

Analog circuit design optimization through the particle swarm optimization technique

Mourad Fakhfakh · Yann Cooren · Amin Sallem ·
Mourad Loulou · Patrick Siarry

Received: 23 September 2008 / Revised: 24 July 2009 / Accepted: 29 July 2009 / Published online: 12 August 2009
© Springer Science+Business Media, LLC 2009

Abstract This paper details the Particle Swarm Optimization (PSO) technique for the optimal design of analog circuits. It is shown the practical suitability of PSO to solve both mono-objective and multiobjective discrete optimization problems. Two application examples are presented: maximizing the voltage gain of a low noise amplifier for the UMTS standard and computing the Pareto front of a bi-objective problem, maximizing the high current cut off frequency and minimizing the parasitic input resistance of a second generation current conveyor. The aptness of PSO to optimize difficult circuit problems, in terms of numbers of parameters and constraints, is shown.

Keywords Metaheuristics · Particle swarm optimization · Mono-objective · Multiobjective · Low noise amplifier · Second generation current conveyor

1 Introduction

Advances in VLSI technology nowadays allow the realization of complex integrated electronic circuits and

systems [1]. Analog components are important part of integrated systems either in terms of elements and area in mixed-signal systems or as vital parts in digital systems [1, 2]. Despite their importance, design automation for analog circuits still lags behind that of digital circuits. As a matter of fact optimal design of analog component is often a bottleneck in the design flow. Analog synthesis is complicated because it does not only consist of topology and layout synthesis but also of component sizing [3]. The sizing procedure is often a slow, tedious and iterative manual process which relies on designer intuition and experience [4]. Optimizing the sizes of the analog components automatically is an important issue towards being able to rapidly design true high performance circuits [2, 4].

Common approaches are generally either fixed topology ones or/and statistic-based techniques (see for instance [5–8]). They generally start with finding a “good” DC operating point, which is provided by the analog ‘expert’ designer. Then a simulation based tuning procedure takes place. However these statistic-based approaches are time consuming and do not guarantee the convergence to the global optimum solution [9].

Some heuristic-based mathematical approaches were also used, such as local search (LS) [10], simulated annealing (SA) [11–14], tabu search (TS) [15, 16], scatter search (SS) [17], genetic algorithms (GA) [11, 18–20] etc. However these techniques do not offer general solution strategies that can be applied to problem formulations where different types of variables, objectives and constraint functions are used (linear/non linear objective and constraint functions, posynomial/signomial objective function, etc.). In addition, their efficiency is also highly dependent on the algorithm parameters, the dimension of the solution space, the convexity of the solution space, and the number of variables.

M. Fakhfakh (✉) · A. Sallem · M. Loulou
University of Sfax, B.P. 1069, Sfax 3018, Tunisia
e-mail: mourad.fakhfakh@ieee.org

A. Sallem
e-mail: sallem_amin@yahoo.fr

M. Loulou
e-mail: mourad.loulou@ieee.org

Y. Cooren · P. Siarry
University of Paris 12, Creteil Cedex, France
e-mail: cooren@univ-paris12.fr

P. Siarry
e-mail: siarry@univ-paris12.fr

Actually, most of the circuit design optimization problems simultaneously require different types of variables, objective and constraint functions in their formulation. Therefore, the abovementioned optimization procedures are generally not adequate or not flexible enough.

In order to overcome drawbacks of these optimization procedures, a new set of nature inspired heuristic optimization algorithms were proposed. These techniques are resourceful, efficient and easy to use. These algorithms are part of Swarm Intelligence. They focus on animal behavior and insect conduct in order to develop some metaheuristics which can mimic their problem solution abilities, namely Ant Colony (AC) [21], Artificial Bee Colony (ABC) [22], Wasp Nets [22] and Particle Swarm Optimization (PSO) [23, 24].

This work deals with optimal analog circuit sizing using the PSO technique. In Sect. 2, we deal with the general formulation of the analog constrained optimization problem. In Sect. 3, we focus on the PSO heuristic for both mono and multiobjective problems. In Sect. 4, two application examples are given. The first application is a mono-objective problem that deals with optimizing the sizing of a low noise amplifier to meet fixed specifications. The second application is a multiobjective problem that consists of generating the trade off surface (Pareto front) linking two conflicting performances of a current conveyor. Conclusion remarks are given in Sect. 5.

2 The constrained problem formulation

Typically, an analog circuit designer formulates the requirements for the design of analog block in terms of bounds on performance functions. Analytical equations that predict these performance functions are then expressed in terms of device model parameters [25]. Next, these model parameters are replaced by their symbolic expressions at the design variables level. For instance, a simple MOSFET amplifier is depicted in Fig. 1. Expression 1 gives its voltage transfer function V_0/V_{in} :

$$\frac{V_0}{V_{in}} = - \frac{g_{mM1}(\vec{x})}{(C_{gsM2}(\vec{x}) + C_{gsM3}(\vec{x}))s + (g_{oM1}(\vec{x}) + g_{oM2}(\vec{x}) + g_{mM2}(\vec{x}))} \quad (1)$$

$$= - \frac{\sqrt{\frac{K_P I_0 W_1}{L_1}} L_1 L_2}{\lambda_N I_0 L_2 + \lambda_P I_0 L_1 + 2\sqrt{\frac{K_P I_0 W_2}{L_2}} L_1 L_2 + \frac{2}{3} C_{ox} L_1 L_2 (L_2 W_2 + L_3 W_3)}$$

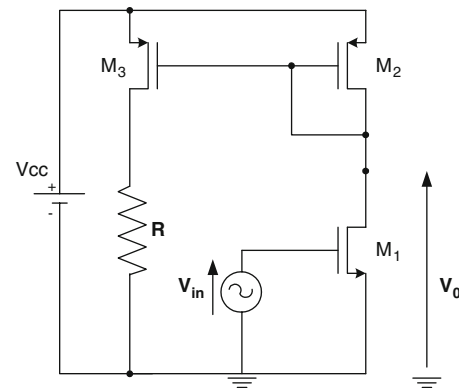


Fig. 1 A MOSFET amplifier

where g_{mMi} , g_{oMi} , C_{gsMi} , are transconductance, conductance and grid to source capacitance of the corresponding MOS transistor, respectively. I_0 is the bias current. W_i and L_i are width and length of the MOSFET's channel, respectively. C_{ox} , λ_N , λ_P , K_N and K_P are technology parameters. The variable vector \vec{x} represents bias voltages, bias currents and device sizes.

Optimal sizing of the circuit consists in finding variable set $\vec{x} = \{L_1, W_1, L_2, W_2, \dots\}$ that optimizes performance function(s) and meets imposed specifications, such as maximizing the amplifier's bandwidth while satisfying $(V_0/V_{in})_{dB} > G_{aindB}$. G_{aindB} is the gain low boundary that depends on the application.

Besides additional constraints have to be satisfied, such as voltage range constraints (MOS saturation conditions...), size ranges (minimum sizing allowed by the technology...), etc.

