Analogue Circuit Optimization through a Hybrid Approach

Mourad Fakhfakh, Amin Sallem, Mariam Boughariou, Sameh Bennour,
Eya Bradai, Emna Gaddour, and Mourad Loulou

Abstract. Optimal analogue circuit sizing is investigated in this chapter. It is shown that hybridization of a global optimization approach with a local one leads to better results in optimization of such circuits, than using classical approaches. The case of merging Genetic Algorithms with the Simulated Annealing technique is considered. The hybrid algorithm is detailed and is evaluated using test functions. It is shown through three application examples, i.e. optimization of performances of a current conveyor, an operational transconductance amplifier and a low noise amplifier, that such hybrid algorithms yield optimal solutions in a much shorter time, when compared to conventional meta-heuristics.

1 Introduction

Advances in very low scale integration (VLSI) technology nowadays allow the realization of complex integrated electronic circuits and systems [1]. Design automation of digital circuits has already been widely explored and many CAD tools are available [1-3]. However, the automated design of analogue circuits is far to be mature. It still lags behind that of digital circuits. Optimal synthesis, design and sizing of analogue components/circuits are very often a bottleneck in the design flow [1-5].

Many analogue circuit optimization techniques are proposed in the literature with the trend to the use of statistic-based approaches, see for instance [3,6-9]. These approaches generally start with finding a "good" DC operating point, and then a simulation-based tuning procedure takes place. However, these approaches are time consuming and do not guarantee the convergence to the 'global' optimum solution. Actually, analogue circuit optimization problems simultaneously deal with different types of variables, objective functions and constraints. Therefore, classical optimization techniques as well as statistic-based approaches are

Mourad Fakhfakh · Amin Sallem · Mariam Boughariou · Sameh Bennour
Eya Bradai · Emna Gaddour · Mourad Loulou
University of Sfax, Tunisia
email: mourad.fakhfakh@ieee.org

M. Köppen et al. (Eds.): Intelligent Comput. Optimization in Eng., SCI 366, pp. 297 – 327.
springerlink.com

generally not adequate. Meta-heuristics are used to solve such hard constrained problems [10,11]. They offer the advantages to be ‘easily’ modified and adapted to suit specific problem requirements [10-13]. Even though they don’t guarantee to find exactly the optimal solution, a fact due to their stochastic nature, they give, within an acceptable computing time, a good approximation of it. Meta-heuristics can be classified into two categories [13]:

- Population based approaches. This category mainly comprises:
    - Evolutionary Algorithms (EA) [14,15]: Genetic Algorithms (GA) [16-18], the Evolutionary Programming (EP) [14,15,19], etc.,
      and
    - Swarm Intelligence Techniques (SI) [20]: Particle Swarm Optimization (PSO) [21,22], Ant Colony Optimization (ACO) [23], Bacterial Foraging Optimization (FBO) [24], etc.

  and
- Single solution based approaches, such as Simulated Annealing (SA) [25-28], Tabu Search (TS) [29,30]…

Some among the aforementioned optimization techniques were used in optimizing analogue circuits, see for instance [16-18,27,28,31-36]. These approaches are part of the so called global optimization (i.e. a global search approaches (GS)). On the other hand, SA is sometimes referred in the literature as a local search (LS) approach, since it is basically based on a LS one, namely the Metropolis algorithm [37].

Both EA and SI techniques are robust global optimization techniques for solving problems having many local optima; nonetheless, they require long computation time. Further, they suffer from poor convergence performances [38]. On the other hand, local search algorithms can converge in a few iterations but are deficient in a global outlook; they rely on a suitable starting point: It is the major hurdle that can meet LS approaches. Combining a global search procedure and a local one should offer the advantages of both, while offsetting their drawbacks.

Some hybrid algorithms [38,39] are proposed in the literature, such as genetic annealing [40]. The latter was used in many optimisation applications, but in the circuit design field it was applied only to solve problems related to floorplanning and circuit placement [41,44]. According to the knowledge of the authors, such hybrid techniques have not been used to the optimal sizing of analog circuits.

This chapter presents a hybrid optimization technique that merges a genetic algorithm with a local search technique, namely the SA. Fig. 1 depicts the proposed idea. The efficiency of this hybrid approach is first shown through a constrained mathematical test-case. The hybrid algorithm, called hybrid GA/SA, is then used for the optimal sizing of different aspect analogue circuits.

The rest of the chapter is structured as follows. In section 2, the analogue circuit optimization problem is detailed. In sections 3 and 4, GA and SA algorithms are presented. Section 5 highlights the hybrid GA/SA algorithm. In sections 6, 7 and 8 three application examples of optimizing the sizing of analogue circuits are presented: Section 6 deals with the optimal sizing of a positive second generation current conveyor. Section 7 focuses on the optimal design of a fully differential folded cascode transconductance amplifier. Section 8 highlights optimization of an ultra wide band low noise amplifier through its symbolic scattering parameters. Finally, concluding remarks are given in section 9.

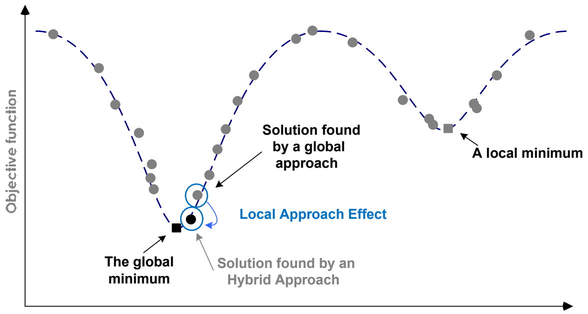

Fig. 1 Effect of combining a LS approach to a GS one.

2 The Analogue Optimization Problem Formulation

A general optimization problem can be defined in the following format:

$$\left|\begin{array}{ll} Minimize\ f(\vec{x}); & f(\vec{x}) \in R \\ such\ that: & \\ \vec{g}(\vec{x}) \leq 0; & \vec{g}(\vec{x}) \in R^{m} \\ and\ \vec{h}(\vec{x}) = 0; & \vec{h}(\vec{x}) \in R^{n} \\ where\ x_{Li} \leq x_i \leq x_{Ui}, & i \in [1, p] \end{array}\right. \tag{1}$$

m, n and p are numbers of inequality constraints to satisfy, equality constraints to assure and parameters to manage, respectively. $\vec{x}_L$ and $\vec{x}_U$ are lower and upper boundary vectors of the parameters.

Classical meta-heuristics cannot handle inequality constraints, so the constrained optimization problem is transformed into an unconstrained one by minimizing the following function:

$$\tilde{f}(x) = f(x) + \gamma(d(x, Fr)) \tag{2}$$

where $d(x, Fr)$ is a distance metric of the infeasible point to the feasible region Fr. It may simply be zero if no constraint violation occurs and is a positive scalar, otherwise [15]. The definition of this distance metric includes the penalty coefficients that are used to stress the importance of a particular constraint violation over others. γ is a weight coefficient to set a global importance of the constraints with respect to the objective function [15].

Optimal design/sizing of analogue circuits consists of finding a variable set $\vec{x} = \{x_1, x_2, \ldots, x_n\}$ that optimizes a performance function(s), such as gain, offset, signal to noise ratio, maximum operating frequency etc., while meeting imposed specifications and/or inherent constraints, for example, saturation conditions of transistors, technology limits, impedance matching, etc. Vector $\vec{x}$ may encompass biases, lengths and widths of MOS transistors, component values etc.

Fig. 2 gives a pictorial view of design optimization approach.

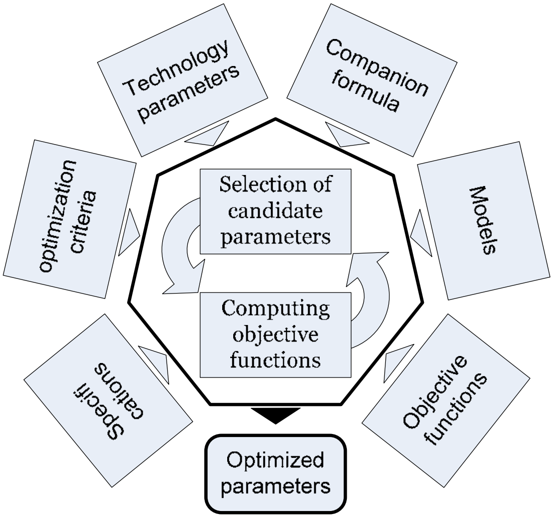

Fig. 2 Pictorial view of a design optimization approach.

Most of analogue optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation. Therefore, classical optimization procedures are generally not adequate, as it was highlighted in section 1. Meta-heuristics allow solving multi-criterion large size problems. They can be adapted to specific problem requirements. Even though they don't guarantee finding the 'exact' optimal solution, they give a 'good' approximation of it within a tolerable CPU computing time.

In this chapter we focus on two meta-heuristics: a Genetic Algorithm and the Simulated Annealing technique, and the hybridization of both.

