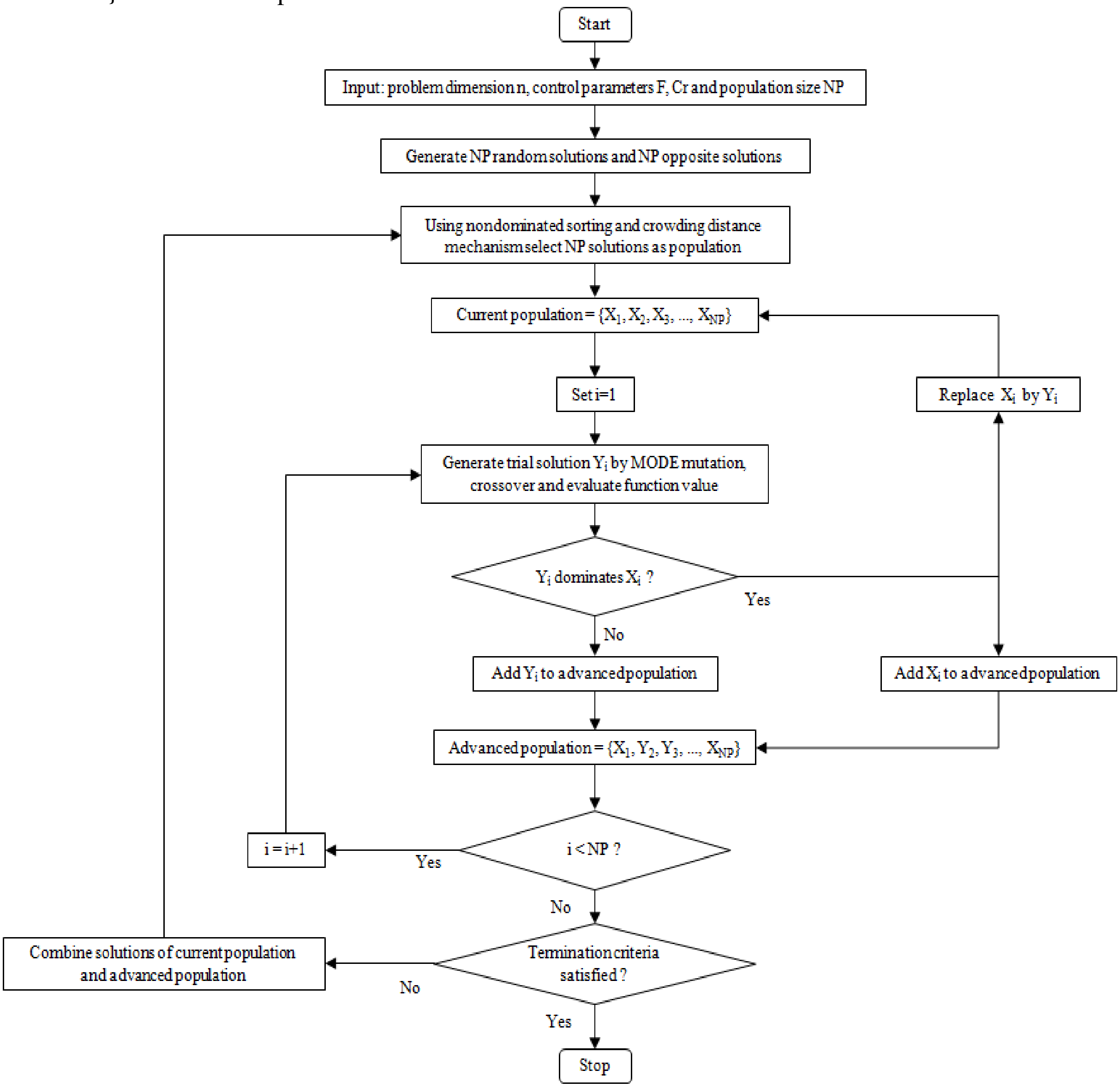

Figure 2: Flowchart of MODE.