Thus, a general optimization problem can be defined in the following format:

$$\begin{cases} \text{Minimize } f(\vec{x}); & f(\vec{x}) \in R \\ \text{such that :} & \\ \vec{g}(\vec{x}) \leq 0; & \vec{g}(\vec{x}) \in R^m \\ \text{and } \vec{h}(\vec{x}) = 0; & \vec{h}(\vec{x}) \in R^n \\ \text{where } x_{Li} \leq x_i \leq x_{Ui}, & i \in [1, p] \end{cases} \quad (2)$$

m inequality constraints to satisfy; n equality constraints to assure; p parameters to manage; and \vec{x}_L and \vec{x}_U are lower and upper boundaries vectors of the parameters.

Actually, circuit optimization problems involve more than one objective. For instance, for the MOSFET amplifier, more than one objective function (OF) may be optimized, such as gain, noise figure, slewrate, bandwidth, etc. Consequently expression 2 should be modified as:

$$\left\{ \begin{array}{l} \text{Minimize } \vec{f}(\vec{x}); \quad \vec{f}(\vec{x}) \in R^k \\ \text{subject to :} \\ \vec{g}(\vec{x}) \leq 0; \quad \vec{g}(\vec{x}) \in R^m \\ \text{and } \vec{h}(\vec{x}) = 0; \quad \vec{h}(\vec{x}) \in R^n \\ \text{where } x_{Li} \leq x_i \leq x_{Ui}, \quad i \in [1, p] \end{array} \right. \quad (3)$$

where k number of objectives (≥ 2); m inequality constraints to satisfy; n equality constraints to assure; and p parameters to manage.

The flowchart depicted in Fig. 2 summarizes main steps of the sizing approach.

The goal of optimization is usually to minimize an objective function; the problem for maximizing $\vec{f}(\vec{x})$ can be transformed to minimizing $-\vec{f}(\vec{x})$. This goal is reached when the variables are located in the set of optimal solutions.

As it was introduced in section I, there exist many papers and books dealing with mathematical optimization methods and studying in particular their convergence properties (see for example [9, 11, 26]). These optimizing methods can be classified into two categories: deterministic methods and stochastic methods, known as heuristics.

Deterministic methods, such as simplex [27], branch and bound [28], goal programming [29], dynamic programming [30]... are effective only for problems of small size. They are not efficient when dealing with NP-hard and multi-criterion problems. In addition, it was proven that these optimization techniques impose several limitations due to their inherent solution mechanisms and their tight

dependence on the algorithm parameters. Besides they rely on the type of objective, the type of constraint functions, the number of variables and the size and the structure of the solution space. Moreover, they do not offer general solution strategies.

Most of the optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation. Therefore, classical optimization procedures are generally not adequate. Heuristics are necessary to solve large size problems and/or with many criteria [31]. They can be ‘easily’ modified and adapted to suit specific problem requirements, as it is illustrated in Fig. 3 [22]. Even though they don’t guarantee to find in an exact way the optimal solution(s), they give ‘good’ approximation of it (them) within an acceptable computing time. Heuristics can be divided into two classes: on the one hand, there are algorithms which are specific to a given problem and, on the other hand, there are generic algorithms, i.e., metaheuristics. Metaheuristics are classified into two categories: local search techniques, such as simulated annealing, tabu search ... and global search ones, like evolutionary techniques, swarm intelligence techniques ...

AC, ABC, PSO are swarm intelligence techniques, they form a subset of metaheuristics. Metaheuristics are inspired from nature and were proposed by researchers to overcome drawbacks of the aforementioned methods. In the following we focus on the use of the Particle Swarm Optimization technique for the optimal design of both mono-objective and multiobjective optimization problems [24], (<http://www.particleswarm.info/>).

3 The particle swarm optimization

The Particle Swarm Optimization (PSO) is a metaheuristic inspired from nature. This technique mimics some animal’s problem solution abilities, such as bird flocking or fish schooling. Interactions between animals contribute to the collective intelligence of the swarm. This approach was proposed in 1995 by Kennedy and Eberhart [23].

In PSO, multiple candidate solutions coexist and cooperate with each other simultaneously. Each solution, called a *particle*, ‘flies’ within the problem search space looking for the optimal position to land [22]. A particle, as time passes through its quest, adjusts its position according to its own ‘experience’ as well as the experience of its neighboring particles. Each particle remembers its best position and is informed of the best position reached by the swarm in the global version of the algorithm, or by the particle’s neighborhood in the local version of the algorithm.

PSO system combines local search method (through self experience) with global search methods (through

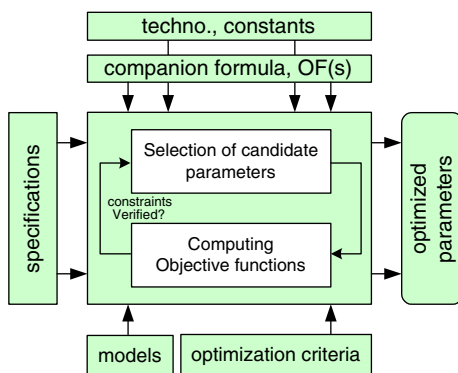
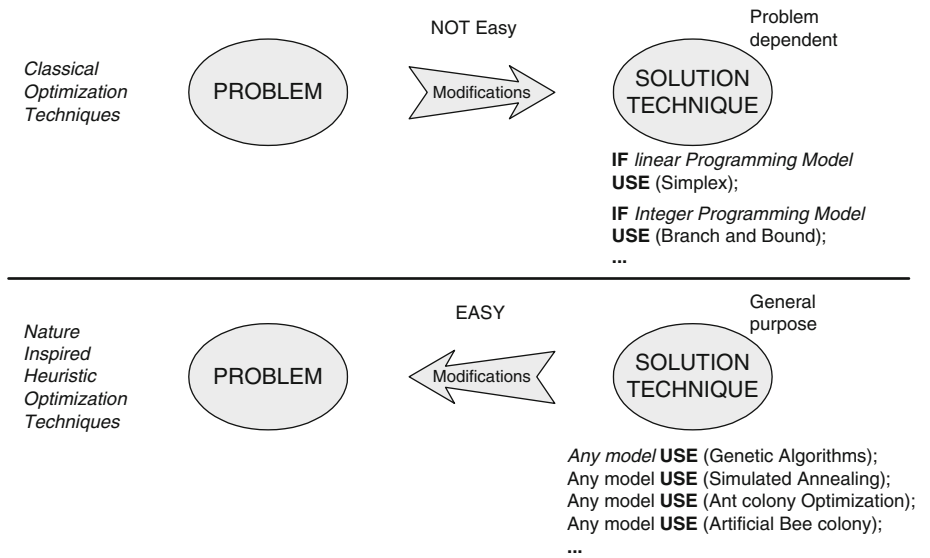


Fig. 2 Pictorial view of a design optimization approach

Fig. 3 A pictorial comparison of classical and modern heuristic optimization strategies (taken from [22])



neighboring experience) [22]. It is becoming popular due to the fact that the algorithm is relatively straightforward and is easy to be implemented [24]. Additionally there is plenty of source codes of PSO available in public domain (see for example <http://www.swarmintelligence.org/codes.php>).