3 The Genetic Algorithms

Genetic algorithms (GAs) [45,46] are inspired by *Darwin*'s theory of evolution; they mimic natural evolution processes to evolve a solution to a problem:

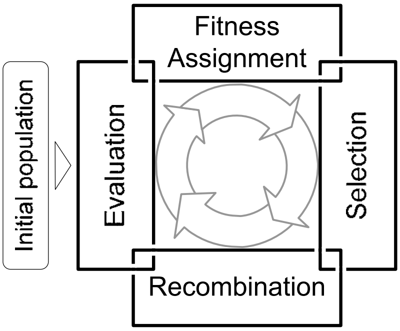

Fig. 3 The basic cycle of evolutionary algorithms.

combination of selection, recombination and mutation. They form a subclass of evolutionary algorithms. Fig. 3 illustrates its basic principle.

where *Evaluation* consists of computing the objective values of the solution candidates. *Fitness Assignment* uses the objective values to determine fitness values. *Selection* chooses the fittest individuals for reproduction, and *Reproduction* creates new individuals from the mating pool by crossover and mutation [15,46].

The pseudo code of a basic GA is given in Fig. 4.

```
Setup the GA.
Main ( )
    InitPopulation ( ) // random initialization of the population
    max_fitness := 0
    for each member chromosome
            fitness := Fitness_Evaluation (chromosome)
            if fitness > max_fitness
                max_fitness := fitness
                fittest_solution = chromosome
            end if
    end for
    while generation < max_generations
          offspring := Selection&Recombination (parents) // Fig. 5
          fitness := Fitness_Evaluation (offspring)
          if fitness > max_fitness
                max_fitness := fitness
                fittest_solution = offspring
          end if
    save fittest_solution
end
```

Fig. 4 The pseudo-code of a GA.

```
Selection&Recombination ( )
    while num of programs < max_prog/2 Do
        parent_1 := Roulette-Wheel_Selection ( )
        parent_2 := Roulette-Wheel_Selection ( )
        randomly choose x-over point
        child_1 := parent_1 [head] ^ parent_2 [tail]
        child_2 := parent_2 [head] ^ parent_1 [tail]
        for each child
            mutation_pt := Random_Mutation_Point (child)
            new_instr := Random_New_Instruction ( )
            Instruction (mutation_pt, new_instr)
        end for
end
```

Fig. 5 The pseudo-code Selection & Recombination routine.

4 The Simulated Annealing Technique

The simulated annealing technique [26] is inspired by the natural annealing process used in metallurgy. The annealing technique is used to create a solid state by slowly cooling a melted metal. The gradual decrease of the metal temperature produces the crystalline lattice, which minimizes its energy probability distribution.

In fact, when the temperature of a metal is high, the particles within the metal are able to move around, changing the structure of the metal, freely. As the temperature is lowered, the particles are limited in the movements they can make as many movements have a high energy cost and are increasingly limited to only those configurations with lower energy than the previous state.

From an algorithmic point of view, this can be modeled as a random exploration on a search graph. Each vertex of the substance represents a solution (or a state) with a certain fitness value, and the adjacent vertices represent other similar solutions (in the sense that their fitness value is not expected to be significantly different). At each step, the algorithm selects randomly an adjacent vertex. It transits to the new state if the current solution improves the actual state. Worse solutions are not radically rejected, but they may be considered and the algorithm may use the new 'worse' state with a probability determined by the global 'temperature' parameter. This behaviour allows escaping from local minima. As the temperature drops, this becomes less likely and the search drops into a nearby local minimum [26-28].

The pseudo code of the SA technique is given in Fig. 6.

```
Setup the SA.
Main ( )
Choose a random solution $x$ which takes the minimum solution $x_{min}$
($x=x_{min}$)
Evaluate the fitness function $f(x)$
Initialize the temperature $T$
Repeat
  Repeat
     Generate a neighbor $X'$ perturbing the solution $X$.
        Acceptance with the criterion of 'Metropolis'// Fig. 7
   Until thermodynamic equilibrium reaches
Decrease temperature
Until maximum iterations or minimum error criteria is attained
end
```

Fig. 6 The pseudo-code of a SA.

```
If $\Delta f<0$
      Update the current solution ($x' \leftarrow x$)
If not
      Compute P=exp($-\Delta f/T$)
      Generate a random number $R \in [0,1]$ (using an uniform distribution)
      If $R \leq P$
      Accept the new solution $x'$ and update $x$
      If not reject the new solution $x'$
      end if
end if
```

Fig. 7 The Metropolis pseudo-code.

Table 1 Advantages of GA and SA

GA	SA
Optimizes with continuous or discrete variables	Very simple to be set up
Deals with a large number of variables	Does not require memory (past) in order to find spaces to seek local following (future)
Optimizes variables with extremely complex –cost surfaces	Can deal with arbitrary systems and cost functions

Table 2 Drawbacks of GA and SA

GA	SA
No guaranteed optimal solution in a finished time,	Important choice of the beginning solution: Knowledge of the problem
Initialization of several parameters, important choice of the methods,	It is necessary to determinate the parameters by hand: initial temperature, elementary modification... by testing various values

5 The Hybrid GA/SA

Merging a local search technique and an evolutionary approach is a very fertile area [39]. Indeed, the idea consists of taking maximum benefits from the robustness of the search technique of a LS method, and those of the recombination process of an evolutionary algorithm (such hybrid algorithms are also known as *memetic* algorithms). At each *n* iterations (*n* is a prefixed number), the algorithm applies the LS method to the elements of the (current) population, and then recalls the recombination mechanism in order to generate new elements. In other words, the approach consists of injecting each *n* iterations a potential solution, obtained using SA techniques, in the GA population, in order to refine the whole solution. It is also to be noticed that accordingly, the algorithm is capable to overcome the premature convergence of the GA algorithm and escape from local optimal solutions.

Tables 1 and 2 summarize main advantages and drawbacks of both GA and SA, respectively [47,48], and Fig. 8 presents the GA/SA flowchart.

In order to show the viability of the proposed approach, the hybrid GA/SA was evaluated using the following test functions [47]. Test functions are given by (3), (4) and (5). Their plots are given in Fig. 9, Fig. 11 and Fig. 13. Corresponding algorithm parameters are given in Table 3.

Table 3 The algorithm parameters

Population size	100
Mutation rate	0.01
Crossover rate	0.9
Initial temperature	1
Stopping temperature	10^{-6}
Cooling schedule	0.9

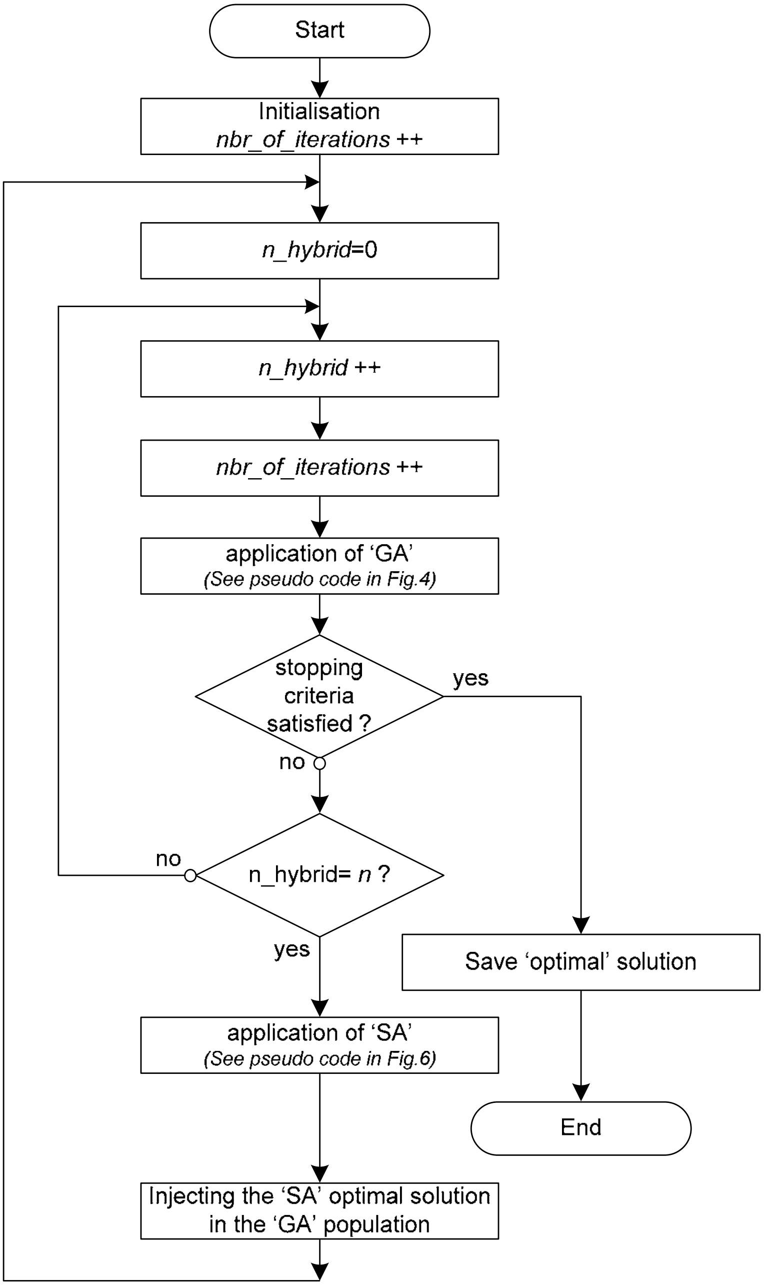

Fig. 8 Flowchart of a hybrid GA/SA approach.