4. Second generation current conveyors

It is well known that current mode circuits overcome most of the voltage mode circuit drawbacks [11, 12]. That's why current mode circuits are receiving much more attention by the analog designers. Among the current mode circuits, current conveyors (CCs) are without a doubt the most famous ones [13, 14].

This is due to the fact that CCs are versatile circuits that form a basic building block that can be used to design a large variety of circuits, such as active filters, simulated inductors, active oscillators, to name a few [1, 15]. In this paper we consider a translinear loop-based positive second generation CMOS current conveyor [16, 17]. Its schematic circuit is shown in Figure 3. Where transistors M_5, M_6, M_9 and M_{10} form the translinear loop. The rest of the transistors mainly form current mirrors, particularly transistors M_8, M_{11}-M_{13} that allow conveying the current from port *X* to port *Z*. I_0 is the CC's bias current. V_{DD} and V_{SS} are the voltage power supplies.

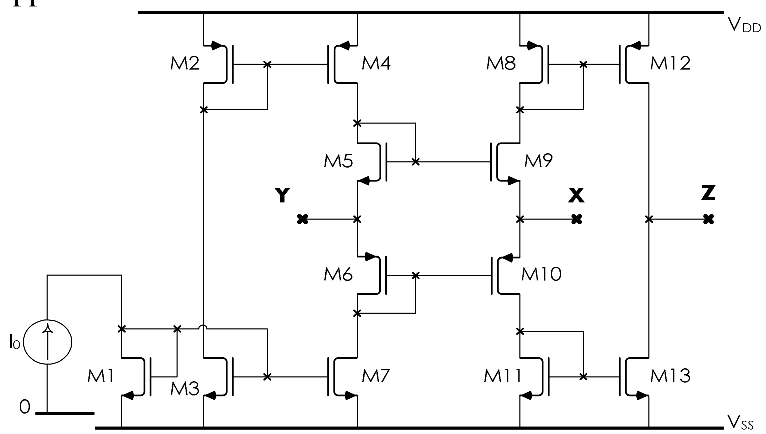

Figure 3: A CMOS translinear loop-based positive second generation current conveyor

Due to its inherent parasitic elements, the behaviour of the current conveyors can differ from its ideal one. It has been shown that among these elements the *X*-port input resistance (R_X) is the most dominant one [16, 17]. Matrix equation (5) gives the relationships between the CC ports taking into account the effect of R_X.

$$\begin{pmatrix} I_Y \\ V_X \\ I_Z \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & R_X & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} V_Y \\ I_X \\ V_Z \end{pmatrix} \tag{5}$$

As aforementioned, for comparison reasons we deal with minimizing the parasitic input resistance (R_X) and maximizing the current high cut-off frequency (f_{chi}). Expression (6) gives the symbolic expression of R_X.

$$R_X = \frac{1}{\sqrt{2K_N(\frac{W}{L})_N(1+\lambda_N V_{DS})I_0} + \sqrt{2K_P(\frac{W}{L})_P(1+\lambda_P V_{DS})I_0}} \tag{6}$$

where, λ_N, λ_P, K_N and K_P are technology parameters. I_0 is the bias current and V_{DS} is the drain to source voltage of the MOS transistor. Subscripts *N* and *P* refer to NMOS and PMOS transistors, respectively. *W* and *L* are width and channel lengths of the MOS transistor, respectively.

Expression of f_{chi} was obtained using CASCADES software [18]; it is not given due to its large number of terms.