When compared to the other well known heuristics, such as GA (NSGA II for multiobjective problems) and SA, PSO presents the advantage of the rapid convergence to the promising regions. On the other hand, PSO can be trapped into a local minimum. This phenomenon highly relies on the global search constituent. We refer the reader to [24] for further details on such characteristics.

3.1 Mono-objective PSO

PSO has been proposed for solving mono-objective problems and has been successfully used for both continuous non linear and discrete mono-objective optimization. In short, the approach consists of the following [24].

The algorithm starts with a random initialization of a swarm of particles in the multi-dimensional search space, where a particle represents a potential solution of the problem being solved. Each particle is modeled by its position in the search space and its velocity. At each time step, all particles adjust their positions and velocities, i.e., directions in which particles need to move in order to improve their current position, thus their trajectories.

This is done according to their best locations and the location of the best particle of the swarm in the global version of the algorithm, or of the neighbors in the local version. Figure 4 illustrates this principle.

Changes to the position of the particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals [32].

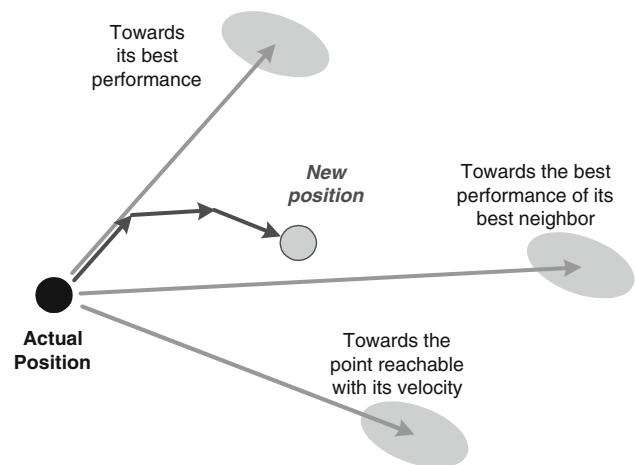


Fig. 4 Principle of the movement of a particle

Thus each particle tends to be influenced by the success of any other particle it is connected to.

Expressions of a particle position at time step t and its velocity are as follows [22, 33]:

$$\vec{v}_i(t) = \underbrace{\omega \vec{v}_i(t-1)}_{\text{Inertia}} + \underbrace{c_1 r_1 (\vec{x}_{pbest_i} - \vec{x}_i(t))}_{\text{Personal Influence}} + \underbrace{c_2 r_2 (\vec{x}_{leader_i} - \vec{x}_i(t))}_{\text{Social Influence}} \tag{4}$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \tag{5}$$

where x_{pbest_i} is the best personal position of a given particle, x_{leader_i} refers to the best position reached by the particle's neighborhood, ω is an inertia weight that controls the diversification feature of the algorithm, c_1 and c_2 control the attitude of the particle of searching around its best location and the influence of the swarm on the particle's behavior, respectively (c_1 and c_2 control the intensification

feature of the algorithm). r_1 and r_2 are random values uniformly sampled in $[0,1]$ for each dimension.

Figure 5 illustrates the flowchart of the mono-objective PSO. N is the swarm's size. $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,N}]$ denotes the best location reached by the i th particle at time t . $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]$ is the i th particle position and $g = [g_1, g_2, \dots, g_N]$ represents the best location reached by the entire swarm.

An application example of performance maximization in terms of voltage gain of a Low Noise Amplifier is given in section IV.

3.2 Multiobjective PSO

As it was introduced in section II, multiobjective optimization is a common task in circuit design. Commonly designers transform the multiobjective problem into a mono-objective one using the aggregation approach [34]. The latter requires the designer to select values of weights for each objective. The so obtained mono-objective function can be written as:

$$F(\vec{x}) = \sum_{i=1}^k \omega_i f_i(\vec{x}) \tag{6}$$

$\omega_i, i \in [1, k]$, are weighting coefficients

For instance, in [35] a normalization method was used to guide the choice of the weighting coefficients. Yet, this

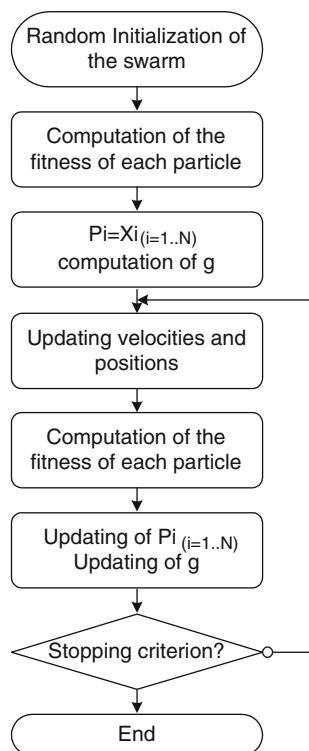


Fig. 5 Flowchart of the mono-objective PSO

approach is not suitable in most problems since it is not able to detect solutions when the solution space is not convex. Furthermore, it was shown in [36] that varying the weighting coefficients ω_i cannot allow sweeping the entire Pareto front and cannot also ensure a homogeneous distribution of the design over the Pareto front. Besides, obtained solutions may be very sensitive to a small variation of these coefficients [36].

Even though some approaches, such as geometric programming, have overcome this main drawback by converting the non convex problem into a convex one [37], using the weighting approach, only a unique “optimal” point can be obtained. This solution depends on the choice of the weighting coefficients.

On the other hand, a multiobjective approach (MO) can be stated as finding vectors $\vec{x} = (x_1, x_2, \dots, x_n)$ that satisfy the constraints and optimize the vector function $\vec{f}(\vec{x})$ [38]. However, the objective functions may be in conflict. Thus, the goal of the MO algorithm is to provide a set of Pareto optimal solutions of the aforementioned problem [26, 38–40]. It is to be noticed that a solution \vec{x} of a MO problem is said Pareto optimal if and only if there does not exist another solution \vec{y} such that $\vec{f}(\vec{y})$ dominates $\vec{f}(\vec{x})$, i.e., no component of $\vec{f}(\vec{x})$ is smaller than the corresponding component of $\vec{f}(\vec{y})$ and at least one component is greater [41]. Without loss of generality, we consider the minimization case. Figure 6 depicts the multiobjective approach.

PSO was adapted to be able to deal with multiobjective optimization problems. The idea consists of the use of an external memory: an archive, in which each particle deposits its ‘flight’ experience at each running cycle. More precisely, the aim of the archive is to store all the non-dominated solutions found during the optimization process. At the end of the execution, all the positions stored in the archive give us an approximation of the theoretical Pareto Front.