Test function #1:

$$\left| \begin{array}{ll} \mathit{Minimize} & f(x,y)=y\sin(4x)+1.1\,y\sin(2y) \\ \mathit{subject\ to} & x\in[0,10],\ y\in[0,10] \end{array} \right. \tag{3}$$

The goal is to find the global minimum of *f(x,y)* among the large number of local minima.

SA, GA and GA/SA algorithms were applied to *f(x,y)*. Fig. 10 shows a comparison between results obtained using GA/SA and GA, where the rapid convergence of the hybrid algorithm can be noticed. Optimal parameters are (*x*,*y*)=(0.9039,0.866), and the optimal fitness is *f*(*x*,*y*)=-18.554. Both GA/SA and SA give (*x*,*y*)=(0.9040,8.664) and *f*(*x*,*y*)=18.559, in 0.9 sec and 14 sec[♦], respectively.

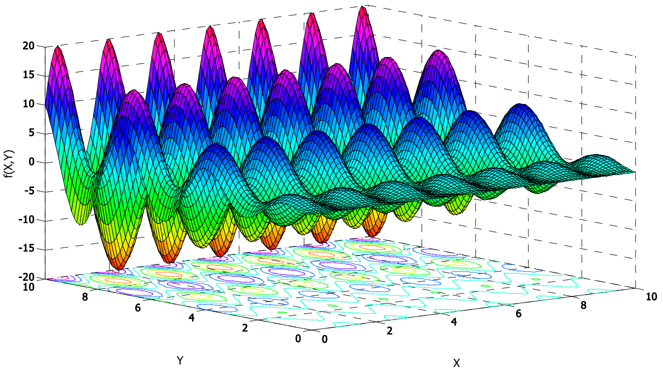

Fig. 9 Plot of the test function #1.

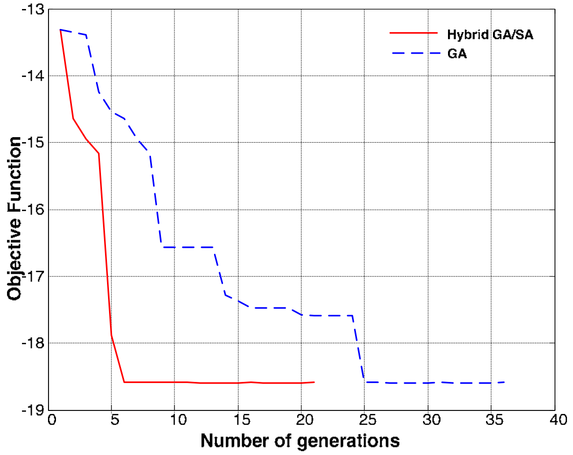

Fig. 10 Optimizing *f(x,y)* (function #1): A comparison between GA and GA/SA results.

[♦] a (2 GHz, 2 Go RAM) core 2 DUO PC was used for this purpose. The same conditions are respected for the three test functions.

Test function #2:

$$\begin{cases} Minimize \quad f(x,y) = 0.5 + \dfrac{sin^2\left(\sqrt{x^2+y^2}\right) - 0.5}{1+0.1(x^2+y^2)} \\ subject\ to \quad x \in [-10,10],\ y \in [-10,10] \end{cases} \tag{4}$$

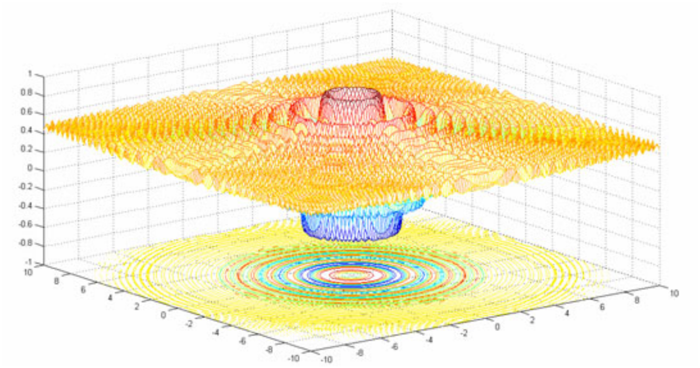

Fig. 11 Plot of the test function #2.

Fig. 12 shows a comparison between results obtained using GA/SA and GA. The rapid convergence of the hybrid algorithm is to be noticed. Optimal parameters are (*x*,*y*)=(1.897,1.006). The optimal fitness is *f*(*x*,*y*)=-0.523. Obtained 'optimal' parameters are (2.107,0.362) and (1.754,1.487) for GA/SA and SA, respectively. Reached fitnesses are respectively -0.522 and -0.374. Convergence times are 1.2 sec (GA/SA) and 20 sec (SA).

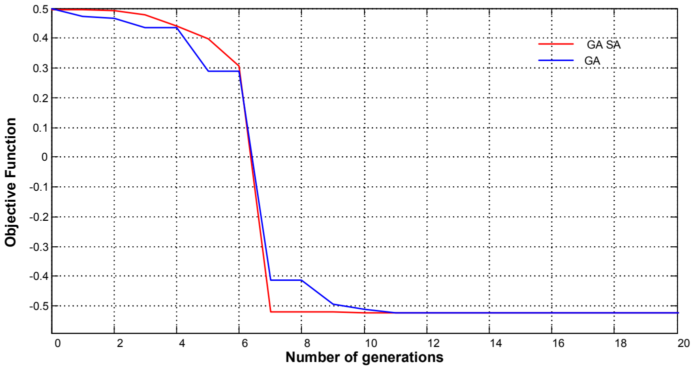

Fig. 12 Optimizing *f(x,y)* (function #2): A comparison between GA and GA/SA results.

Test function #3:

$$\left|\begin{array}{ll} Minimize & f\left(x,y\right)=-x.sin(\sqrt{\left|x-(\,y+9\,)\right|}-(\,y+9\,).sin(\sqrt{\left|y+0.5.x+9\right|} \\ subject\ to & x\in\left[-20,20\right],\ y\in\left[-20,20\right] \end{array}\right. \qquad (5)$$

Fig. 14 shows a comparison between results obtained using GA/SA and GA, where the rapid convergence of SA/GA algorithm can be easily noticed. For the test function #3, the optimal parameters are (-14.580,-20.000) and the fitness is -23.806. GA/SA gives (-17.007,-20.730) and -25.230 in 0.7 sec, whereas SA gives (-15.356,20.475) and a fitness of -24.646 in 12 sec.

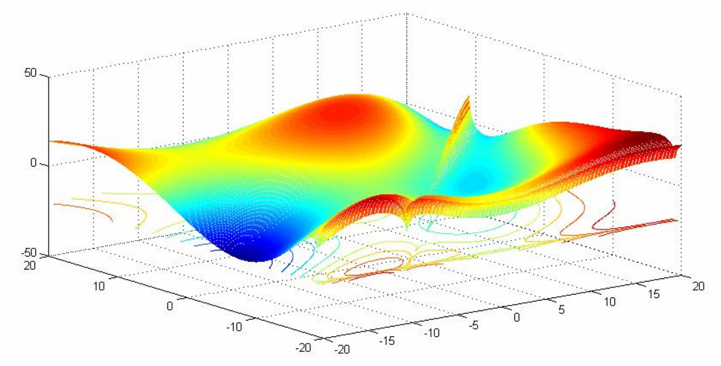

Fig. 13 Plot of the test function #3.

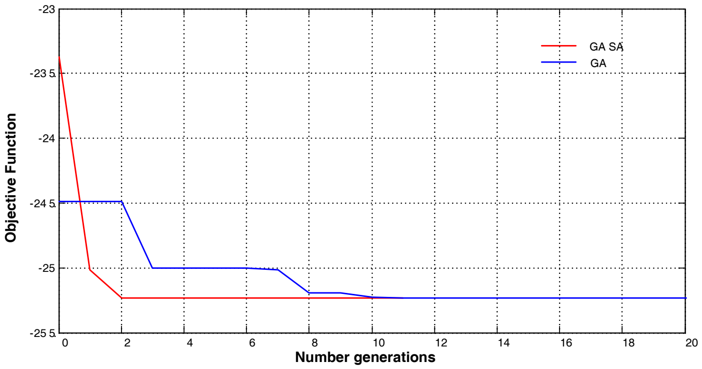

Fig. 14 Optimizing *f(x,y)* (function #3): A comparison between GA and GA/SA results.

6 Application of GA/SA to the Optimization Current Conveyors

Current conveyors are the most well known current mode circuits. They can perform many analogue signal processing functions. Besides, they simplify in many ways the design of analogue circuits in comparison to their voltage mode counterparts, i.e. operational amplifiers [3]. Current conveyors (CC) can be represented as shown in Fig. 15.

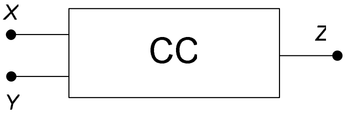

Fig. 15 General representation of current conveyor.