In addition to the constraints imposed by the technology, the circuit is subject to a set of intrinsic constraints that are due to the saturation working mode of the MOS transistors, mismatch, symmetry, etc. Main such constraints are given by (7)-(10).

$$K_N\left(\frac{W}{L}\right)_N = K_P\left(\frac{W}{L}\right)_P \tag{7}$$

$$W_N L_N = W_P L_P \tag{8}$$

$$\frac{1}{2}V_{DD} - V_{TN} - \sqrt{\frac{I_0}{K_N\left(\frac{W}{L}\right)_N}} > \sqrt{\frac{I_0}{K_P\left(\frac{W}{L}\right)_P}} \tag{9}$$

$$\frac{1}{2}V_{DD} - V_{TP} - \sqrt{\frac{I_0}{K_P\left(\frac{W}{L}\right)_P}} > \sqrt{\frac{I_0}{K_N\left(\frac{W}{L}\right)_N}} \tag{10}$$

$V_{T(N,P)}$ is the threshold voltage of the (N, P) MOS transistor.

5. Parameter settings and comparison results

The parameter values for MODE used for this problem are fixed as follows: NP (the initial population size) is set to 100, the scaling factor F and the crossover probability C_r are taken as 0.5 and 0.3, respectively. Maximum number of function evaluations *max_fun* is set to 100000. The optimal values for the physical dimensions of MOS transistors (W_N, L_N, W_P and L_P for each NMOS and PMOS transistor) are obtained by minimizing R_X and maximizing f_{chi}, with different values of I_0 ∈{50; 100; 150; 200; 250; 300}µA. The simulations are performed using the technology CMOS AMS 0.35µm technology with a supply voltage of 2.5V.

Table 1 presents the obtained results by applying MODE with different values of I_0. In this table the given parameters' values refer to the lower and higher edge of the Pareto front in each case, and the corresponding fitness value (functions' values of f_{chi} and R_X) is also provided. Figure 4 shows the obtained non-dominated solutions for different bias current values I_0.

Besides, and for comparison reasons, NSGA-II [5] algorithm was applied to generate the Pareto front (R_X, f_{chi}) for the same values of I_0. Table 2 presents the obtained results by applying NSGA-II with different values of I_0. The number of non-dominated solutions in each Pareto front obtained by NSGA-II is smaller than the obtained number of MODE which signifies that the latter has better Pareto front. Moreover, the solutions given by NSGA-II lie in a very small range compared to those given by MODE and this shows the better diversity of the latter. Figure 5 illustrates the case of I_0=150µA.

Table 1: Optimal parameters values obtained by MODE

		L_N (μm)	W_N (μm)	L_P (μm)	W_P (μm)	f_{chi} (GHz)	R_X (ohm)
I_0=50μA	Lower edge of the Pareto front	0.57	19.69	0.35	29.93	0.771	545
	Higher edge of the Pareto front	0.52	2.91	0.35	4.81	2.128	1391
I_0=100μA	Lower edge of the Pareto front	0.56	19.61	0.35	29.99	1.101	386
	Higher edge of the Pareto front	0.52	5.81	0.35	9.59	2.132	696
I_0=150μA	Lower edge of the Pareto front	0.57	19.89	0.35	29.99	1.325	314
	Higher edge of the Pareto front	0.52	8.71	0.35	14.39	2.131	464
I_0=200μA	Lower edge of the Pareto front	0.56	19.32	0.35	29.92	1.582	274
	Higher edge of the Pareto front	0.52	11.62	0.35	19.19	2.131	348
I_0=250μA	Lower edge of the Pareto front	0.57	20.00	0.35	30.00	1.701	242
	Higher edge of the Pareto front	0.52	14.52	0.35	23.98	2.131	278
I_0=300μA	Lower edge of the Pareto front	0.57	19.94	0.35	30.00	1.871	221
	Higher edge of the Pareto front	0.52	18.02	0.35	29.81	2.093	228

L_N, W_N, L_P and W_P refer to length and width of NMOS and PMOS transistors, respectively.

