Chapter 7
Analog Circuit Design Based on Robust POFs Using an Enhanced MOEA with SVM Models

Nuno Lourenço, Ricardo Martins, Manuel Barros, and Nuno Horta

Abstract. In this chapter, a multi-objective design methodology for automatic analog integrated circuits (IC) synthesis, which enhances the robustness of the solution by varying technological and environmental parameters, is presented. The automatic analog IC sizing tool GENOM-POF was implemented and used to demonstrate the methodology, and to verify the effect of corner cases on the Pareto optimal front (POF). To enhance the efficiency of the tool, a supervised learning strategy, which is based on Support Vector Machines (SVM), is used to create feasibility models that efficiently prune the design search space during the optimization process, thus, reducing the overall number of required evaluations. The GPOF-SVM optimization kernel consists of a modified version of the multi-objective evolutionary algorithm (MOEA), NSGA-II, and uses HSPICE® as the evaluation engine. The usage of standard inputs and outputs eases the integration with other design automation tools, either at system level or at physical level, which is the case of LAYGEN, an in-house layout generation tool. Finally, the approach was validated using benchmark examples, which consist of circuits tested with similar tools, particularly, the former GENOM tool and other tools from literature.

7.1 Introduction

In the last decades, Very Large Scale Integration (VLSI) technologies have been widely improved, allowing the proliferation of consumer electronics and enabling the growth of IC market from $10 billion in 1980 to over than $300 billion in 2013

Nuno Lourenço · Ricardo Martins · Nuno Horta
Instituto de Telecomunicações, Lisboa, Portugal, and Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal
e-mail: nuno.horta@lx.it.pt

Manuel Barros
Instituto de Telecomunicações, Lisboa, Portugal, and Instituto Politécnico de Tomar, Tomar, Portugal

M. Fakhfakh et al. (Eds.): *Analog/RF & Mixed-Signal Circuit Sys. Design*, LNEE 233, pp. 149–167.
DOI: 10.1007/978-3-642-36329-0_7

(according to IC Insights Inc) [1]. IC designers are building systems that are increasingly more complex and integrated. In the System on Chip (SoC) age is common to find devices where the whole system is integrated in a single chip [2].

The need of new functionalities, longer battery times, smaller (thinner) devices, more power efficiency, less production and integration costs and less design cost, makes the design of electronic systems a truly challenging task. The complexity of electronic systems design and the strict time-to-market impose the use of Computer Aided Design (CAD) tools to support the design process. In digital IC design, mature Electronic Design Automation (EDA) tools and design methodologies are available helping the designers to keep up with the new capabilities offered by the technology. Currently almost all low-level phases of the process are automated. The level of automation is far from the push-button stage, but is keeping up reasonably well with the complexity supported by the technology. On the other hand, analog IC design automation tools strive to keep up with the new challenges created by technological evolution [3,4]. Due to the lack of automation, designers keep exploring the solution space almost manually. This method causes long design times, and allied to the non-reusable nature of analog IC, makes analog IC design a cumbersome task.

In this work a multi-objective design methodology and tool for automatic analog IC synthesis, GPOF-SVM, is presented. GPOF-SVM stems from GENOM [5-7] and GENOM-POF [8], the first is a former single objective optimizer enhanced by an SVM feasibility model and the second is a multi-objective circuit optimizer. This chapter is organized as follows: in section 7.2 an overview of related work in analog IC design automation at circuit/system-level sizing is presented; section 7.3 explains the architecture of GPOF-SVM; section 7.4 presents case studies; and finally, in section 7.5 some conclusions are drawn and future work proposed.

7.2 Related Work

Historically, the tools for automated circuit sizing are classified as knowledge-based or optimization based. This classification, illustrated in Fig. 7.1, is based on the fundamental techniques used to address the problem.

Early strategies, like IDAC [9] and BLADES [10], tried to systematize the design by using a design plan derived from expert knowledge. In these methods, a pre-designed plan is built with design equations and a design strategy that produce component sizes that meet the performance requirements. The knowledge-based approach was applied with moderate success to automatic analog IC sizing. The main advantage of this approach is the short execution time. On the other hand, deriving the design plan is hard and time-consuming, the design plan requires constant maintenance in order to keep it up to date with technological evolution, and the results are not optimal, suitable only as a first-cut-design.

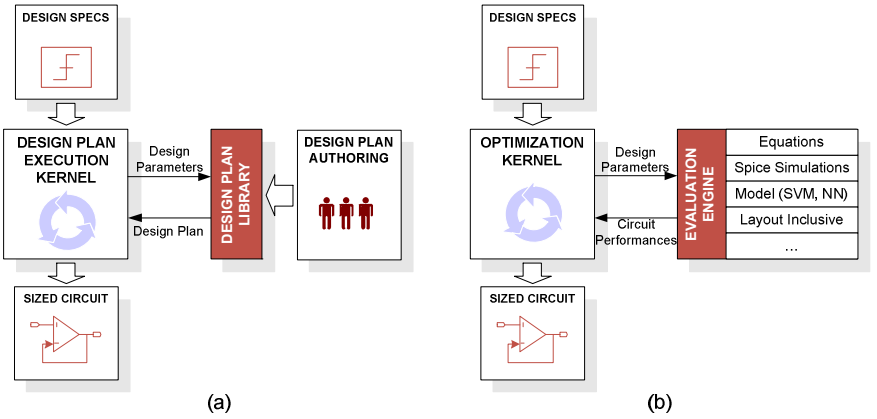

Fig. 7.1 Automatic specification translation approaches: (a) knowledge-based and (b) optimization-based

Aiming for optimality, the next generations of sizing tools apply optimization techniques to analog IC sizing. Based on the evaluation techniques employed, the optimization-based sizing tools can be further classified into three main sub-classes: equation-based, electrical-simulation-based, and numerical-model-based.