The electric behaviour of a positive second generation current conveyor (CCII+) [3,49] is described as follows:

$$\begin{cases} V_X = V_Y \\ I_Y = 0 \\ I_Z = I_X \end{cases} \tag{6}$$

Fig. 16 presents a translinear loop based CMOS CCII+ [3], where transistors M_1-M_4 instantiate the translinear loop and insure V_X=V_Y. I_0 is the bias current and transistors M_9-M_{12} are current mirrors. Transistors M_5-M_8 reproduce the current applied to pole X, at pole Z.

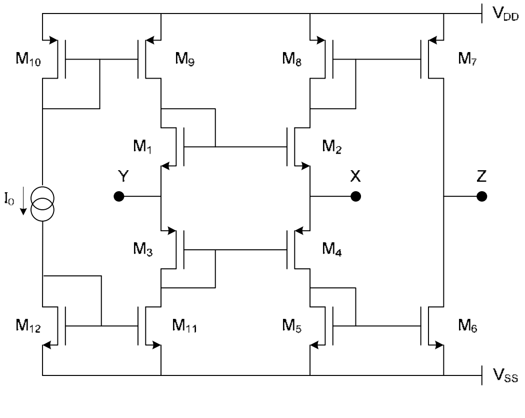

Fig. 16 A CMOS positive second generation current conveyor.

It has been confirmed that current bandwidth limits the frequency application range of a current conveyor, since the voltage frequency range is intrinsically higher than the current one [3]. Thus, in this example we focus on maximizing the current high cut off frequency (f_{ci}) of the CCII+.The symbolic expression of f_{ci} is not given due to their large number of terms.

MOS transistors forming the CCII+ have to operate in the saturation mode. Expressions (7) and (8) give these transistors' saturation conditions for all the MOS transistors:

$$\frac{V_{DD}}{2} - V_{Tn} - \sqrt{\frac{8I_0}{\mu_n C_{ox}\left(\frac{W_i}{L_i}\right)_N}} > \sqrt{\frac{8I_0}{\mu_p C_{ox}\left(\frac{W_i}{L_i}\right)_P}} \tag{7}$$

$$\frac{V_{DD}}{2} - V_{Tp} - \sqrt{\frac{2I_0}{\mu_p C_{ox}\left(\frac{W_i}{L_i}\right)_P}} > \sqrt{\frac{2I_0}{\mu_n C_{ox}\left(\frac{W_i}{L_i}\right)_N}} \tag{8}$$

where μ_n, μ_p and C_{ox} are MOS technology parameters. V_{Tn} and V_{Tp} represent respectively the threshold voltages of NMOS and PMOS transistors. I_0 is the bias current and V_{DD} is the supply voltage. W_i/L_i is the aspect ratio of the corresponding MOS transistor.

GA and GA/SA were applied to maximize f_{ci}. Fig. 17 depicts a comparison between obtained results, where the rapid convergence of the hybrid algorithm can be clearly noticed. Fig. 18 presents SA results (f_{ci} vs stages of temperature). Table 4 gives the algorithm parameters.

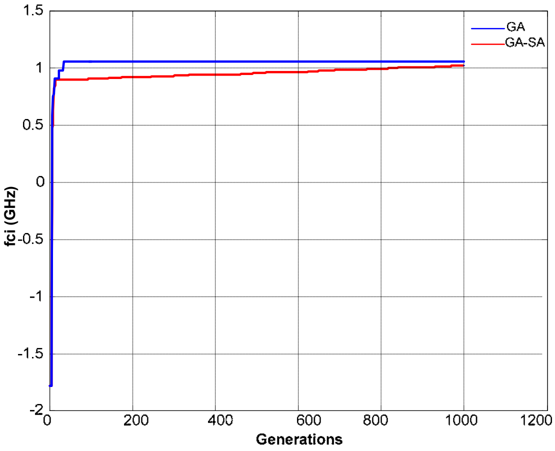

Fig. 17 Optimizing f_{ci}: A comparison between GA and GA-SA results.

Fig. 19 shows SPICE simulation results of the CCII+ current transfer corresponding to the optimal values of the CCII+ parameters.

Table 4 The algorithm parameters of GA/SA

Population size	1000
Numbers of iteration	1000
Mutation rate	0.01
Crossover rate	0.8
Initial temperature	1
Stopping temperature	10^{-6}
Cooling schedule	0.95

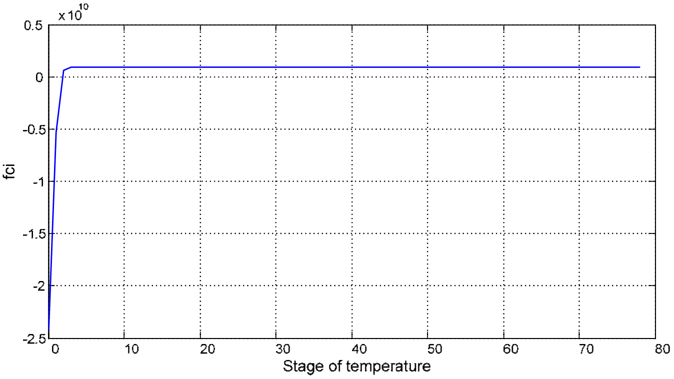

Fig. 18 Optimizing f_{ci} using SA: f_{ci} vs stage of temperature.

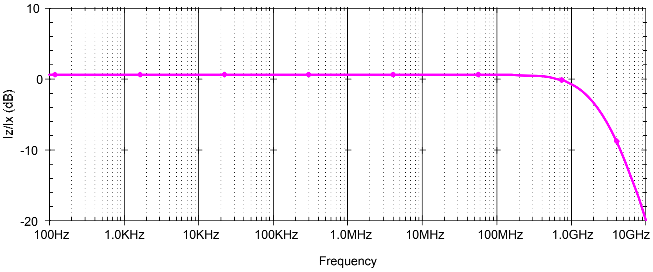

Fig. 19 SPICE simulation results: f_{ci}=931 MHz.

Table 5 presents the optimal device sizing and performances obtained by GA/SA and SA.

Table 5 Optimal device sizing and obtained performances: a comparison.

	GA/SA	**SA**
Technology	0.35 µm AMS	0.35 µm AMS
Voltage supply (V)	±2.5	±2.5
Wn(µm)/Ln(µm)	30.25/0.52	19.30/0.54
Wp(µm)/Lp(µm)	49.95/0.35	33.34/0.35
GBW (GHz)	1.1	0.924
Running Time (min)	0.68	1.25

Finally, and in order to highlight robustness of the proposed hybrid algorithm, Fig. 20 presents results obtained for 100 runs of the hybrid algorithm. The mean value of f_{ci} is 0.956 GHz.

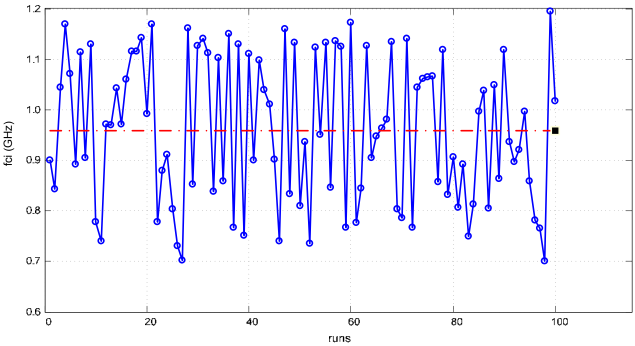

Fig. 20 Values of f_{ci} for 100 runs.

7 Application of GA/SA to the Optimization of Operational Transconductance Amplifiers

The operational transconductance amplifier (OTA), whose schematic symbol is represented in Fig. 21, is an amplifier whose output current is proportional to the differential input voltage. Expression (9) gives the linear function between the differential input voltage and the output current.

$$I_{out} = (V_{in+} - V_{in-})g_m \tag{9}$$

where V_{in+} and V_{in-} are voltages at the non-inverting and the inverting inputs, respectively. g_m is the transconductance of the amplifier.

The amplifier's output voltage is expressed as follows:

$$V_{out} = I_{out}\, R_L \tag{10}$$

R_L is the load of the OTA.

The voltage gain is then the output voltage divided by the differential input voltage:

$$A_V = \frac{V_{out}}{(V_{in+} - V_{in-})} = R_L\ g_m \tag{11}$$

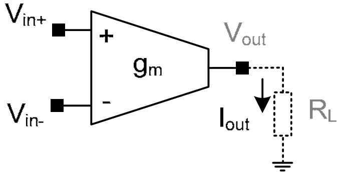

Fig. 21 Schematic symbol for the OTA.

OTAs are generally used to drive small capacitive loads at high frequencies. An OTA is basically an Op-Amp without any output buffer, preventing it from driving resistive or large capacitive loads. They are preferred over op-amps mainly because of their smaller size and their simplicity [50]. Typically, OTAs can be classified into two basic architectures, namely folded-cascode OTAs and telescopic OTAs. The advent of deep submicron technologies enables increasingly high speed circuits, but makes designing high DC gain OTAs more difficult [51]. In this application we deal with maximizing the voltage gain of a fully differential folded cascode OTA (FDFC). Fig. 22 shows the CMOS implementation of a FDFC.