Table 2: Optimal parameters values obtained by NSGA-II

		L_N (μm)	W_N (μm)	L_P (μm)	W_P (μm)	f_{chi} (GHz)	R_X (ohm)
I_0=50μA	Lower edge of the Pareto front	0.57	20.05	0.35	30.00	0.759	542
	Higher edge of the Pareto front	0.52	2.90	0.35	4.80	2.131	1392
I_0=100μA	Lower edge of the Pareto front	0.58	20.08	0.35	30.00	1.072	383
	Higher edge of the Pareto front	0.52	5.81	0.35	9.59	2.132	696
I_0=150μA	Lower edge of the Pareto front	0.57	20.08	0.35	30.00	1.313	313
	Higher edge of the Pareto front	0.52	18.15	0.35	30.00	1.476	322
I_0=200μA	Lower edge of the Pareto front	0.57	20.07	0.35	30.00	1.516	217
	Higher edge of the Pareto front	0.52	11.65	0.35	19.22	2.126	348
I_0=250μA	Lower edge of the Pareto front	0.58	20.08	0.35	30.00	1.694	242
	Higher edge of the Pareto front	0.52	14.52	0.35	23.98	2.131	278
I_0=300μA	Lower edge of the Pareto front	0.58	20.08	0.35	30.00	1.855	221
	Higher edge of the Pareto front	0.52	17.42	0.35	28.77	2.132	232

L_N, W_N, L_P and W_P refer to length and width of NMOS and PMOS transistors, respectively.