- *Equation:* These methods use analytic design equations to evaluate the circuit performance. The strong point of equation-based methods like GPCAD [11], Kuo-Hsuan et. *al.* [12] among others is the short evaluation time, making them, like the knowledge-based approaches, extremely suited to derive first-cut designs. The main drawbacks are: not all design characteristics can be easily mapped by analytic equations and the approximations introduced in the equations yield low accuracy designs. To reduce the long time dispended in model development, automatic techniques were proposed (Gielen et al. in [13] pro-vide a good overview on symbolic analysis applied to analog ICs).
- *Electrical Simulation:* These sizing techniques use a circuit simulator to evaluate the circuit's performance. The strong points of this approach are generality and easy-and-accurate model, however, typified by long execution time. To cope with this limitation Kuo-Hsuan et. *al.*[12] used equations to derive an approximate initial solution, Cheng et *al.* [14] solving the bias of the transistors first, the transistor sizes are then derived from the bias point using electric simulation. In MAELSTROM and ANACONDA [15] the evaluation is done using a parallel mechanism that shares the evaluation load among multiple computers.
- *Numerical Model:* The numerical-model-based tools like Alpaydin et. *al.* [16] and Barros et. *al.* [6] use macro models, e.g., neural-networks or support vector machines, to speed up the evaluation of the circuit's performance, reducing the high execution times caused by the exclusive use of electrical simulation inside the optimization loop, especially at system-level. A different approach is the usage of POF, where a suitable solution is selected from the pre-generated set

of optimal solutions, these models are then used hierarchically for system level sizing [17,18].

In MINLP [19], DARWIN [20], SEAS [21] and MOJITO [22,23] device sizing and topology selection are done simultaneously. These methods are more reliable than other topology selection techniques, as they treat the problem in a unified manner. The computation time, however, is extremely high. Koza [24], Lohn [25], Sripramong [26], Shoou-Jin [27] and more recently Hongying [28] presented a design methodology that creates new topologies. This approach is typified by high computation time, which limits the number of components in the circuit. Another issue with bottom-up generation is that designers are suspicious of the generated structures as they may differ "too much" from well-known trusted analog circuits [29]. Fig. 7.2 shows the panorama of analog circuit synthesis contributions.

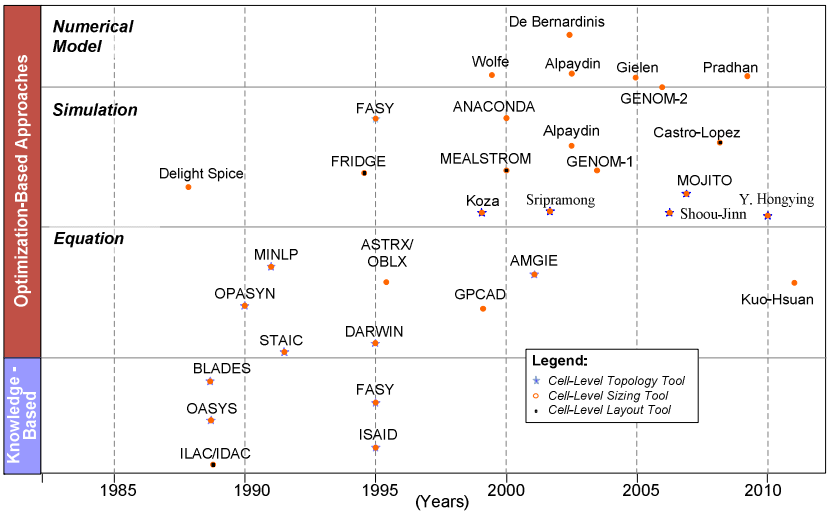

Fig. 7.2 Overview of analog design automation tools

7.3 GPOF-SVM Architecture

GPOF-SVM addresses the problem of automatic specification translation at circuit level, also known as circuit sizing, where from the set of specifications, the designer finds out the sizes for the components (widths and lengths of the transistors, resistors, capacitors, etc.). To verify if the design is robust, i.e., the vast majority of the fabricated circuits will work according to specifications, corner analysis is employed. Corner analysis is among the most common techniques for analog IC design centering, and consists in a worst-case approach where the circuit is simulated over multiple combinations of process parameters variations (power supply, temperature, etc.). In Fig. 7.3 the 27 corners cases obtained by considering 3 values for power supply, operating temperature and library models are shown.

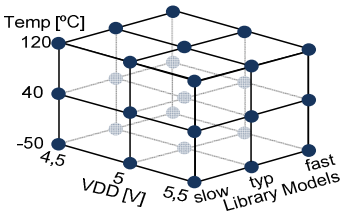

Fig. 7.3 Corner cases example

GPOF-SVM, whose architecture is shown in Fig. 7.4, is based on the elitist multi-objective evolutionary optimization kernel NSGA-II [30], and uses the industrial grade simulator HSPICE® [31] to evaluate the performance of the design. GPOF-SVM targets the design of robust circuits, by allowing the consideration of corner cases during optimization. In addition, an SVM [32], which models the functional feasibility of the circuit, is used to speed up the convergence to feasible areas on the design space.

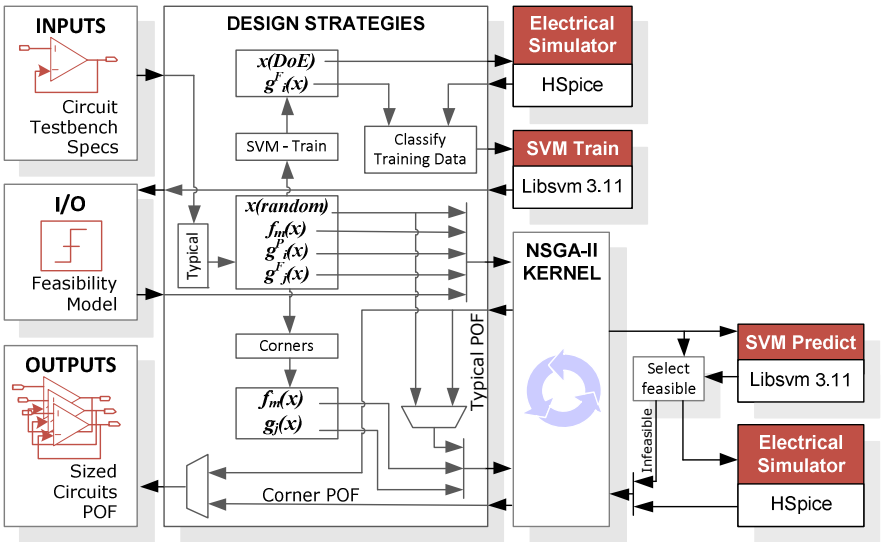

Fig. 7.4 GPOF-SVM architecture

In order to use GPOF-SVM, the designer inputs the circuit netlist and testbench, defines the optimization variables, design constraints and objectives, and the corners cases. Then, GPOF-SVM, models the circuit as an optimization problem, defined by the tuple $\{x,F,G\}$, where x is the vector of design variables, F is the vector objectives and G is the vector of inequality constraints, suitable to be

optimized by the NSGA-II kernel. The functional constraints, which are a sub-set of G, are used to define the functional feasibility regions used to train the feasibility model, where the training data is obtained using fractional Design of Experiments (DOE) to generate a set of circuits that are simulated to evaluate how well they met the functional constraints. The tools' output is a family of Pareto optimal circuits that fulfill all the constraints and represent the feasible tradeoffs between the different optimization objectives. The next subsections provide the details of the architecture using a simple circuit to illustrate the descriptions.