This OTA uses cascoding in the output stage combined with an unusual implementation of the differential amplifier to achieve good input common-mode range. Thus, the folded cascode OTA offers self-compensation, good input common-mode range, and the gain of a two-stage OTA [52].

The open-loop voltage gain (A_v) and the unity-gain frequency (F_t) of the FDFC are given respectively by equations (12) and (13):

$$A_v = g_{m9} R_{out} \tag{12}$$

$$F_t = g_{m9} / 2\pi C_L \tag{13}$$

where $R_{out} = R_2 /\!/ (g_{m3} r_{03} R_1)$, $R_1 = r_{01} /\!/ r_{09}$ and $R_2 = g_{m3} r_{05} r_{07}$. g_{m3} and g_{m9} are respectively the transconductances of transistors M_3 and M_9. r_{01}, r_{03}, r_{05}, r_{07} and r_{09} are

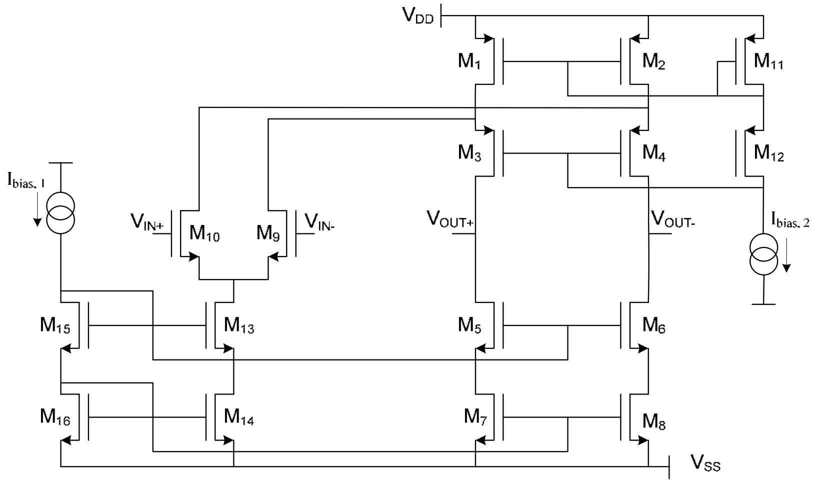

Fig. 22 Folded cascade OTA.

respectively the drain-source resistances of transistors M_1, M_3, M_5, M_7 and M_9. C_L is the load capacitance.

The problem consists in maximizing the voltage gain A_v while satisfying a set of inherent constraints, i.e. MOS saturation conditions, whose expressions are given by Eqns. (14)-(18).

$$V_{DD} - V_{outMAX} \geq \sqrt{\frac{3\,I_{bias1}}{K_P\left(\frac{W_1}{L_1}\right)}} + \sqrt{\frac{2\,I_{bias1}}{K_P\left(\frac{W_3}{L_3}\right)}} \tag{14}$$

$$V_{out\min} - V_{SS} \geq \sqrt{\frac{I_{bias1}}{K_N\left(\frac{W_5}{L_5}\right)}} + \sqrt{\frac{I_{bias1}}{K_N\left(\frac{W_7}{L_7}\right)}} \tag{15}$$

$$V_{DD} - V_{inMAX} + V_{TN} \geq \sqrt{\frac{3\,I_{bias1}}{K_P\left(\frac{W_1}{L_1}\right)}} \tag{16}$$

$$V_{in\min} - V_{SS} - V_{TN} \geq \sqrt{\frac{I_{bias1}}{K_N\left(\frac{W_9}{L_9}\right)}} + \sqrt{\frac{2\,I_{bias1}}{K_N\left(\frac{W_{13}}{L_{13}}\right)}} + \sqrt{\frac{2\,I_{bias1}}{K_N\left(\frac{W_{14}}{L_{14}}\right)}} \tag{17}$$

$$Slew\ Rate = \frac{I_{bias1}}{C_L} \tag{18}$$

where, I_{bias1} is the bias current, W_i and L_i are respectively widths and lengths of the corresponding transistors, V_{DD} and V_{SS} are the supply voltages, K_N and K_P are technology parameters. V_{TN} is the NMOS threshold voltage.

The static and dynamic performances of the OTA are set according to the specifications of high gain, wide band applications given in Table 6 [53]. The optimization problem consists of maximizing the voltage gain A_v while satisfying a set of constraints, namely the saturation conditions of MOS transistors and the frequency bandwidth.

Table 6 OTA circuit specifications

Technology	CMOS AMS 0.35µm
F_t (MHz)	≥ 200MHz
C_L (pF)	0.1
V_{DD}/V_{SS} (V)	-1.8/+1.8
Slew Rate (V/µsec)	≥ 200

Table 7 gives optimal device sizing obtained thanks to GA/SA and GA, the circuit's performance, and comparison with two published works. I_{bias1} equals 60µA. This value gives a slew rate that equals 300V/µsec. I_{bias2} directly depends on the ratio between aspect ratios of the current mirrors.

Table 7 Optimal device sizing of the FDFC OTA

	GA/SA	**SA**	**[54]**	**Gm/Id** [55]
Technology	0.35 µm AMS	0.35 µm AMS	0.35 µm AMS	0.35 µm AMS
Voltage supply (V)	±1.8	±1.8	±1.8	±2
W1(µm)/L1(µm)	50/1	44.03/1	34.85/1	10.8/1
W3(µm)/L3(µm)	33/1	29.06/1	23/1	5.4/1
W5(µm)/L5(µm)	50/1	47.15/1	47.15/1	2/1
W9(µm)/L9(µm)	50/1	49.9/1	49.9/1	14/1
DC Gain (dB)	84.42	83.89	82.89	77.53
Running Time (min)	0.086	0.47	--	--

Fig. 23 shows a comparison between results obtained using GA and hybrid GA/SA when maximizing the open-loop voltage gain (A_v), where the rapid convergence of the hybrid algorithm can be clearly noticed. Fig. 24 presents SA results: Av vs. stages of temperature.

Table 8 gives the GA/SA algorithm parameters.

Table 8 The algorithm parameters of GA/SA

Population size	2000
Numbers of iteration	2000
Mutation rate	0.01
Crossover rate	0.8
Initial temperature	1
Stopping temperature	10^{-6}
Cooling schedule	0.95

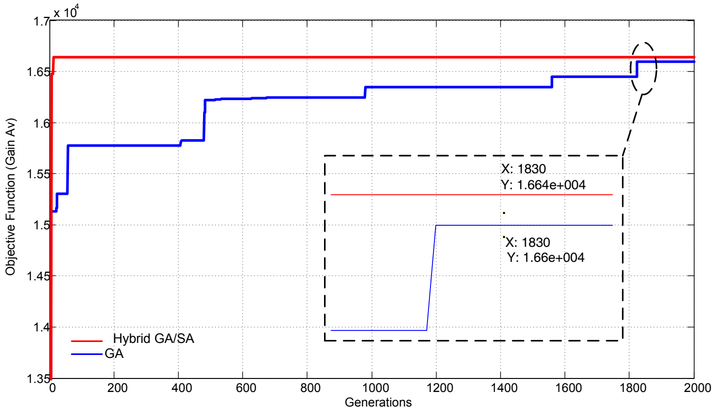

Fig. 23 Optimizing the FDFC OTA Gain: A comparison between GA and GA-SA algorithms.

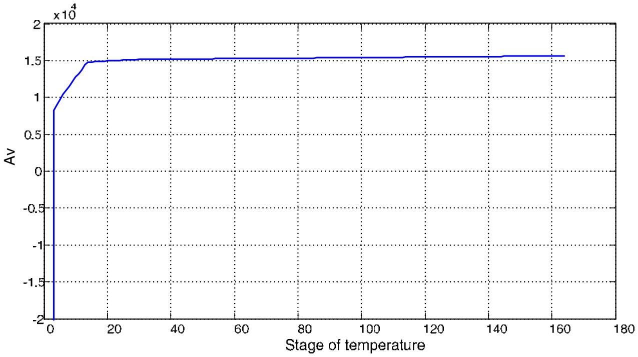

Fig. 24 Optimizing the OTA gain using SA: Av vs stage of temperature.

Table 9 gives a comparison between theoretical (hybrid GA/SA) and SPICE simulation results.

Table 9 Theoretical and simulation results of hybrid GA/SA

Performances	Theoretical values (MATLAB)	Simulation values (SPICE)
A_v(dB)	84.42	84.33
F_t (MHz)	534	517

These values show the good agreement between the SPICE simulation results and the theoretical ones obtained by applying hybrid GA/SA.

Fig. 25 shows SPICE simulation results of the FDFC OTA.