Algorithm 1 Archive’s update

For $i = 1 \dots N$

If $particle[i]$ dominates some elements of the archive

Delete dominated elements

End If

If $particle[i]$ is nondominated

If the archive is not full

Add $particle[i]$ to the archive

Else

Compute the crowding distances

Sort the element of the archive by crowding distance

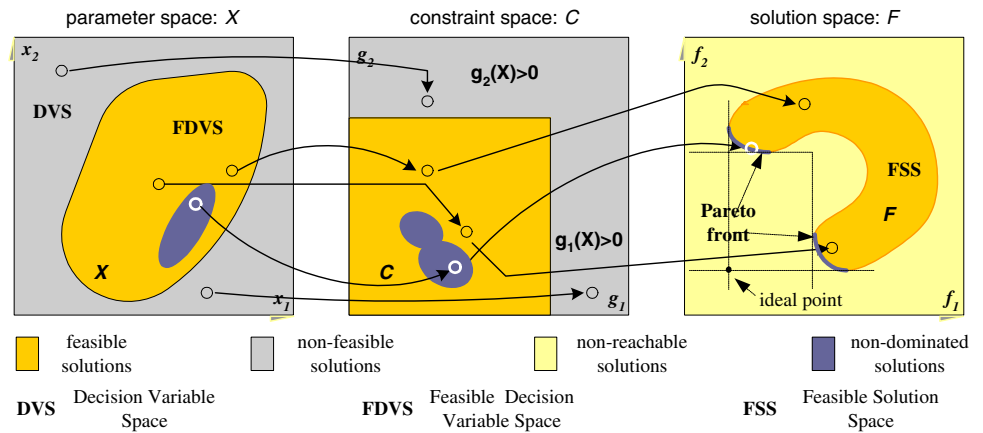
Replace the particle with the lower crowding distance by $particle[i]$

End If

End If

End For

Fig. 6 Illustration of a general multiobjective optimization problem



In order to avoid excessive growing of this memory, its size is fixed. This implies to define some rules for the update of the archive. Algorithm 1 depicts the pseudo-code of these rules.

The crowding distance of the i th element of the archive estimates the size of the largest cuboid enclosing the point i without including any other point in the archive. The idea is to maximize the crowding distance of the particles in order to obtain a Pareto front as uniformly spread as possible. In Fig. 7, the crowding distance of the i th solution of the archive (black spots) is the average side length of the cuboid shown by the dashed box. The flowchart presented in Fig. 8 depicts the MO-PSO [41, 42].

In the following section we give two application examples where mono-objective and multiobjective PSO are considered.

4 Application examples

4.1 A mono-objective problem:

The problem consists of optimally sizing a common source low noise amplifier (LNA) with an inductive degeneration [43] presented at Fig. 9, for the UMTS standard.

The problem is defined in the following way.

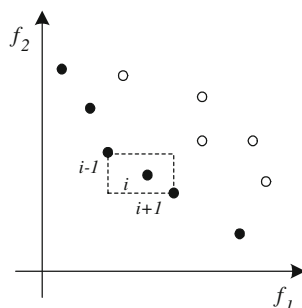


Fig. 7 Crowding distance

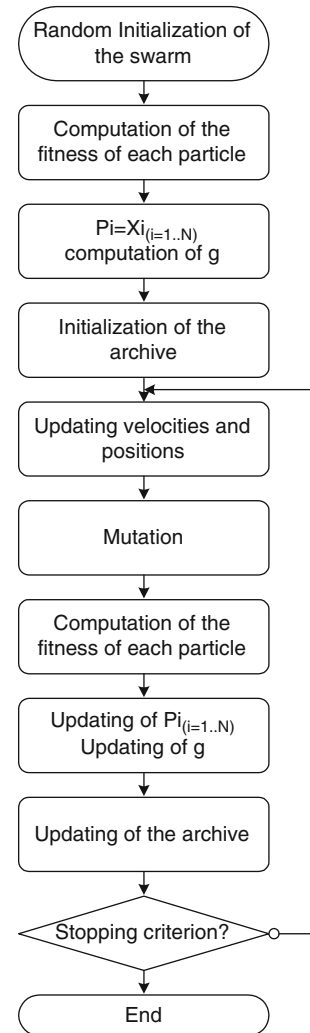


Fig. 8 Flowchart of a MO-PSO

4.1.1 Constraints

- Ensure input and output matching of the LNA to guarantee a maximum power transfer from the antenna

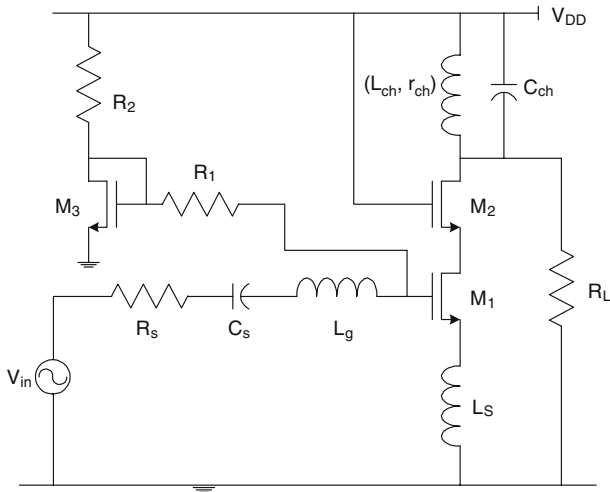


Fig. 9 A CMOS LNA with biasing network

to the LNA then to the pre-amplifier. This leads to the equality constraints 7 and 8:

$$L_s = \frac{R_s C_{gs}}{g_m} \tag{7}$$

$$L_g = \frac{1}{\omega_0^2 C_{gs}} - L_s \tag{8}$$

where C_{gs} and g_m are the grid to source capacitance and the transconductance of the MOS transistor M1, respectively. ω_0 is the pulsation of the UMTS standard, $\omega_0 = 2\pi f_0$, with $f_0 = 2.14$ GHz.

- Linearity, which is highly dependent on the saturation of the MOS transistors. Thus inequalities 9 and 10 must be satisfied. These conditions were determined by fixing the 1 dB compression point value (CP1=0 dBm).

$$V_{DS1\min} > V_{DSsat1} \tag{9}$$

$$V_{DS2\min} > V_{DSsat2} \tag{10}$$

where $V_{DSi\min}$ and V_{DSsati} are the minimum drain to source voltages and the saturation voltages of transistors M1 and M2, respectively.

- Noise Figure: the reduced NF expression proposed in [44] was used. The constraint is:

$$NF < (NF_{\max} = 1dB) \tag{11}$$

- Transit frequency: f_T must be five times higher than the UMTS frequency.

4.1.2 Objective function

- Gain: its symbolic expression was computed using the small-signal equivalent circuit and the objective function is that the gain has to present its maximum value at the UMTS frequency. The gain's expression is not given due to its large number of terms.