7.3.1 Inputs and Outputs

The inputs from the designer are the circuit and testbench in the form of HSPICE® netlists. The netlist must have the optimization variables as parameters, and must include means to measure the circuit's performance; the corner's parameter variations are also included in the netlist. Fig. 7.5 shows a simple differential amplifier with the testbench schematic and parts of the corresponding netlist.

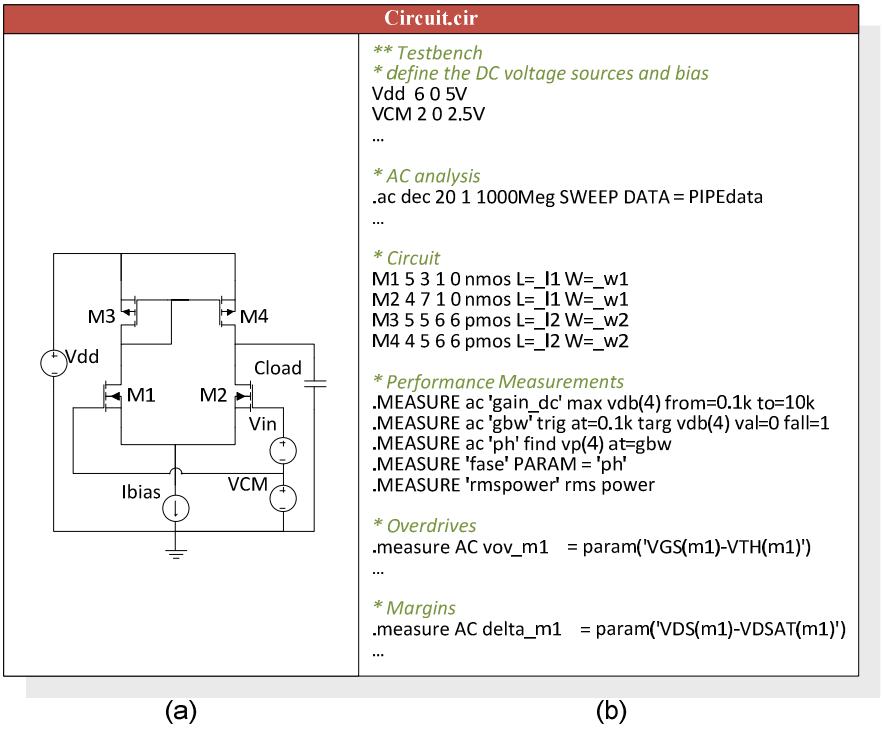

Fig. 7.5 Example circuit: (a) schematic; (b) partial view of netlist

In addition, the designer defines ranges for the optimization variables, design constraints, and optimization objectives. Tables 7.1 and 7.2 illustrate these definitions for the circuit in Fig. 7.5. The output is a family of sized circuits representing the possible tradeoffs between the objectives being optimized.

Table 7.1 Variable ranges for the example in Fig. 7.5

Var.	**W1**	**W2**	**L1**	**L2**	**Ib**
Max.	500.0e-6	500.0e-6	15.0e-6	15.0e-6	400.0e-6
Min.	1.0e-6	1.0e-6	0.35e-6	0.35e-6	30.0e-6

Table 7.2 Objectives and design constraints for the example in Fig.7.5

Constraint	**Measure**	**Target**	**Units**	**Description**
Performance	gbw	≥ 35	MHz	Unit-gain frequency
	pm	$65 \leq \text{pm} \leq 90$	Degree	Phase margin
Functional	vov_m1	$50 \leq \text{vov\_m1} \leq 200$	mV	Vgs –Vt
	vov_m2	$50 \leq \text{vov\_m2} \leq 200$	mV	Vgs –Vt
	vov_m3	$100 \leq \text{vov\_m3} \leq 300$	mV	Vgs –Vt
	vov_m4	$100 \leq \text{vov\_m4} \leq 300$	mV	Vgs –Vt
	delta_m1	≥ 50	mV	Vds – Vdsat
	delta_m2	≥ 50	mV	Vds – Vdsat
	delta_m3	≥ 50	mV	Vds – Vdsat
	delta_m4	≥ 50	mV	Vds – Vdsat
Objective	gain_dc	maximize	dB	Gain DC
	rms_power	minimize	W	RMS power

7.3.2 Optimization Kernel

The optimization engine in GPOF-SVM is a modified NSGA-II to interface with HSPICE® and SVM, used to estimate feasibility and evaluate the individual objective and constraint functions. The NSGA-II was selected over SPEA and other multi-objective evolutionary algorithms because of the good characteristics of the output Pareto [30]. The option of using HSPICE® to evaluate the circuit's performance was due to the accuracy of the results and the availability of models for the devices provided by the foundries. The multi-objective optimization kernel module was designed to solve the problem:

$$\begin{aligned} &\textit{find } x \textit{ that minimize } f_m(x) && m = 1,2,...M \\ &\textit{subject to } g_j(x) \geq 0 && j = 1,2,...J \\ &x_i^L \leq x_i \leq x_i^U && i = 1,2,...N \end{aligned} \tag{7.1}$$

where, x is a vector of N optimization variables, $g_j(x)$ one of the J constraints and $f_m(x)$ one of the M objective functions. Except for minor changes, it was

implemented as in [30], using simulated binary crossover and mutation operators [33], tournament selection, and constrained based dominance check.

7.3.3 Design Strategies

GENOM-POF supports three design strategies: Typical, Corners, and Typical plus Corners. The next subsections describe each of the strategies.