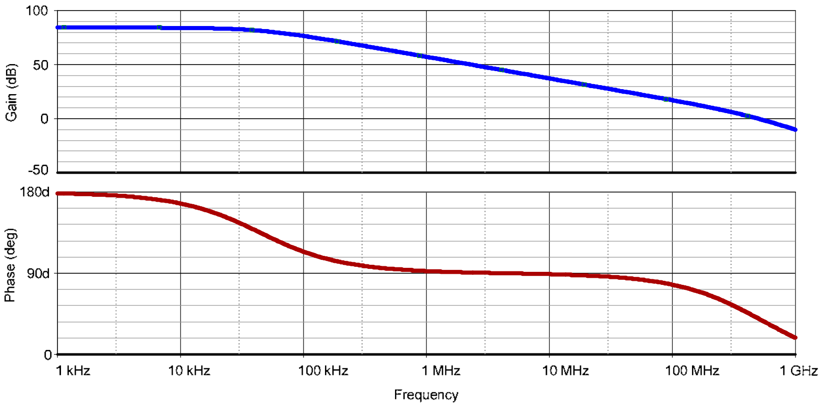

Fig. 25 Spice simulation results for Gain and Phase curve

The folded cascode OTA presents a high gain (84.33 dB), a large unity-gain frequency (517.030 MHz) and a good linearity depicted by its phase margin (51.34°).

Finally, Fig. 26 presents results obtained for 100 runs of the GA/SA algorithm. The mean value of Av is 1503.

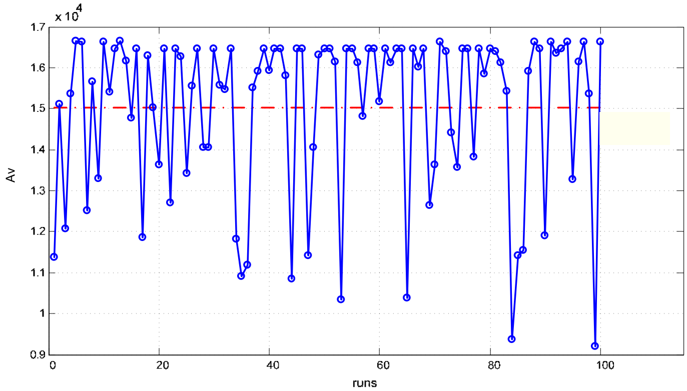

Fig. 26 Values of Av for 100 runs.

8 Application of GA/SA to the Optimization of a RF Circuit

Low Noise Amplifiers (LNA) form an important block in any Radio Frequency chain [56]. The LNA is responsible of amplifying the signal and minimizing the noise [57]. Scattering parameters allow characterizing a LNA circuit [56]. In fact, these parameters reflect the power gain, the input matching and the output matching of the LNA. It has been shown that the symbolic expressions of the scattering parameters can deducted from the symbolic expressions of the Impedance parameters (*Z-parameters*) [59].

The definition of the scattering parameters, noticed *S-Parameters*, is based on the theory of the incident and reflected waves [59,60]. Thus, *S-parameters* describe the relationship between the different waves of a system. Fig. 27 represents a network with two ports including the incident and reflected microwaves [61].

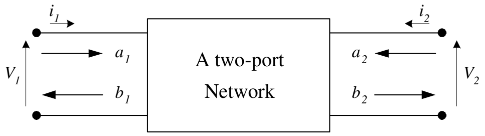

Fig. 27 A two-port network with incident and reflected waves.

a_1 and a_2 represent the electric field of the microwave signal entering the network input and output respectively. b_1 and b_2 represent the electric field of the microwave signal leaving the network input and output respectively.

Consequently, the *S-parameters* are defined by the following expressions [61]:

$$S_{11} = \left.\frac{b_1}{a_1}\right|_{a_2=0}, \quad S_{21} = \left.\frac{b_2}{a_1}\right|_{a_2=0}, \quad S_{12} = \left.\frac{b_1}{a_2}\right|_{a_1=0}, \quad S_{22} = \left.\frac{b_2}{a_2}\right|_{a_1=0} \tag{19}$$

S_{11}, S_{21}, S_{12}, S_{22} are the input reflection coefficient, the forward transmission coefficient, the reverse transmission coefficient, and the output reflection coefficient, respectively.

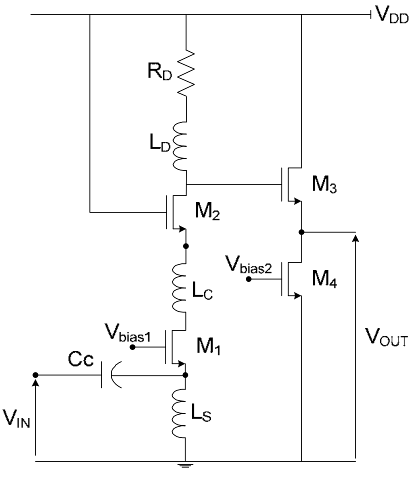

Fig. 28 UWB Common Gate Low Noise Amplifier.

Fig. 28 shows a Common Gate Low Noise Amplifier [62]. This structure is dedicated to the Ultra Wide Band (UWB) standard. The frequency band of this standard is defined from 3 GHz to 10 GHz. The input impedance of this structure is formed essentially by the inductance L_S and the equivalent impedance of the transistor M_1. The transistor M_2 increases low frequency gain and improves isolation. However, the parasitic capacitances of the transistor M_2 decrease the gain at high frequency. Therefore, the role of the inductance L_C is to balance this degradation. Transistors M_3 and M_4 improve the output matching of the structure. The inductance L_D and the resistance R_D permit to obtain a flat gain along the frequency band [3, 10] GHz. Due to the large number of terms of the *S-parameters*, their symbolic expressions are not given; they were generated using the symbolic analyser CASCADES.1 [63,64].

The hybrid GA/SA approach was used to compute the optimal sizing of the transistors forming the common gate LNA and the optimal values of the biases. The objective is to maximize the voltage gain (S_{21}). Design and inherent constraints (maximum acceptable noise figure value, MOS saturation conditions, minimum transition frequency f_T, and impedance matching) are given by expressions (20)-(27).

$$\langle Noise\_Figure \rangle_{dB} < 1dB \qquad (20)$$

where

$$Noise\_Figure = 1 + \frac{\gamma}{\alpha g_{m1} R_s} + \frac{\delta\alpha}{5 g_{m1} R_s}\left(\frac{\omega}{\omega_T}\right)^2 + \frac{R_D}{\left(\omega^2 L_D{}^2 + R_D{}^2\right) R_S g_{m1}{}^2 \left(\frac{|Z_{11}|}{|Z_{11}| + R_S}\right)^2}$$

γ, α and δ are process dependant parameters, g_{m1} is the transconductance of transistor M_1, R_S is the terminal (load) resistance which is equal to 50Ω, ω and ω_T are the pulsation and the transit pulsation, respectively. $|Z_{11}|$ is the input impedance of the LNA.

$$(V_{DD} - V_{TN2} - \sqrt{\frac{2 I_{d1} L_2}{\mu_n C_{ox} W_2}} - |S_{21}| V_{inMAX}) > V_{gs1} - V_{TN1} \tag{21}$$

$$(V_{gs2} - R_D I_{d1} - V_{ds1\min} - |S_{21}| V_{inMAX}) > V_{gs1} - V_{TN1} \tag{22}$$

$$(V_{TN3} - \sqrt{\frac{2 I_{d2} L_3}{\mu_n C_{ox} W_3}} - V_{ds4\min}) > V_{gs3} - V_{TN3} \tag{23}$$

$$(V_{DD} - V_{TN3} - \sqrt{\frac{2 I_{d3} L_3}{\mu_n C_{ox} W_3}} - |S_{21}| V_{inMAX}) > V_{gs4} - V_{TN4} \tag{24}$$

$$f_T > 5 f_0 \tag{25}$$

where $f_T = \frac{1}{2\Pi}\frac{g_m}{C_{gs}} = \frac{3}{4\Pi}\frac{\mu_n}{L^2}(V_{gs} - V_{TN})$, f_0 is the central frequency of the considered band, i.e. 6.3 GHz. V_{TNi} is the threshold voltage, V_{inMAX} is the input maximal voltage and V_{gsi} are gate to source voltages of the corresponding transistor (1≤i≤4).

$$|S_{11}|_{dB} < -10dB \tag{26}$$

$$|S_{22}|_{dB} < -10dB \tag{27}$$

Fig. 29 presents a comparison between results obtained using GA and GA-SA. Fig. 30 shows SA results: S_{21} vs. stages of temperature. Table 10 gives the GA/SA algorithm parameters. Table 11 presents values of the optimized parameters and the reached performances.

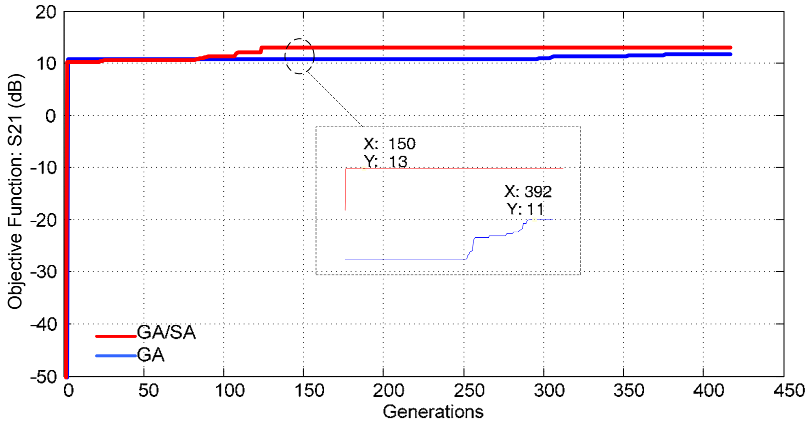

Fig. 29 A comparison between GA and GA/SA results.