Table 1 Parameters of the mono-objective algorithm

Size of the swarm	Number of iterations	c_1	c_2	ω
20	10,000	$0.5 + \log(2)$	$0.5 + \log(2)$	$1/(2 \log(2))$

Table 2 Optimal parameters' values

C_c (pF)	L_g (nH)	L_s (nH)	L_{ch} (nH)
10	8.110	0.320	0.715
C_{ch} (pF)	r_{ch} (Ω)	R_1 (k Ω)	R_2 (k Ω)
7.44	1.92	1	1.03
$W_1(\mu\text{m})/L_1(\mu\text{m})$	$W_2(\mu\text{m})/L_2(\mu\text{m})$	$W_3(\mu\text{m})/L_3(\mu\text{m})$	
640/0.35	630/0.35	40/0.35	

Table 3 Simulation conditions

Technology	AMS 0.35 μm
Supply voltage (V_{DD})	Single 2.5 V
LNA bias current (I_d)	29 mA

Table 4 Reached performances (Gain at umts frequency)

Theoretical (Maple software)	17.46 dB
Simulation (ADS software)	13.35 dB

This optimizing problem was solved using mono-objective PSO technique. The algorithm was programmed using C++ software. Notice that, boundaries of parameters were imposed by the used technology. PSO algorithm parameters are given in Table 1 [33]. Optimal obtained parameters are presented in Table 2. We notice that 25 runs were performed, and the average computing time equals 1.57 s.¹ Simulation conditions and reached performances, obtained using ADS software (Agilent Advanced Design System), are given in Tables 3 and 4, respectively.

ADS simulation results are depicted in the following figures; they show the good agreement between ADS and expected results. Figure 10(a) and (b) prove that input and output matching are insured (50 Ω). Figure 11 shows the LNA performs a compression point CP1 of -2 dBm. Figure 12(a) and (b) illustrate LNA noise figure and voltage gain that equal 0.91 dB, and 13.35 dB, respectively.

4.2 A multiobjective problem

The problem consists of optimizing performances of a second generation current conveyor (CCII) [45, 46] regarding to its main influencing performances. The aim

¹ a (2 GHz, 2 Go RAM) core 2 DUO PC was used for this purpose.

Fig. 10 Input (a) and output (b) matching (scattering parameters S(1,1) and S(2,2))

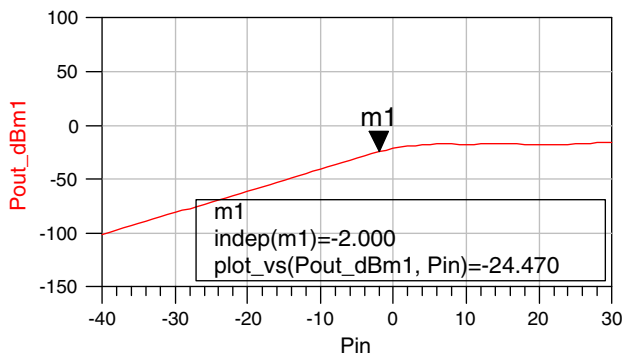
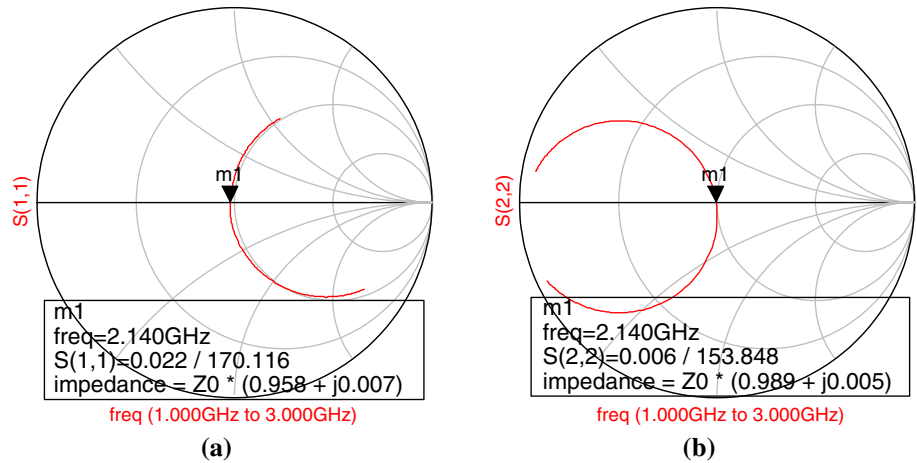


Fig. 11 The LNA compression point (CP1)

consists of maximizing the conveyor high current cutoff frequency and minimizing its parasitic X-port resistance [47]. Figure 13 illustrates the conventional positive second generation current conveyor.

4.2.1 Constraints

- Transistor saturation conditions: all the CCII transistors must operate in the saturation mode. Saturation constraints of each MOSFET were determined. For instance, expression 12 gives constraints on M2 and M8 transistors:

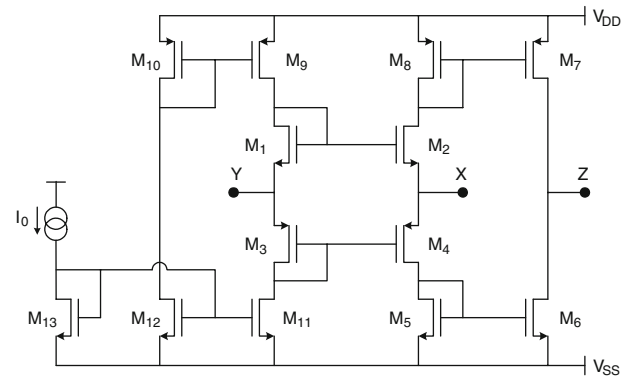


Fig. 13 The second generation current conveyor

$$\frac{V_{DD}}{2} - V_{TN} - \sqrt{\frac{I_0}{K_P W_2/L_2}} \geq \sqrt{\frac{I_0}{K_N W_8/L_8}} \quad (12)$$

where I_0 is the bias current, W_i/L_i are the aspect ratio of the i th MOS transistor. K_N , K_P and V_{TN} are technology parameters. V_{DD} is the DC voltage power supply.

4.2.2 Objective functions

- R_X : the value of the X-port input parasitic resistance has to be minimized.

Fig. 12 The noise figure (a) and the voltage gain (b)

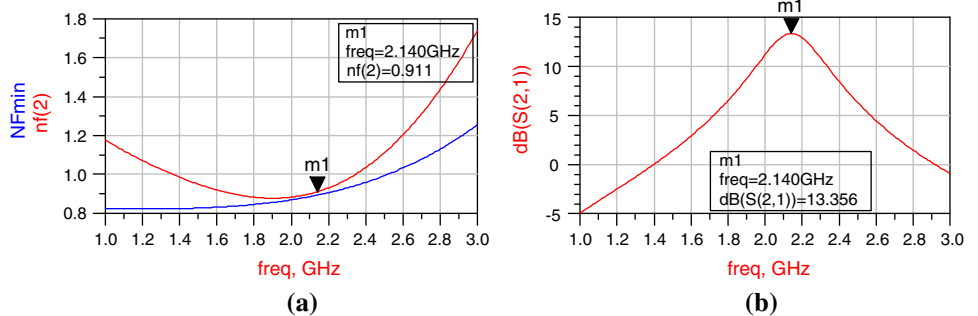


Table 5 Parameters of the multiobjective algorithm

Size of the swarm	Number of iterations	c_1	c_2	ω
20	10,000	1	1	0.4

Table 6 Optimal parameters' values

	L_N (μm)	W_N (μm)	L_P (μm)	W_P (μm)
Lower edge of the Pareto front	0.35	30.00	0.57	20.07
Higher edge of the Pareto front	0.35	7.28	0.55	12.60

Table 7 Simulation conditions

Technology	AMS 0.35 μm
Supply voltage (V_{SS}/V_{DD})	-2.5 V/+2.5 V
CCH bias current (I_0)	100 μA

- f_{chi} : the high current cut off frequency has to be maximized.