7.3.3.1 Typical (T)

As the name states, in this strategy the circuit is evaluated using only typical conditions, this strategy is faster, and despite the output does not consider the limitations imposed by the corners it is useful for design tradeoffs analysis. First the design problem is described as an optimization problem, and then the NSGA-II optimization kernel can be executed. In order to satisfy the problem formulation in eq. (7.1), the design objectives being minimized are used directly as one of the $f_m(x)$, and the ones being maximized are multiplied by −1. The design constraints are normalized and multiplied by −1, if necessary, according to eq. (7.2).

$$g_i(x) = \begin{cases} \dfrac{p_i - P_i}{|P_i|} \text{ when the design constraint is } p_i \geq P_i \\ \dfrac{P_i - p_i}{|P_i|} \text{ when the design constraint is } p_i \leq P_i \end{cases} \tag{7.2}$$

where, p_j is the measured circuit characteristic, and P_j is the correspondent acceptable limit. Table 7.3 illustrates the objective and constraint functions for the circuit in Fig. 7.5 using the design specifications in Table 7.2.

Table 7.3 $f_m(x)$ and $g_j(x)$ for the example from Fig. 7.5

Performance Constraints	$g_0(x) = \dfrac{gbw}{35\times10^6} - 1$	$g_1(x) = \dfrac{pm}{65} - 1$	$g_2(x) = 1 - \dfrac{pm}{90}$
Functional Constraints	$g_3(x) = \dfrac{vov\_m1}{50\times10^{-3}} - 1$	…	$g_{15}(x) = \dfrac{delta\_m4}{50\times10^{-3}} - 1$
Objectives	$f_0(x) = -gain\_dc$	$f_1(x) = rmspower$	-

7.3.3.2 Corners (C)

In the Corners strategy, the design is optimized from the beginning using all the corners, i.e., for each evaluation the circuit is simulated once for each corner case, this makes it the slower strategy, but the output circuits are feasible in all tested corner conditions. To handle the multiple corners, the objective and constraint functions are modified using eq. (7.3).

$$\hat{f}_m(x) = \max_{c=1,2,..C} \left(f_m^c(x) \right)$$
$$\hat{g}_j(x) = \sum_{c=1}^{C} c_j^c(x) \text{ with } c_j^c(x) = \begin{cases} 0 \text{ if } g_j^c(x) \geq 0 \\ g_j^c(x) \text{ if } g_j^c(x) < 0 \end{cases} \quad (7.3)$$

where, C is the number of corners, and $f_m^c(x)$ and $g_j^c(x)$ are respectively the objective $f_m(x)$ and the constraint $g_j(x)$, as defined for the typical case evaluated in corner case c. In this worst case approach, each objective, which is being minimized, is evaluated using the maximum value obtained from the simulation of circuit in all the corner cases, and each constraint is evaluated as the sum of the normalized violation in all the corner cases where it is violated.

7.3.3.3 Typical Plus Corners (TC)

In this strategy, typical optimization is done until it stops evolving or the maximum number of generation is reached. Then, the typical POF is used as starting point for corner optimization. This strategy is a tradeoff between the execution time and robustness of the solution, and the reduction of the genetic information (localization of the search) imposed by the use of the typical POF as starting point for the corner optimization.

7.3.4 Functional Feasibility Model

In order to improve the convergence of the optimization kernel and reduce the time consumed in the evaluation using HSPICE® a functional feasibility model, which is reusable for different objectives and performance constraints, is used to avoid the simulation of infeasible solutions.

The derivation of the functional and performance constraints depends of the circuit in question, and is up to the designer to define which constraints are functional constraints and which are performance constraints. For example, for the circuit in Fig. 7.5 the performance constraints impose limits to the DC gain and phase margin, while the functional constraints impose limits to the overdrives voltages and saturation of the devices.

This separation has to do with the intrinsic behavior of the circuit, the functional constraints relates to the topology of the circuit and represent design strategies used to ensure proper behavior, whereas the performance constraints relate to some performance metric usually defined from the design specifications. Another way to view this separation is that to ensure the linearity of an amplifier it is recommended to have the transistors in saturation (functional requirement), and for a given design the DC gain must be greater than 30 dB (performance requirement), whereas in another design the unity frequency must be larger 40 MHz (performance requirement).

An important property of the functional constraints is that, for a given process and topology, they must be valid for a wide range of designs, otherwise the functional feasibility model is not reusable. The next sections describe how the functional feasibility model is derived and how it is used to enhance GPOF-SVM performance.

7.3.4.1 Building the Feasibility Model

The feasibility model, which follows the approach taken by GENOM [7], now for the multi-objective case, uses a SVM classifier [32] to estimate the compliance with functional constraints. To train the classifier, a training set is obtained using a fractional DOE strategy, then those points are simulated and their functional feasibility evaluated. The overall functional feasibility is computed as shown in eq. (7.4).

$$ffeas(x) = \frac{1}{J^F}\sum_{j=1}^{J^F} c_j^F(x), \text{ with } c_j^F(x) = \begin{cases} 0 \text{ if } g_j^F(x) \geq 0 \\ g_j^F(x) \text{ if } g_j^F(x) < 0 \end{cases} \quad (7.4)$$

where J^F is the number of functional constraints and $g_j^F(x)$ the functional constraint j.

The sampled points are then sorted into 3 classes, feasible, quasi-feasible, and infeasible, based on the value of *ffeas(x)*. The limits are *ffeas(x)*=0, *ffeas(x)*≥-*T*, and *ffeas(x)*<-*T* respectively.

As noticed on GENOM [7], the data sets are highly unbalanced with very few feasible points, and unbalanced data creates difficulties to the classifier. The main reason is that most classifiers, like SVM, tend to optimize the overall accuracy without considering the weight of relative distribution of each class and they are designed to generalize from sample data to avoid the noise, and in this case they would treat the feasible points as noise and would ignore then. To overcome this issue, strongly infeasible points, i.e., points where *ffeas(x)*<-T_2 with T_2<T, are discarded and not used to train the model.

To select the values T and T_2, first a fractional sampling [34] is performed (for small problems the full combinatory sampling can be used). At this point is likely to have few (or none) feasible points in the data set. Then, set the values T and T_2 in such way that the number of feasible plus quasi-feasible samples is approximately the same as the number of not-discarded-infeasible samples, and at least 5% of the total available samples. This last condition is to ensure that there are a reasonable number of points in each class. The SVM classifier is then trained using the grid search technique suggested in [35].