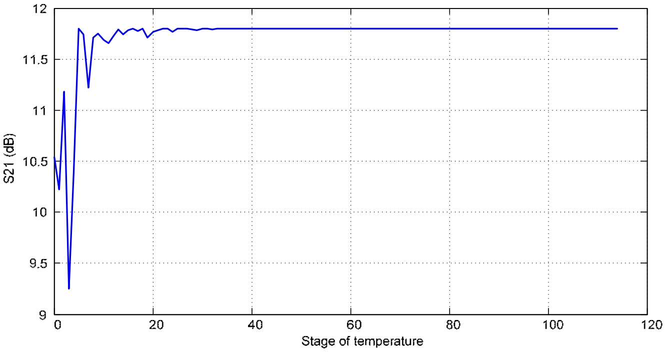

Fig. 30 Optimizing the LNA scattering parameter S_{21} using SA: S_{21} vs stage of temperature.

Table 10 The GA/SA algorithm parameters

Population size	400
Numbers of iteration	400
Mutation rate	0.01
Crossover rate	0.8
Initial temperature	1
Stopping temperature	10^{-6}
Cooling schedule	0.95

Table 11 Optimal device sizing and reached performances for the LNA.

	GA/SA	SA
Technology	0.35 µm AMS	0.35 µm AMS
W_1 (µm)/L_1 (µm)	772/0.35	781/0.35
W_2 (µm)/L_2 (µm)	772/0.35	781/0.35
W_3 (µm)/L_3 (µm)	34.07/0.35	31/0.35
W_4 (µm)/L_4 (µm)	34.07/0.35	31/0.35
Id_1 (mA)	5.23	5.18
Id_2 (pA)	6.23	6.23
S_{21}	13.0 dB	11.8 dB
S_{11}	-10.9 dB	-11.6 dB
S_{22}	-15.0 dB	-18.8 dB
Running time (min)	5.2	11.4

Table 12 gives a comparison between theoretical (GA/SA) and simulation (Advanced Design System :ADS) results.

Table 12 Theoretical and simulation results

	Theoretical values (MATLAB)	Simulation values (ADS)
S_{21}	13.0 dB	12.5 dB
S_{11}	-10.9 dB	-12.8 dB
S_{22}	-15.0 dB	-12.6 dB

Figs 31-33 show the good agreement between ADS simulation results and the theoretical ones (MATLAB) obtained by applying hybrid GA/SA.

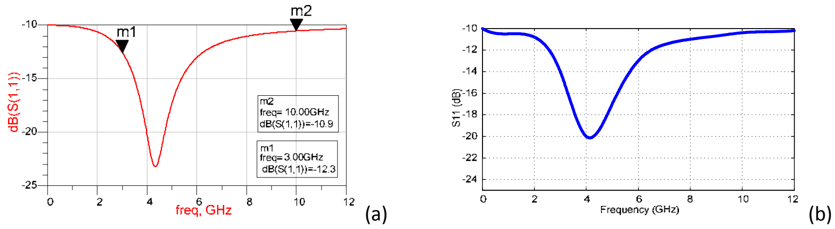

Fig. 31 S_{11}=*f*(frequency): (a) ADS, (b) MATLAB

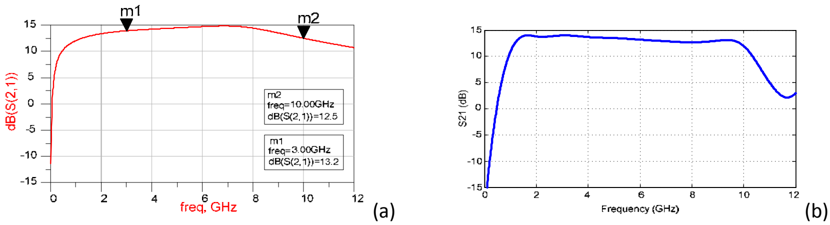

Fig. 32 S_{21}=*f*(frequency): (a) ADS, (b) MATLAB

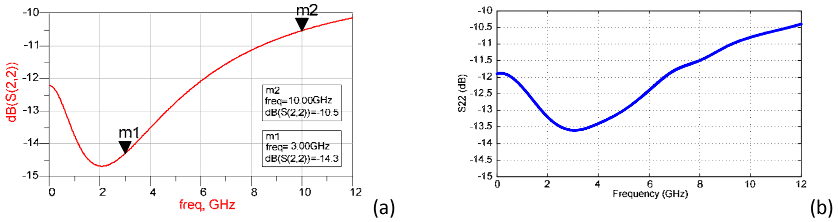

Fig. 33 S_{22}=*f*(frequency): (a) ADS, (b) MATLAB.

Finally, Fig. 34 presents results obtained for 100 runs of the GA/SA algorithm. The mean value of S_{21} is 12.60 dB.

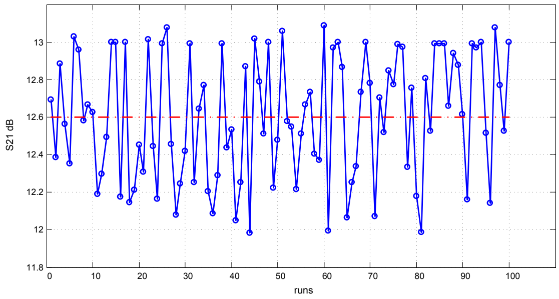

Fig. 34 Values of S_{21} for 100 runs.

9 Conclusion

In this chapter, the hybridization of a genetic algorithm and the simulated annealing technique, and its application to the optimal design of analogue circuits, are proposed. The GA/SA algorithm was implemented in MATLAB. First, it was evaluated using some test functions and advantages of such hybridization, when compared to conventional meta-heuristics, were highlighted. The hybrid GA/SA algorithm was used to optimize the sizing of three typical analogue circuits, namely a positive second generation CMOS current conveyor, a fully differential folded cascode operational transconductance amplifier, and a radio-frequency circuit, i.e. a low noise amplifier. Improvements obtained thanks to the use of GA/SA, in terms of CPU computing time, were highlighted. Further, robustness of the proposed hybrid approach was tested and simulation results (SPICE/ADS) were given to show concordance with theoretical results.

References

[1] Gielen, G., Sansen, W.: Symbolic Analysis for Automated Design of Analog Integrated Circuits. Kluwer Academic Publishers, Dordrecht (1991)

[2] Fakhfakh, M., Tlelo-Cuautle, E., Fernández, F.V.: Design of Analog Circuits through Symbolic Analysis. Bentham Scientific Publisher (2010); eISBN: 978-1-60805-095-6

[3] Toumazou, C., Lidgey, F.J., Haigh, D.G.: Analog Integrated Circuits: The current mode approach. IEEE circuit and systems series 2 (1993)

[4] Roca, E., Fakhfakh, M., Castro-López, R., Fernández, F.V.: Applications of Evolutionary Computation Techniques to Analog, Mixed-Signal and RF Circuit Design – An Overview. In: The IEEE International Conference on Electronics, Circuits, and Systems, ICECS, Tunisia (2009)

[5] Tlelo-Cuautle, E., Duarte-Villasenor, M.A.: Evolutionary electronics: automatic synthesis of analog circuits by GAs. In: Yang Ang, B.L.T., Yin, S. (eds.) Success in Evolutionary Computation. SCI, pp. 165–188. IGI Global (2008)

[6] Liu, B., Fernández, F.V., Gielen, G., Castro-Lopez, R., Roca, E.: A Memetic Approach to the Automatic Design of High-Performance Analog Integrated Circuits. ACM Trans on Design Automation of Electronic Systems 14(3) Article 42 (2009)

[7] Medeiro, F., Rodríguez-Macías, R., Fernández, F.V., Domínguez-Castro, R., Huertas, J.L., Rodríguez-Vázquez, A.: Global Design of Analog Cells Using Statistical Optimization Techniques. Analog integrated circuits and signal processing 6, 179–195 (1994)

[8] Fakhfakh, M., Loulou, M., Masmoudi, N.: A Novel Heuristic for Multi-Objective Optimization of Analog Circuit Performances. Analog Integrated Circuits & Signal Processing 61(1), 47–64 (2009)

[9] Fakhfakh, M.: A Novel Alienor-Based Heuristic for the Optimal Design of Analog Circuits. Microelectronics Journal 40(1), 141–148 (2009)

[10] Talbi, E.G.: A Taxonomy of Hybrid Metaheuristics. Journal of Heuristics 8(5), 541–564 (2002)

[11] Siarry, P., Michalewicz, Z.: Advances in Metaheuristics for Hard Optimization, pp. 1619–7127. Springer, Heidelberg (2007); ISSN: 1619-7127