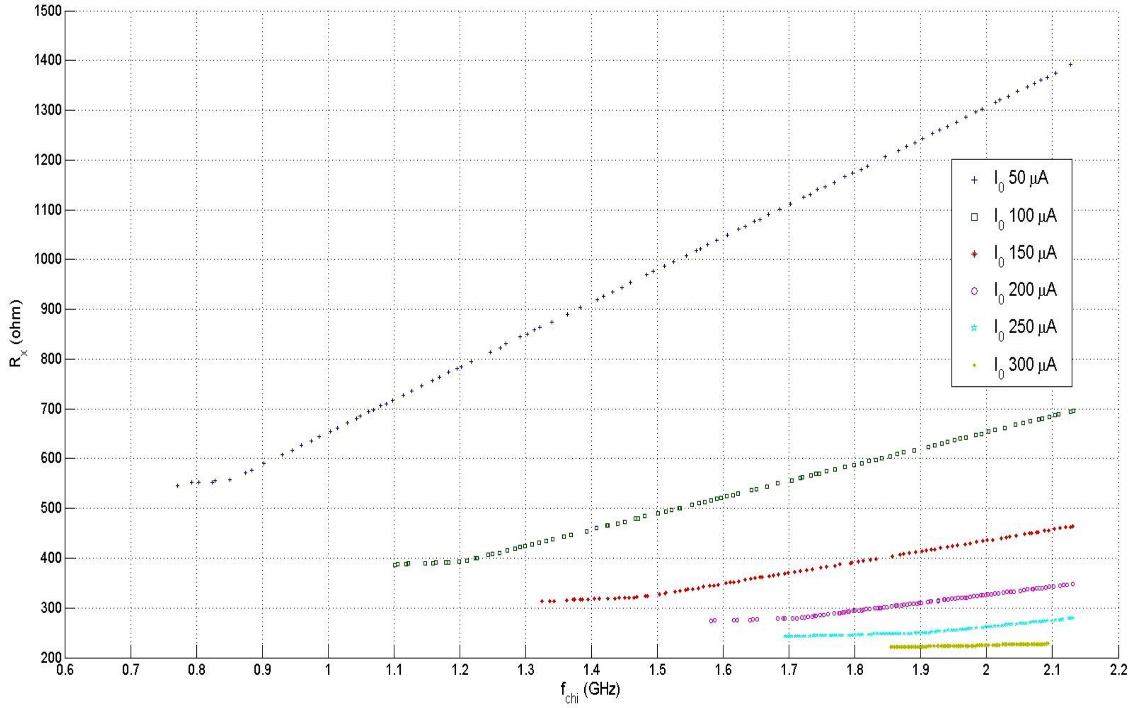

Figure 4: Pareto fronts obtained for various values of the bias current I_0

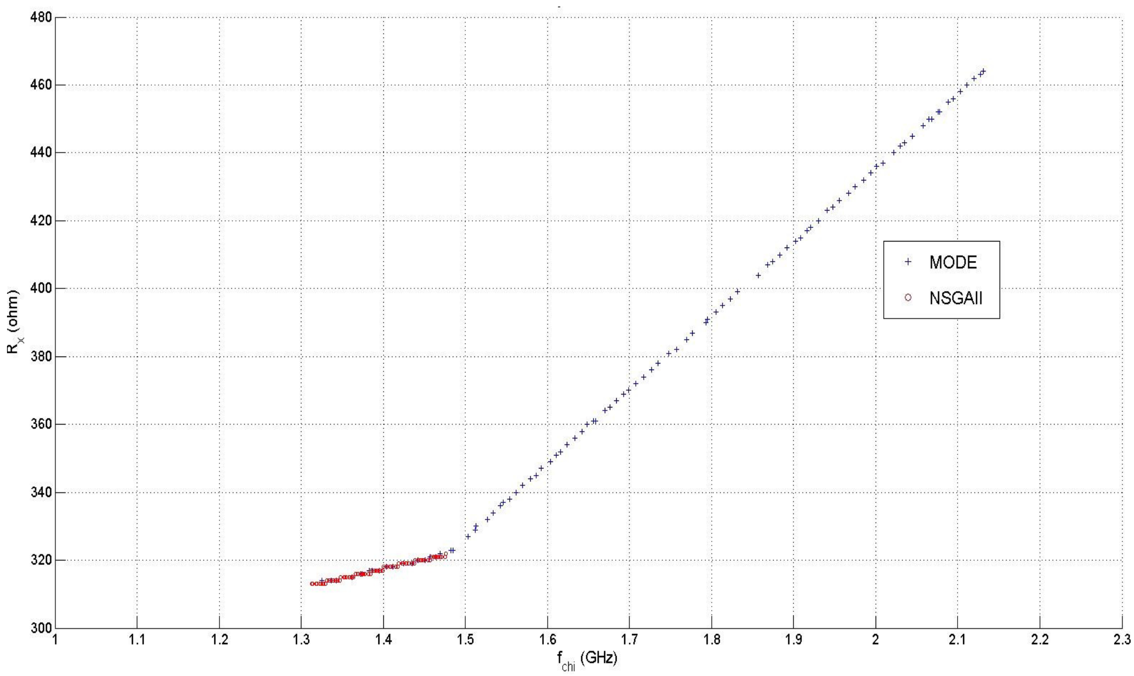

Figure 5: Pareto front obtained by MODE and NSGA-II for I_0=150μA

Another advantage of MODE should be taken into consideration which is its significant low CPU consumption compared to NSGA-II. Indeed, different executions were carried on using MODE and NSGA-II for a series of bias currents to record their execution times. The results are presented in Table 2. One can notice that the execution time of MODE is very competitive compared to NSGA-II.

Table 2: Execution times of MODE and NSGA-II (in seconds) with different values of I_0

I_0	50µA	100µA	150µA	200µA	250µA	300µA
MODE	4.409	4.071	3.339	3.454	2.108	4.239
NSGA-II	785.618	854.600	995.832	830.319	881.310	1039.486

6. Conclusion

In this paper, we have successfully applied the multi-objective differential evolution algorithm (MODE) to optimize the sizing of analog circuits, more precisely CMOS current conveyors (CCIIs).

A bi-objective optimization problem was considered in order to minimize the *X*-port parasitic resistance R_X while simultaneously maximizing the current high end cut-off frequency f_{chi}. A series of simulation experiments were conducted for different bias current values, by applying MODE, using crowding distance metric, and were compared to NSGA-II algorithm performances. The results show that MODE can produce a better diversity of solutions in Pareto fronts and offers a significant reduction of computing time, when compared to NSGA-II, which makes it very suitable to be implemented within a CAD algorithm.