7.3.4.2 Evaluation with the Feasibility Model

The integration of the feasibility model in the evaluation is done in the trivial way. First the model is used to classify the individuals being evaluated as feasible, quasi-feasible and infeasible, and then unfeasible ones are discarded, and not put for electrical simulation. Due to the nature of the analog IC design space, some precautions must be taken when using the feasibility model: first the model can consider infeasible areas of feasibility that were not sampled, second in the beginning of the optimization, is likely to have only infeasible points, and without electrical evaluation there is no way to tell which ones are better and to guide the evolution of the population properly.

To accommodate these two factors, the functional feasibility pruning is enabled at each generation with 50% change. Moreover, when full feasibility is attained,

i.e., both performance and functional constraints are met; it was noticed that the evolution of the algorithm do not generate a significant amount of infeasible points, and therefore the classification despite being much faster that electrical simulation is not pruning and represents an extra cost, therefore after full feasibility is attained the feasibility model is no longer used.

7.4 Case Study

The GPOF-SVM tool is here demonstrated for typical and corner cases by designing a single ended folded cascode amplifier and a fully differential telescopic amplifier. The folded cascode circuit is described in section 7.4.1, and the telescopic amplifier in section 7.4.2.

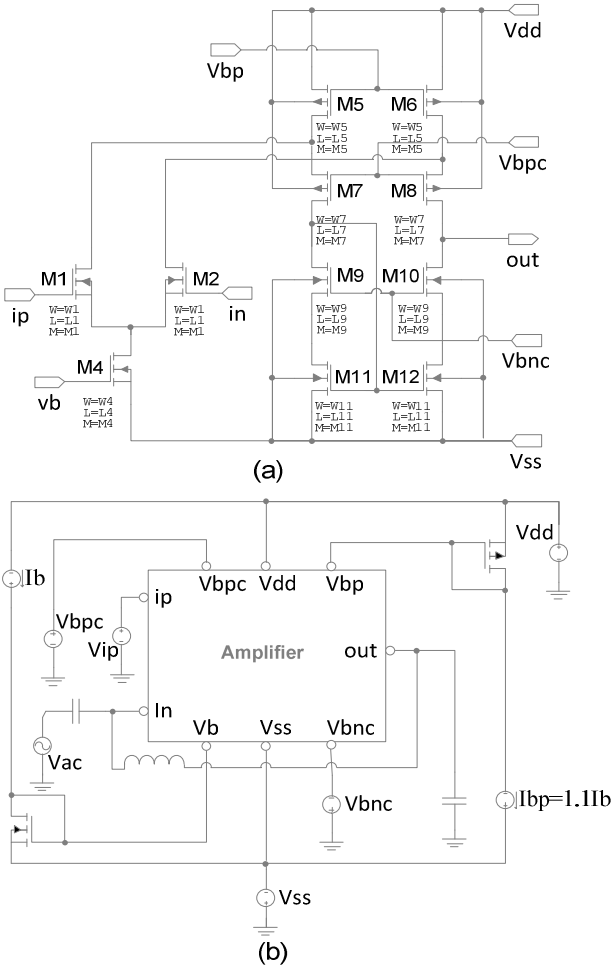

Fig. 7.6 Single-ended folded cascode amplifier (a) schematic (b) testbench

7.4.1 Single Ended Folded Cascade Amplifier

The circuit schematic is shown in Fig. 7.6, and the ranges, objectives and constraints are listed in Tables 7.4 and 7.5. The problem has 15 optimization variables, 2 objectives and 19 constraints. In addition, 9 corner cases were defined using the combination of technology models (typical, fast and slow) and temperature values (-40ºC, 50ºC, 120ºC). All the presented results are for *UMC 0,13µm* technology and include only feasible solutions.

Table 7.4 Variable ranges

Var. [1]	**l1, l4, l5, l7, l9, l11**	**w1, w4, w5, w7, w9, w11**	**Ib [µA]**	**Vbcn [V]**	**Vbcp [V]**
Max.	0.80 µ*m*	400.0 µ*m*	500	0.0	0.4
Min.	0.12 µ*m*	0.24 µ*m*	30	-0.4	0.0

[1] The variables l1 and w1 are dimensions, in [µm], of M1 and M2; l4 and w4 of M4; l5 and w5 of M5 and M6; l7 and w7 of M7 and M8; l9 and w9 of M9 and M10; l11 and w11 of M11 and M12.

Table 7.5 Objectives and design constraints

Constraints	Measure	Target	Units	Description
Performance	gbw	≥ 24	MHz	Unit-gain frequency
	a0	≥ 40	dB	DC Gain
	sr	≥ 10	V/µs	Slew Rate
	pm	55 ≤ pm ≤ 90	Degree	Phase margin
Functional	ov[1]	≥ 30	mV	Vgs –Vt
	d[1]	≥ 1.2	V/V	(Vds – Vdsat)/Vdsat
	osp	≥ 0.3	V	
	osn	≤ -0.3	V	
Objectives	area[2]	minimize	µm	Area
	a0	maximize	dB	DC Gain

[1] The constraint applies to: M1, M4, M5, M7, M9 and M11.
[2] The area is the sum of all the devices gate area (WxL) excluding bias devices.

7.4.1.1 Synthesis

Figure 7.7 shows the Pareto fronts and execution time that were obtained by running the 3 strategies T, C and TC until the max generation limit. The algorithm parameters were: a population of 32 elements, crossover and mutation rate of 90% and 10%, respectively, and 400 generations for T and C, and 200/200 for TC (200 for the first step and 200 for the second step). The functional feasibility model was not used because finding a feasible solution in this example is easy (less than 5 generations), and after finding a feasible solution the optimizer tends to generate only feasible solutions rendering the feasibility model useless.