Symbolic expressions of these functions are not given due to their large number of terms.

Table 5 gives PSO algorithm parameters. PSO optimal parameter values and simulation conditions are shown in Tables 6 and 7, respectively, where L_N , W_N , L_P and W_P refer to length and width of NMOS and PMOS transistors, respectively. The average time taken for 25 runs, under the same conditions as in the mono-objective case, equals 2.32 s Fig. 14 shows both the parameter space and the solution space (Pareto front) and the correspondence between each parameter value and its solution. Notice that additional degrees of freedom are given to the designer to choose his 'optimal' parameter vector within the set of the

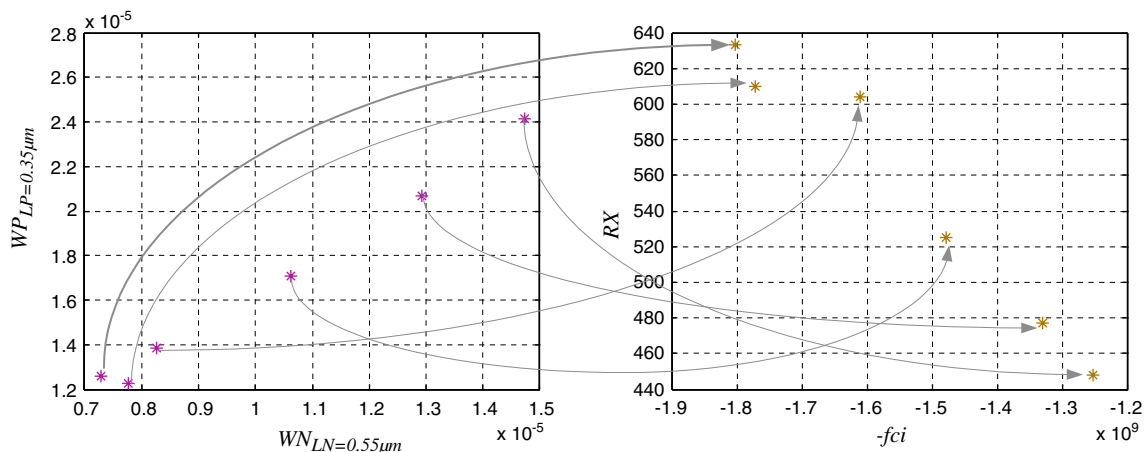


Fig. 14 Illustration of the correspondence between parameter and solution spaces

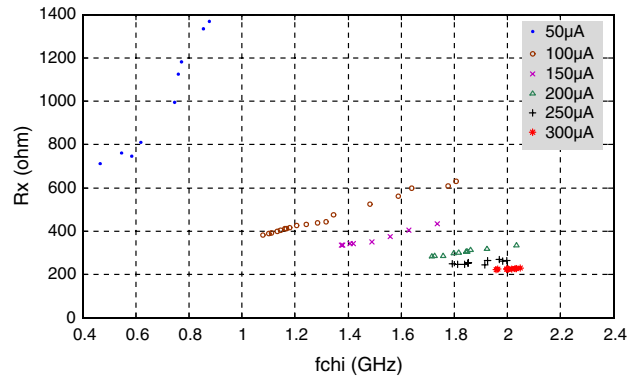


Fig. 15 Pareto fronts obtained for various values of the bias current I_0

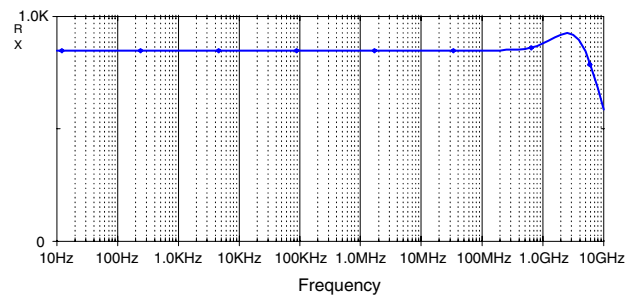


Fig. 16 SPICE simulation results for R_X : solution giving maximum f_{chi}

non-dominated solutions. For instance, an additional criterion can be the occupied area. Figure 15 illustrates the solution space (R_X vs. f_{chi}) for different values of the bias current I_0 . Finally, Figs 16, 17, 18, 19 show SPICE simulation results performed for two particular solutions, i.e., edges of the Pareto boarder (solutions giving the maximum f_{chi} and the minimum R_X , respectively). These results are in good agreement with the expected theoretical ones. Table 8 summarizes and compares reached performances.

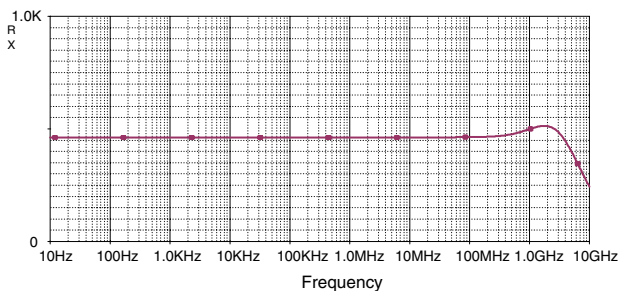


Fig. 17 SPICE simulation results for R_X : solution giving minimum R_X

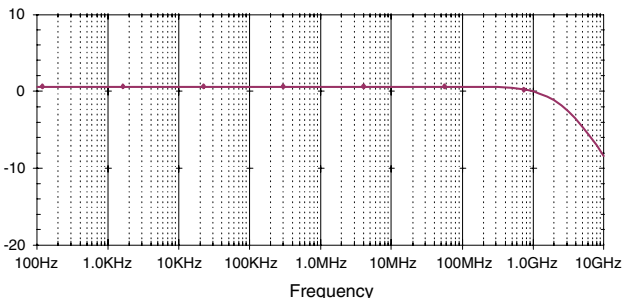


Fig. 18 SPICE simulation results for f_{chi} : solution giving maximum f_{chi}

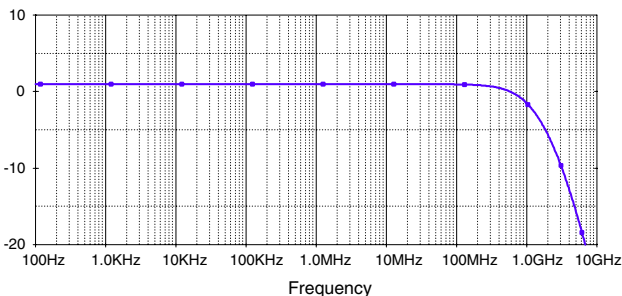


Fig. 19 SPICE simulation results for f_{chi} : solution giving minimum R_X

Finally, it is to be highlighted that this CCII was optimized in [48], but using a weighting approach. The unique ‘optimal’ solution given in [48] belongs to the Pareto front and it corresponds to its high boundary. Besides, and for comparison reasons, NSGA II algorithm (<http://www.mathworks.com/matlabcentral/fileexchange/10351>) was used to