References

[1] M. Fakhfakh, E. Tlelo-Cuautle and R. Castro-Lopez (Eds.), Analog/RF and Mixed-Signal Circuit Systematic Design, Springer, 2013.

[2] A. R. Conn, P. K. Coulman, R. A. Haring, G. L. Morrill and C. Visweswariah, "Optimization of custom MOS circuits by transistor sizing," in *The international conference on computer aided design, (ICCAD'96),* San Jose, CA, USA, November 10-14, 1996.

[3] M. Fakhfakh, E. Tlelo-Cuautle and P. Siarry (Eds.), Computational Intelligence in Electronic Design: AMS/RF Circuit Design, Springer, 2015.

[4] A. El Dor, M. Fakhfakh and P. Siarry, "Performance optimization of CMOS second generation current conveyors using a multi-swarm algorithm," *AEU - International Journal of Electronics and Communications,* vol. 68, no. 6, pp. 496-503, 2014.

[5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation* , vol. 6, no. 2, pp. 182-197, 2002.

[6] C. R. Raquel and P. C. Naval, "An effective use of distance in multiobjective particle swarm optimization," in *The Genetic and Evolutionary Computation Conference (GECCO'2005),* Washington, DC, USA, pp. 257-364, June 25-29, 2005.

[7] A. El Dor, M. Clerc and P. Siarry, "Hybridization of differential evolution and particle swarm optimization in a new algorithm: DEPSO-2S," in *Proceedings of the 2012 International Conference on Swarm and Evolutionary Computation, ICAISC (SIDE-EC)*, Zakopane, Poland, pp. 57-65, April 29-May 3, 2012.

[8] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization,* vol. 11, no. 4, pp. 341–359, 1997.

[9] F. Xue, A. Sanderson and R. Graves, "Pareto-based multi-objective differential evolution," in *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, Canberra, Australia, pp. 862-869, 2003.

[10] M. Ali, P. Siarry and M. Pant, "An efficient differential evolution based algorithm for solving multi-objective optimization problems," *European Journal of Operational Research,* vol. 217, no. 2, pp. 404-416, 2012.

[11] C. Toumazou, F. J. Lidgey and D. G. Haigh, Analog IC design: the current mode approach, IET, 1993.

[12] B. Wilson, F. J. Lidgey and C. Toumazou, "Current mode signal processing circuits," *IEEE International Symposium on Circuits and Systems,* 1988.

[13] A. Sedra and K. Smith, "A second generation current conveyor and its application," *IEEE Transactions on Circuit Theory,* vol. CT-17, pp. 132-134, 1970.

[14] I. A. Awad and A. M. Soliman, "Inverting second generation current conveyors: the missing building blocks, CMOS realizations and applications," *International Journal of Electronics,* vol. 86, no. 4, pp. 413-432, 1999.

[15] E. Tlelo-Cuautle, Integrated circuits for analog signal processing, Springer, 2012.

[16] A. Chatterjee, M. Fakhfakh and P. Siarry, "Design of second generation current conveyors employing bacterial foraging optimization," *Microelectronics Journal,* vol. 41, p. 616–626, 2010.

[17] S. BenSalem, M. Fakhfakh, D. S. Masmoudi, M. Loulou and N. Masmoudi, "A high performances CMOS CCII and high frequency applications," *Journal of Analog Integrated Circuits and Signal Processing,* vol. 49, no. 1, p. 71–78, 2006.

[18] M. Fakhfakh and M. Loulou, "Live Demonstration: CASCADES.1: a flow-graph-based symbolic analyzer", in *The IEEE International Symposium on Circuits and Systems (ISCAS),* Paris, France, May 30-June 2, 2010.