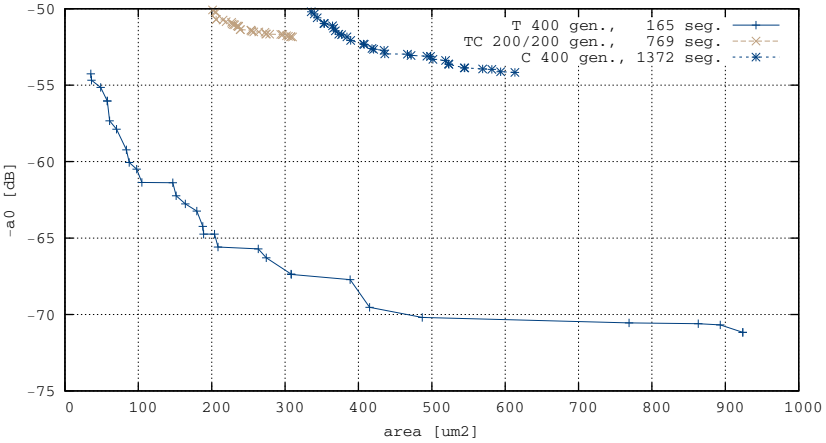

Fig. 7.7 POF obtained using the 3 design strategies T, TC and C

The POF obtained using T was found faster (in 165 seconds), and dominates the others (because it has fewer constraints). TC strategy was faster than C and in a region of the POF provided circuits with smaller area for the same gain; however it does not dominate the one obtained with C completely. By starting the corner

Table 7.6 Summary of the Corner and Typical plus Corner run illustrated in Fig. 7.7

Strategy	Corner (C)			Typical plus Corner (TC)		
	Smaller	Middle	Larger	Smaller	Middle	Larger
Time [s]	1372			769		
Area [μm^2]	335.62	471.75	613.05	201.07	254.72	310.11
Gain [dB]	50.27	53.05	54.16	50.08	51.47	51.84
L1 [µm]	0.44	0.57	0.61	0.42	0.42	0.41
W1 [µm]	27.59	52.01	52.04	16.45	16.44	16.44
L4 [µm]	0.42	0.58	0.63	0.62	0.60	0.72
W4 [µm]	19.88	31.53	54.07	22.36	35.31	50.96
L5 [µm]	0.24	0.27	0.32	0.28	0.28	0.26
W5 [µm]	225.74	187.58	344.72	100.83	104.66	156.58
L7 [µm]	0.33	0.41	0.42	0.40	0.51	0.57
W7 [µm]	188.92	265.16	259.84	126.70	145.28	145.35
L9 [µm]	0.59	0.59	0.60	0.36	0.36	0.36
W9 [µm]	55.30	59.76	60.12	15.59	15.59	15.59
L11 [µm]	0.18	0.17	0.18	0.16	0.16	0.16
W11 [µm]	13.26	14.51	12.93	5.34	5.34	5.34
Ib [µA]	313.51	321.97	291.61	156.65	148.86	154.59
Vbcn [V]	-0.106	-0.1062	-0.1059	-0.0789	-0.0789	-0.0789
Vbcp [V]	0.0862	0.0870	0.0870	0.0890	0.0912	0.0950

optimization from the already optimized typical POF, it is easier to fulfill the additional constraints imposed by the corners and leads to better and faster results, however there is some biasing that narrow the search range. Table 7.6 summarizes the output of C and TC strategies.

In Fig. 7.8 the layout, which was generated using LAYGEN [36,37], is shown, for the extreme points from Table 7.6, i.e. the point with larger gain C larger, and the point with smaller area TC smaller.

7.4.2 Fully Differential Telescopic Amplifier

The fully differential telescopic amplifier circuit including bias schematic is shown in Fig. 7.9, and the variable ranges, objectives and constraints are listed in Table 7.7 and Table 7.8. The problem has 16 optimization variables, 3 objectives, 6 performance constraints and 32 functional constraints. In addition, the same 9 corner cases were defined using the combination of technology models (typical, fast and slow) and temperature values (-40ºC, 50ºC, 120ºC). All the presented results are for *UMC 0,18μm* technology and include only feasible solutions.

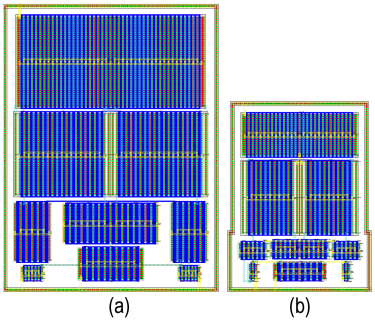

Fig. 7.8 Layout for the extreme points of Table 1.6: (a) C larger, (b) TC smaller

7.4.2.1 Synthesis

The functional feasibility model was used in this example because the convergence of the algorithm to feasible solutions was not immediate. In the single objective case the SVM model reduced the time to obtain the first feasible solution in 15-20%, in the multi-objective case, after 100 runs, the drop was around 10%. The multi-objective optimization explores the solution space more efficiently, reducing the effects of pruning infeasible solutions.

Two 2-D projections of the 3 objective POF, obtained using the TC strategy with parameters: population of 80 elements, crossover and mutation rate of 90% and 10% respectively and 300/200 generations, are shown in Fig. 7.10, and illustrate the effect of corners cases in decreasing the performance achieved by the circuit, Table 7.9 summarizes the synthesis results.

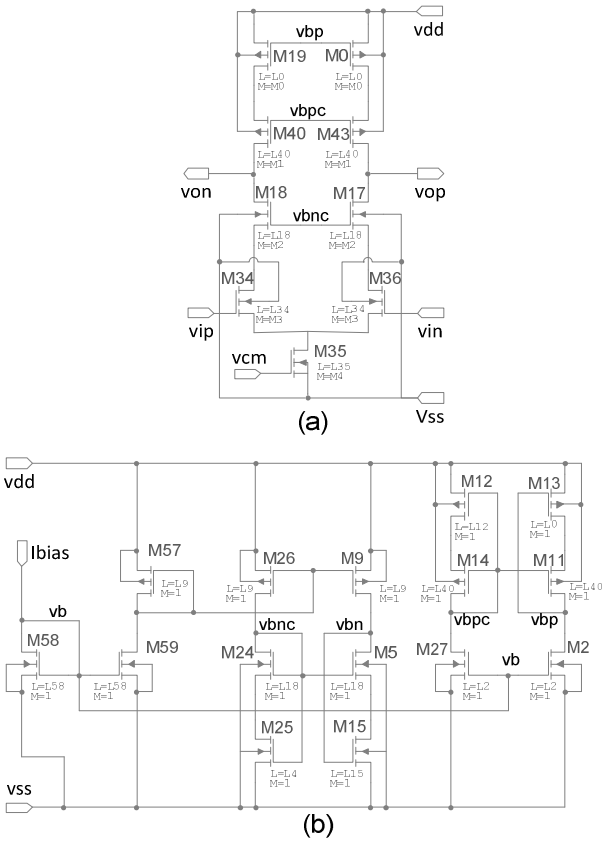

Fig. 7.9 Fully differential telescopic amplifier schematic: (a) amplifier, (b) bias