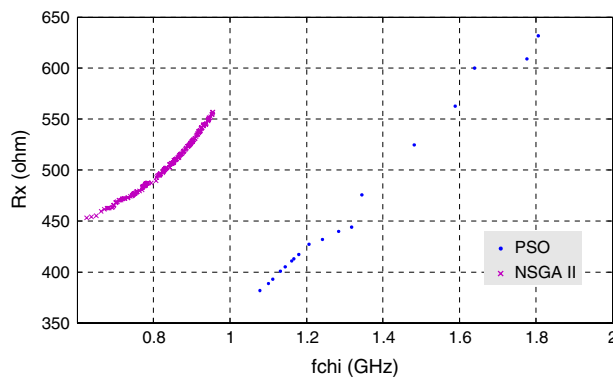


Fig. 20 PSO front versus NSGA II front

generate the Pareto front (R_X, f_{chi}). Figure 20 presents fronts obtained using PSO and NSGA II, for $I_0 = 100\mu A$. Results obtained using PSO are much better than those of NSGA II. This is mainly due to the fact that PSO handles constraints better than its counterpart.

5 Conclusion

This paper details the particle swarm optimization (PSO) technique for the optimal design of analog circuits. Reached results can be summarized as follows:

- We show the practical applicability of PSO to optimize performances of analog circuits and its suitability for solving both a mono-objective optimization problem and a multiobjective one. Two applications were presented. The first deals with maximizing the voltage gain of a low noise amplifier: a mono-objective optimization problem. The second is a bi-objective one. It consists in computing the Pareto tradeoff surface in the space solutions: parasitic input resistance vs. high current cutoff frequency.
- According to the simulation results, severe parameter tuning is not required. Indeed, PSO only requires less than 10,000 iterations for obtaining “optimal” solutions, even for large-scale systems.

Our current work focuses on integrating PSO approach in an automated design flow.

Table 8 Reached performances (points located at the pareto front edges)

	Theoretical (MATLAB software)		Simulation (SPICE software)	
	f_{chi} (GHz)	R_X (Ω)	f_{chi} (GHz)	R_X (Ω)
Lower edge of the Pareto front	1.078	382	1.122	462
Higher edge of the Pareto front	1.802	633	1.620	848

References

- Gielen, G., & Sansen, W. (1991). *Symbolic analysis for automated design of analog integrated circuits*. Dordrecht: Kluwer Academic Publishers.
- Toumazou, C., Lidgley, F. J., & Haigh, D. G. (1993). *Analog integrated circuits: The current mode approach*, IEEE circuit and systems series 2.
- Graeb, H., Zizala, S., Eckmueller, J., & Antreich, K. (2001). *The sizing rules method for analog integrated circuit design*. IEEE/ACM International conference on computer-aided design, ICCAD'01. November 4–8, 2001. San Jose, CA, USA.
- Conn, A. R., Coulman, P. K., Haring, R. A. Morrill, G. L., & Visweswariah, C. (1996). *Optimization of custom MOS circuits by transistor sizing*. The international conference on computer aided design, ICCAD'96. November 10–14, 1996. San Jose, CA, USA.
- Medeiro, F., Rodríguez-Macías, R., Fernández, F. V., Domínguez-Astro, R., Huertas, J. L., & Rodríguez-Vázquez, A. (1994). Global design of analog cells using statistical optimization techniques. *Analog integrated circuits and signal processing*, 6(3), 179–195.
- Silveira, F., Flandre, D., & Jespers, P. G. A. (1996). A gm/Id based methodology for the design of CMOS analog circuits and its application to the synthesis of a SOI micropower OTA. *IEEE Journal of solid state circuits*, 31(9), 1314–1319.
- O'Connor, I., & Kaiser, A. (2000). Automated synthesis of current memory cells. *IEEE Transactions on computer aided design of integrated circuits and systems*, 19(4), 413–424.
- Loulou, M., Ait Ali, S., Fakhfakh, M., & Masmoudi, N. (2002). *An optimized methodology to design CMOS operational amplifier*. IEEE International conference on microelectronic, ICM'2002, December 14–16, 2002. Beirut, Lebanon.
- Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8, 541–564.
- Aarts, E., & Lenstra, K. (2003). *Local search in combinatorial optimization*. Princeton: Princeton University Press.
- Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for hard optimization: Methods and case studies*, New York: Springer.
- Siarry, P., Berthiau, G., Durbin, F., & Haussy, J. (1997). Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Transactions on Mathematical Software*, 23, 209–228.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Journal of science*, 220, 671–680.
- Courat, J. P., Raynaud, G., Mrad, I., & Siarry, P. (1994). Electronic component model minimization based on log simulated annealing. *IEEE Transactions on circuits and systems*, 41, 790–795.
- Glover, F. (1989). Tabu search-part I. *ORSA Journal on computing*, 1(3), 190–206.
- Glover, F. (1990). Tabu search-part II. *ORSA Journal on computing*, 2(1), 4–32.
- Laguna, M., & Martí, R. (2003). *Scatter search: Methodology and implementation in C*, (Vol. 24). Series Operations Research/Computer Science Interfaces Series. New York: Springer.
- Grimbleby, J. B. (2000). Automatic analogue circuit synthesis using genetic algorithms. *IEE Proceedings-Circuits, Devices and Systems*, 147(6), 319–323.
- Dinger, R. H. (1998). *Engineering design optimization with genetic algorithm*. IEEE Northcon conference. October 21–23, 1998. Seattle WA, USA.
- Marseguerra, M., & Zio, E. (2000). *System design optimization by genetic algorithms*, *The IEEE annual reliability and maintainability symposium*, January 24–27, 2000. Los Angeles, California, USA.
- Dorigo, M., DiCaro, G., & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life Journal*, 5, 137–172.
- Chan, F. T. S., & Tiwari, M. K. (2007). *Swarm Intelligence: focus on ant and particle swarm optimization*. Vienna: I-Tech Education and Publishing.
- Kennedy, J., & Eberhart, R. C. (1995). *Particle swarm optimization*. The IEEE International conference on neural networks, November 27–December 1, 1995. WA, Australia.
- Clerc, M. (2006). *Particle swarm optimization*, International scientific and technical encyclopaedia.
- Maulik, P. C., & Carley, R. (1991). *Automating analog circuit design using constrained optimization techniques*. The IEEE international conference on computer-aided design. ICCAD'91. November 11–14, 1991. Santa Clara, CA, USA.
- Siarry, P., & Michalewicz, Z. (2007). *Advances in metaheuristics for hard optimization*. New York: Springer.
- Nelder, J. A., & Mead, R. (1965). A simplex method for function optimization. *Computer Journal*, 7, 308–313.
- Doig, A. G., & Land, A. H. (1960). An automatic method for solving discrete programming problem. *Econometrica*, 28, 497.
- Scniederjans, M. J. (1995). *Goal programming methodology and applications*. Dordrecht: Kluwer Publishers.
- Bellman, R. E. (2003). *Dynamic programming*. New York: Dover publication.
- Basseur, M., Talbi, E. G., Nebro, A., & Alba, E. (2006). Metaheuristics for multiobjective combinatorial optimization problems: review and recent issues, Report no. 5978. National institute of research in informatics and control (INRIA), September 2006.
- Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., & Nakanish, Y. (2001). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *The IEEE Transactions on Power Systems*, 15(4), 1232–1239.
- Clerc, M., & Kennedy, J. (2002). The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. *The IEEE Transactions on evolutionary computation*, 6, 58–73.
- Reyes-Sierra, M., & Coello-Coello, C. A. (2006). Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3), 287–308.
- Fakhfakh, M., Loulou, M., & Masmoudi, N. (2007). Optimizing performances of switched current memory cells through a heuristic. *Analog Integrated Circuits and Signal Processing*, 50(2), 115–126.
- Fakhfakh, M. (2009). A novel alienor-based heuristic for the optimal design of analog circuits. *Microelectronics Journal*, 40(1), 141–148.
- del Mar-Hershenson, M., Boyd, S., & Lee, T. H. (2001). CMOS operational amplifier design and optimization via geometric programming. *The IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(1), 1–21.
- Zitzler, E., Laumanns, M., & Bleuler, S. (2004). A Tutorial on evolutionary multiobjective optimization. In X. Gandibleux, M. Sevaux, K. Sörensen & V. T'kindt (Eds.), *Metaheuristics for multiobjective optimization*. Lecture notes in economics and mathematical systems (Vol. 535, pp. 3–37). New York: Springer.
- Collette, Y., & Siarry, P. (2003). *Multiobjective optimization—principles and case studies*. New York: Springer.
- Bayart, B., Kotlicki, P., & Nowacki, M. (2002). Multiobjective optimization, Collection of reports for topics of evolutionary computation, Denmark.