[12] Abraham, A., Hassanien, A.E., Siarry, P., Engelbrecht, A.: Foundations of Computational Intelligence Volume 3: Global Optimization. Springer, Heidelberg (2009); ISSN:1860949X

[13] Dréo, J., Petrowski, A., Siarry, P., Taillard, E.: Metaheuristics for Hard Optimization: methods and case studies. Springer, Heidelberg (2006); ISBN: 978-3-540-23022-9

[14] Daniel, A.: Evolutionary Computation for Modeling and Optimization. Springer, Heidelberg (2006); ISBN: 978-0-387-22196-0

[15] Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing. Springer, Heidelberg (2007); ISBN: 978-3-540-40184-1

[16] Grimbleby, J.B.: Automatic analogue circuit synthesis using genetic algorithms. IEE Proceedings-Circuits, Devices and Systems 147, 319–323 (2000)

[17] Dinger, R.H.: Engineering design optimization with genetic algorithm. In: IEEE Northcon Conference, Seattle WA, USA (1998)

[18] Marseguerra, M., Zio, E.: System design optimization by genetic algorithms. In: The IEEE Annual Reliability and Maintainability Symposium, Los Angeles, California USA (2000)

[19] Fogel, L., Owers, A., Walsh, M.: Artificial intelligence through simulated evolution. Wiley, Chichester (1996); ISBN-13: 978-0471265160

[20] Chan, F.T.S., Tiwari, M.K.: Swarm Intelligence: focus on ant and particle swarm optimization. I-Tech Education and Publishing (2007); ISBN 978-3-902613-09-7

[21] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference On Neural Networks, pp. 1942–1948.

[22] Clerc, M.: Particle swarm optimization. In: International Scientific and Technical Encyclopedia (2006); ISBN-10: 1905209045

[23] Dorigo, M., Di-Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization. Artificial life Journal 5, 137–172 (1999)

[24] Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 52–67 (2002)

[25] Siarry, P., Berthiau, G., Durdin, F., Haussy, J.: Enhanced simulated annealing for globally minimizing functions of many-continuous Variables. ACM Transactions on Mathematical Software 23, 209–228 (1997)

[26] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Journal of Science 220, 671-220-680 (1983)

[27] Courat, J.P., Raynaud, G., Mrad, I., Siarry, P.: Electronic component model minimization based on Log Simulated Annealing. IEEE Transactions on Circuits and Systems 41, 790–795 (1994)

[28] Durbin, F., Haussy, J., Berthiau, G., Siarry, P.: Circuit performance optimization and model fitting based on simulated annealing. International Journal of Electronics 73, 1267–1271 (1992)

[29] Glover, F.: Tabu search- part I. ORSA Journal on Computing 1(3) Summer (1989)

[30] Glover, F.: Tabu search- part II. ORSA Journal on Computing 2(1) Winter (1990)

[31] Fakhfakh, M., Cooren, Y., Sallem, A., Loulou, M., Siarry, P.: Analog Circuit Design Optimization through the Particle Swarm Optimization Technique. Analog Integrated Circuits & Signal Processing 63(1), 71–82 (2009)

[32] Tlelo-Cuautle, E., Duarte-Villasenor, M.A., Guerra-Gomez, I.: Automatic synthesis of VFs and VMs by applying genetic algorithms. Circuits, Systems, Signal Processing 27, 391–403 (2008)

[33] Tlelo-Cuautle, E., Guerra-Gomez, I., Reyes-Garcıa, C.A., Duarte-Villasenor, M.A.: Synthesis of Analog Circuits by Genetic Algorithms and their Optimization by Particle Swarm Optimization. In: Chiong, R. (ed.) Intelligent Systems for Automated Learning and Adaptation: Emerging Trends and Applications, pp. 173–192. IGI Global (2010), doi: 10.4018/978-1-60566-798-0. ch008

[34] Chu, M., Allstot, D.J.: Elitist nondominated sorting genetic algorithm based RF IC optimizer. IEEE Transactions on Circuits and Systems – I 52, 535–545 (2005)

[35] Yoshida, H., Kawata, K., Fukuyama, H., Takayama, S., Nakanish, Y.: A particle swarm optimization for reactive power and voltage control considering voltage security assessment. The IEEE Transactions on Power Systems 15, 1232–1239 (2001)

[36] Maitra, M., Chatterjee, A.: A novel technique for multilevel optimal magnetic resonance brain image thresholding using bacterial foraging. Elsevier B.V. 41, 1124–1134 (2008)

[37] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 6(21), 1087–1092 (1953)

[38] Kerlner, V., et al.: A hybrid optimization technique coupling an evolutionary and a local search algorithm. Journal of Computational and Applied Mathematics 215, 448–456 (2008); ISSN:0377-0427

[39] Crina, G., Ajith, A., Hisao, I.: Hybrid Evolutionary Algorithms. Springer, Heidelberg (2007)

[40] El-Hosseini, M.A., Hassanien, A.E., Ajith, A., Al-Qaheri, A.: Genetic Annealing Optimization: Design and Real World Applications. In: The International Conference on Intelligent Systems Design and Applications, Kaohsiung, Taiwan (2008)

[41] Kurbel, K., Schneider, B., Singh, K.: Solving optimization problems by parallel recombinative simulated annealing on a parallel computer-an application to standard cell placement in VLSI design. IEEE Transactions on Systems, Man and Cybernetics, Part B 28(3), 454–461 (1998)

[42] Zhang, L., Raut, R., Jiang, Y., Kleine, U.: Placement algorithm in analog-layout designs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25(10), 1889–1903 (2006)

[43] Somani, A., Chakrabarti, P.P., Patra, A.: Mixing Global and Local Competition in Genetic Optimization based Design Space Exploration of Analog Circuits. In: The Design, Automation and Test in Europe Conference, Munich, Germany (2005)

[44] Esbensen, H., Mazumder, P.: SAGA: A unification of the genetic algorithm with simulated annealing and its application to macro-cell placement. In: The International Conference on VLSI Design, Calcutta, India (1994)

[45] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

[46] Weise, T.: Global optimization algorithms – theory and applications (2009), `http://www.it-weise.de`

[47] Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. John Wiley & Sons, Chichester (2004); ISBN 0-471-45565-2

[48] Rutenber, R.A.: Simulated Annealing Algorithms: An Overview. IEEE Circuits and Devices Magazine 5(1), 19–26 (1989)

[49] Sedra, A., Smith, K.C.: A second generation current conveyor and its applications. IEEE Transactions on Circuit Theory 17, 132–134 (1970)

[50] Razavi, B.: Design of Analog CMOS Integrated Circuit. TheMcGraw-Hill Companies, Inc., United States (2001); ISBN:0-07-118815-0

[51] Berntsen, O., Wulff, C., Ytterdal, T.: High Speed, High Gain OTA in a Digital 90 nm CMOS Technology. In: NORCHIP Conference, pp. 129–132 (2005)

[52] Phillip, E., Allen, D., Holberg, R.: CMOS Analog Circuit Design. Oxford University Press, Inc., Oxford (2002)

[53] Zhang, L., Kim, H.J., Nadig, V., Ismail, M.: A 1.8 V tri-mode ΣΔ modulator for GSM/WCDMA/WLAN wireless receiver. Analog Integrated Circuits and Signal Processing 49(3), 323–341 (2006)

[54] Daoud, H., Bennour, S., Ben-Salem, S., Loulou, M.: Low power SC CMFB folded cascode OTA optimization. In: The IEEE International Conference on Electronics, Circuits and Systems, ICECS, Malta (2008)

[55] Daoud, H., Ben-Salem, S., Zouari, S., Loulou, M.: Design of folded cascode OTA in different regions of operation through gm/ID methodology. The world academy of science, engineering and technology 45, 28–33 (2008)

[56] Razavi, B.: RF Microelectronics. Prentice Hall press, Englewood Cliffs (1998)

[57] Andreani, P., Sjoland, H.: Noise optimization of an inductively degenerated CMOS low noise amplifier. IEEE Transactions on Circuits and Systems 48(9), 835–841 (2001)

[58] Ellinger, F.: Radio frequency integrated circuits and technologies. Springer, Heidelberg (2007)

[59] Kurokawa, K.: Power waves and the scattering matrix. IEEE Transactions on Microwave Theory and Techniques 13(2), 194–202 (1965)

[60] Kurokawa, K.: An introduction to the theory of microwave circuits. Academic Press, London (1969)

[61] Scott, A.W.: Understanding Microwaves.TK7876.S36. John Wiley & Sons. Inc., Chichester (1993) ISBN 0-471-57567-4

[62] Heng, Z., Xiaohua, F., Sánchez, S.E.: A Low-Power, Linearized, Ultra-Wideband LNA Design Technique. IEEE Journal of solid-state circuits 44(2), 320–339 (2009)

[63] Fakhfakh, M., Loulou, M.: A Software for the Automated Computing of Symbolic Transfer Functions of Analog Circuits. In: The International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design, Erfurt, Germany (2008)

[64] Fakhfakh, M., Loulou, M.: Live Demonstration: CASCADES.1: a Flow-Graph-Based Symbolic Analyzer. In: The IEEE International Symposium on Circuits and Systems, ISCAS, Paris, France (2010)