Table 7.7 Variable ranges

Var.	**L0, L2, L9, L12, L15, L18, L24, L34, L35, L40, L58**	**M0, M1, M2, M3, M4**
Max.	10.0 µ*m*	110
Min.	0.18 µ*m*	1

By having more than 2 objectives the 2-D projections are not monotonic like in the previous example. This is due to the fact that there are solutions that seem to be dominated but have better performance in the objectives not present in the 2-D projection. In Fig. 7.10, the projections of Gain *vs.* Power and GBW *vs.* Power are overlapped, each point in the Pareto front is represented by the 2 points in the graphic that have the same value of Power, one from each of the projections.

Table 7.8 Objectives and design constraints

Constraints	Measure	Target	Units	Description
Performance	gain_dc	≥ 75	dB	DC Gain
	gbw	≥ 100	MHz	Unity-gain frequency
	fase	60 ≤ fase ≤ 90	Degree°	Phase margin
	power	≤ 10	mW	Power
	iavdd	≤ 10	mA	Vdd current
Functional	vov[1]	≥100	mV	Vgs –Vt
	vov_m18, vov_m17	≥45	mV	Vgs –Vt
	vov_m34, vov_m36	≥50	mV	Vgs –Vt
	vov[2]	≤200	mV	Vgs –Vt
	vov[3]	≤300	mV	Vgs –Vt
	d[4]	50 ≤ d ≤ 200	mV	Vds – Vdsat
	d[5]	≥ 50	mV	Vds – Vdsat
Objectives	Power	Minimize	W	Power
	gbw	Maximize	Hz	Unity-gain frequency
	gain_dc	Maximize	dB	DC Gain

[1] The constraint applies to: M19, M0, M40, M43 and M35.
[2] The constraint applies to: M19, M0, M18 and M17.
[3] The constraint applies to: M40, M43, M34, M36 and M35.
[4] The constraint applies to: M40, M43, M17, M18 and M35.
[5] The constraint applies to: M19, M0, M34, and M36.

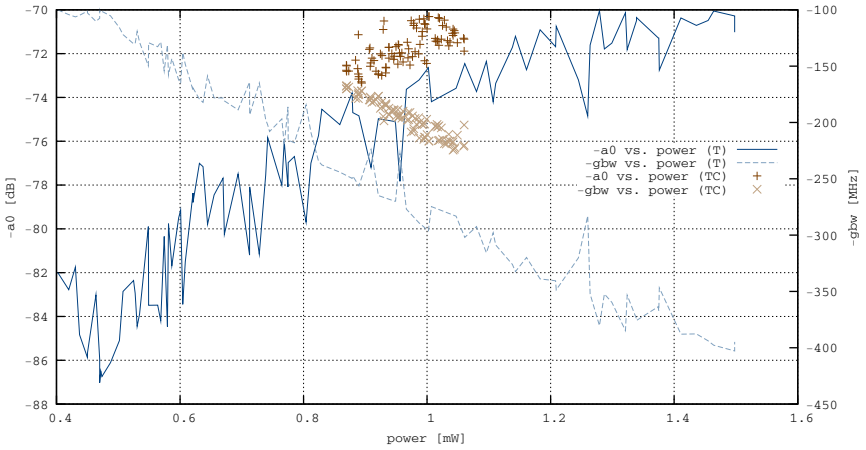

Fig. 7.10 2-D Projections of the 3-D POF obtained using T and TC

Table 7.9 Summary of the synthesis results

Strategy	Typical step (637 [s])			Corner step (5363 [s])		
	Less Power	Larger GBW	Larger Gain	Less Power	Larger GBW	Larger Gain
Power[mW]	0.402	1.498	0.470	0.869	1.04	0.894
GBW [MHz]	100.8	394.8	100.8	170.1	224.6	172.0
Gain[dB]	81.98	71.02	87.02	72.53	70.87	73.32

7.5 Conclusion

The proposed methodology and tool, GPOF-SVM, were used to successfully design well known analog circuits, taking into account robustness consideration by the inclusion of corner cases. Moreover, the multi-objective nature of the IC design synthesis makes it well suited for automatic design using multi-objective optimization strategies. In this approach, the output is not one solution, but a set of completely designed non-dominated solutions, all meeting the specifications. It is up to the designer to select the tradeoff between the concurrent objectives that is more interesting for the target project. The usefulness of GPOF-SVM to designers was shown using different design strategies. First, using the Typical (T) design strategy, the designer explores several design tradeoffs in a matter of minutes, which is useful for system level design. Then, using the Corners (C) or TC strategies the designer can obtain a family of optimum robust circuits that comply with the specification in all corner cases considered. Additionally, in order to enhance the efficiency of the NSGA-II based optimization kernel, a supervised learning strategy, which is based on a SVM approach, is used to create functional feasibility models. These models allow the efficient pruning of the design search space during the optimization process with absolute gains ranging from 10 to 20% in terms of the overall number of required evaluations and larger gains in terms of time consumption once electrical simulation, particularly for large circuits, is clearly more time expensive than the SVM model evaluation. Finally, the layout generation is demonstrated by linking the GPOF-SVM output with the entry of the in-house tool LAYGEN-II.

Acknowledgments. This work was supported by the Portuguese Science and Technology Foundation (Grant FCT-DFRH-SFRH/BD/72698/2010) and by the Instituto de Telecomunicações (Research project AIDA - IT/LA/1112/2011).