41. Raquel, C. R., & Naval, P. C. (2005). *An effective use of distance in multiobjective particle swarm optimization*. The genetic and evolutionary computation conference, GECCO'2005, June 25–29, 2005. Washington, DC, USA.
42. Coello-Coello, C. A., & Lechuga, M. S. (2002). *MOPSO: A proposal for multiple objective particle swarm optimization*, *The evolutionary computation congress*. May 12–17, 2002. Hawaii.
43. Razavi, B. (1998). *RF Microelectronics*. Englewood cliffs: Prentice Hall press.
44. Andreani, P., & Sjolund, H. (2001). Noise optimization of an inductively degenerated CMOS low noise amplifier. *The IEEE Transactions on Circuits and Systems II*, 48(9), 835–841.
45. Smith, K. C., & Sedra, A. (1968). The current conveyor—a new circuit building block. *Proceeding of the IEEE*, 56(8), 1368–1369.
46. Sedra, A., & Smith, K. C. (1970). A second generation current conveyor and its applications. *IEEE Transactions on Circuit Theory*, 17(1), 132–134.
47. Cooren, Y., Fakhfakh, M., Loulou, M., & Siarry, P. (2007). *Optimizing second generation current conveyors using particle swarm optimization*. The IEEE International conference on microelectronics, ICM'2007, December 29–31, 2007. Cairo, Egypt.
48. Ben Salem, S., Fakhfakh, M., Masmoudi, D. S., Loulou, M., Loumeau, P., & Masmoudi, N. (2006). A High Performances CMOS CCI and High Frequency Applications. *Analog Integrated Circuits and Signal Processing*, 49(1), 71–78.



Mourad Fakhfakh was born in Sfax-Tunisia in 1969. He received the engineering and the PhD degrees from the national engineering school of Sfax Tunisia in 1996 and 2006, respectively. From 1998 to 2004 he worked in the Tunisian National Society of Electricity and Gas (STEG) as the head of the technical intervention department. In September 2004, he joined the higher institute of electronics and communications (ISECS) where he is working as

an assistant professor. Since 2002 he has been with the electronics and information technology laboratory (LETI/ENIS) where he is currently a researcher. His research interests include symbolic analysis techniques, analog design automation, and optimization techniques.



Yann Cooren was born in Bordeaux, France in 1982. He received his engineering degree and his master from the École Nationale Supérieure d'Électronique, d'Informatique et de Radiocommunications de Bordeaux (ENSEIRB). In 2008, he obtained his PhD from the Laboratoire d'Images, Signaux et Systèmes Intelligents (LiSSi) of the University of Paris 12. His main research interests are metaheuristics and, particularly, the Particle Swarm Optimization method.



Amin Sallem was born in Sfax, Tunisia in 1981. He received the electrical engineering diploma and the master degree in electronics from the National Engineering School of Sfax (ENIS) in 2004 and 2008, respectively. Since 2004, he is a PhD student at the Electronic and Information Technology Laboratory of Sfax (LETI/ENIS) In February 2008, he joined the Higher Institute for Technological Studies of Sousse (ISET) where he is working as an assistant.

His current research interests are on analogue CMOS integrated circuits design and on optimization techniques.



Mourad Loulou was born in Sfax, Tunisia in 1968. He received the degree of Diplôme d'Ingénieur de l'École Nationale d'Ingénieurs de Sfax in 1993. He received his PhD degree in 1998 in electronics system design from the University of Bordeaux France. He joined the electronic and information technology laboratory of Sfax "LETI" since 1998 and he has been assistant Professor at Ecole nationale d'Ingénieurs de Sfax from 1999. Since 2004 he

has been an associate Professor at the same institution. Actually he supervises the Analogue and Mixed Mode Design Group of LETI Laboratory. His current research interests are on analogue, mixed and RF CMOS integrated circuits design and automation.



Patrick Siarry was born in France in 1952. He received the PhD degree from the University Paris 6, in 1986 and the Doctorat of Sciences (Habilitation) from the University Paris 11, in 1994. He was first involved in the development of analog and digital models of nuclear power plants at Electricité de France (E.D.F.). Since 1995 he is a professor in automatics and informatics. His main research interests are computer-aided design of electronic circuits, and

the applications of new stochastic global optimization heuristics to various engineering fields. He is also interested in the fitting of process models to experimental data, the learning of fuzzy rule bases, and of neural networks.