References

[1] McClean, B.: IC Market to Top $300 Billion for First Time in 2013, http://www.icinsights.com/ (accessed December 13, 2011)

[2] Mori, K., Yamada, H., Takizawa, S.: System on Chip Age. In: The International Symposium on VLSI Technology, Systems, and Applications, Taipei, Taiwan (1993)

[3] ITRS: International Technology Roadmap for Semiconductors - 2010 Update. ITRS, http://www.itrs.net (accessed December 10, 2011)
[4] Gielen, G.G.E., Rutenbar, R.A.: Computer-aided design of analog and mixed-signal integrated circuits. Proc. IEEE 88(12), 1825–1852 (2000)
[5] Barros, M., Guilherme, J., Horta, N.: Analog circuits optimization based on evolutionary computation techniques. Integration, The VLSI Journal 43(1), 136–155 (2010)
[6] Barros, M.F.M., Guilherme, J.M.C., Horta, N.C.G.: Analog Circuits and Systems Optimization based on Evolutionary Computation Techniques. SCI, vol. 294. Springer, Heidelberg (2010)
[7] Barros, M., Guilherme, J., Horta, N.: GA-SVM feasibility model and optimization kernel applied to analog IC design automation. In: ACM Great Lakes Symposium on VLSI, Stresa-Lago Maggiore, Italy (2007)
[8] Lourenco, N., Horta, N.: GENOM-POF: Multi-Objective Evolutionary Synthesis of Analog ICs with Corners Validation. In: Genetic and Evolutionary Computation Conference, Philadelphia, USA (2012)
[9] Degrauwe, M.G.R., Nys, O., Dijkstra, E., Rijmenants, J., Bitz, S., Goffart, B., Vittoz, E.A., Cserveny, S., Meixenberger, C., Stappen, G., Oguey, H.J.: IDAC: an interactive design tool for analog CMOS circuits. IEEE Journal of Solid State Circuits 22(6), 1106–1116 (1987)
[10] El-Turky, F., Perry, E.E.: BLADES: an artificial intelligence approach to analog circuit design. IEEE Transactions on Computer Aided Design 8(6), 680–692 (1989)
[11] Hershenson, M.D.M., Boyd, S.P., Lee, T.H.: GPCAD: a tool for CMOS op-amp synthesis. In: IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA (1998)
[12] Kuo-Hsuan, M., Po-Cheng, P., Hung-Ming, C.: Integrated hierarchical synthesis of analog/RF circuits with accurate performance mapping. In: International Symposium on Quality Electronic Design, Santa Clara, California, USA (2011)
[13] Gielen, G., Wambacq, P., Sansen, W.M.: Symbolic analysis methods and applications for analog circuits: a tutorial overview. Proc. of IEEE 82(2), 287–304 (1994)
[14] Cheng-Wu, L., Pin-Dai, S., Ya-Ting, S., Soon-Jyh, C.: A bias-driven approach for automated design of operational amplifiers. In: International Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan (2009)
[15] Phelps, R., Krasnicki, M., Rutenbar, R.A., Carley, L.R., Hellums, J.R.: Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. IEEE Transactions on Computer Aided Design 19(6), 703–717 (2000)
[16] Alpaydin, G., Balkir, S., Dundar, G.: An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. IEEE Transactions on Evolutionary Computation 7(3), 240–252 (2003)
[17] Deniz, E., Dundar, G.: Hierarchical performance estimation of analog blocks using Pareto Fronts. In: Conference on Ph.D. Research in Microelectronics and Electronics, Berlin, Germany (2010)
[18] Roca, E., Castro-Lopez, R., Fernandez, F.V.: Hierarchical synthesis based on pareto-optimal fronts. In: European Conference on Circuit Theory and Design, Antalya, Turkey (2009)
[19] Maulik, P.C., Carley, L.R., Rutenbar, R.A.: Integer programming based topology selection of cell-level analog circuits. IEEE Transactions on Computer Aided Design 14(4), 401–412 (1995)
[20] Kruiskamp, W., Leenaerts, D.: DARWIN: CMOS opamp synthesis by means of a genetic algorithm. In: ACM/IEEE Design Automation Conference, San Francisco, California, USA (1995)

[21] Ning, Z.Q., Mouthaan, T., Wallinga, H.: SEAS: a simulated evolution approach for analog circuit synthesis. In: IEEE Custom Integrated Circuits Conference, San Diego, California, USA (1991)
[22] McConaghy, T., Palmers, P., Steyaert, M., Gielen, G.G.E.: Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks. IEEE Transactions on Evolutionary Computation 15(4), 557–570 (2011)
[23] Palmers, P., McConnaghy, T., Steyaert, M., Gielen, G.: Massively multi-topology sizing of analog integrated circuits. In: Design, Automation & Test in Europe, Nice, France (2009)
[24] Koza, J.R., Bennett, F.H., Andre, D., Keane, M.A., Dunlap, F.: Automated synthesis of analog electrical circuits by means of genetic programming. IEEE Transactions on Evolutionary Computation 1(2), 109–128 (1997)
[25] Lohn, J.D., Colombano, S.P.: A circuit representation technique for automated circuit design. IEEE Transactions on Evolutionary Computation 3(3), 205–219 (1999)
[26] Sripramong, T., Toumazou, C.: The invention of CMOS amplifiers using genetic programming and current-flow analysis. IEEE Transactions on Computer Aided Design 21(11), 1237–1252 (2002)
[27] Shoou-Jinn, C., Hao-Sheng, H., Yan-Kuin, S.: Automated passive filter synthesis using a novel tree representation and genetic programming. IEEE Transactions on Evolutionary Computation 10(1), 93–100 (2006)
[28] Hongying, Y., Jingsong, H.: Evolutionary design of operational amplifier using variable-length differential evolution algorithm. In: International Conference on Computer Application and System Modeling, Taiyuan, China (2010)
[29] Martens, E., Gielen, G.: Classification of analog synthesis tools based on their architecture selection mechanisms. Integration, the VLSI Journal 41(2), 238–252 (2008)
[30] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[31] Synopsis, Hspice, `http://www.synopsys.com` (accessed December 14, 2011)
[32] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent System Technology 2(3), 1–27 (2011)
[33] Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9(2), 115–148 (1995)
[34] Montgomery, D.C.: Design and analysis of experiments, 6th edn. John Wiley & Sons, Hoboken (2005)
[35] Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical Report, Department of Computer Science, National Taiwan University (2003)
[36] Lourenço, N., Vianello, M., Guilherme, J., Horta, N.: LAYGEN - Automatic Layout Generation of Analog ICs from Hierarchical Template Descriptions. In: Conference on Ph.D. Research in Microelectronics and Electronics, Otranto (Lecce), Italy (2006)
[37] Martins, R., Lourenco, N., Horta, N.: LAYGEN II: Automatic Analog ICs Layout Generator based on a Template Approach. In: Genetic and Evolutionary Computation Conference, Philadelphia, USA (2012)