Multi-Objective Design of Analog Integrated Circuits Using Simulated Annealing with Crossover Operator and Weight Adjusting

T. O. Weber[1], and Wilhelmus A. M. V. Noije[2]

[1,2]Electronic Systems Engineering Department, Polytechnic School, University of São Paulo, Brazil
e-mail: tweber@lsi.usp.br

ABSTRACT

This paper approaches the problem of analog circuit synthesis through the use of a Simulated Annealing algorithm with capability of performing crossovers with past anchor solutions (solutions better than all the others in one of the specifications) and modifying the weight of the Aggregate Objective Function specifications in order to escape local minimums. Search for the global optimum is followed by search for the Pareto front, which represents the trade-offs involved in the design and it is performed using the proposed algorithm together with Particle Swarm Optimization. In order to check the performance of the algorithm, the synthesis of a Miller Amplifier was accomplished in two different situations. The first was the comparison of 40 syntheses for Adaptive Simulated Annealing (ASA), Simulate Annealing/Quenching (SA/SQ) and the proposed SA/SQ algorithm with crossovers using a 20-minute bounded optimization with the aim of comparing the solutions of each method. Results were compared using Wilcoxon-Mann-Whitney test with a significance of 0.05 and showed that simulated annealing with crossovers have higher change of returning a good solution than the other algorithms used in this test. The second situation was the synthesis not bounded by time aiming to achieve the best circuit in order to test the use of crossovers in SA/SQ. The final amplifier using the proposed algorithm had 15.6 MHz of UGF, 82.6 dBV, 61° phase margin, 26 MV/s slew rate, area of 980 μm² and current supply of 297 μA in a 0.35 μm technology and was performed in 84 minutes.

Index Terms: analog design, CAD, meta-heuristic, simulated annealing, optimization.

1. INTRODUCTION

Analog design is present in most of the technology used in modern applications. From sound amplifiers to cell phones, they are responsible for interfacing the processing capabilities of the digital domain with the real signals existent in the world.

The demands of the microelectronics market and the boundaries of the present technologies make the design of analog integrated circuit a very challenging research topic. While there are several knowledge fields and techniques involved in developing analog designs themselves, from device physics to noise considerations and system analysis, this field is also constrained by market demands, specially small time-to-market. The final design has to meet several specifications (e.g., gain, unity gain frequency, slew rate, power consumption, area...) and also needs to be finished fast in order to save resources and continue the product flow.

Despite of its several constraints and complexity, there is still no structured design flow for analog design. Therefore, the evolution of tools for this domain has been much slower than in the digital domain and consequently the analog section in mixed-signal projects is usually the bottleneck in time required to be accomplished.

To address this problem, several approaches have been proposed using design automation. They are mainly divided in knowledge-based and optimization-based. The initial studies in circuit level synthesis were mostly knowledge-based approaches [1]. In optimization-based approaches the problem is described in a form that can be solved through numerical methods. This approach started in the late 1980's and it is where most of the research focus is on the present [2],[3],[4]. Optimization-based requires less preparatory time before synthesis and it is also more flexible to serve to different analog blocks and topologies.

The tools can also be classified by the way they evaluate a new solution. Simulation-based evaluation rather than equation-based evaluation was progressively being adopted in order to reduce the preparatory effort and to provide higher accuracy.

The optimization-based approach with simulation-based evaluation is promising due to the increase in

computer power and also to the advances made in the optimization and computational intelligence field in general. Due to the multi-objective nature of analog synthesis problems, algorithms attempting to solve them need to be multi-objective or to be able to convert all objectives into one through an AOF (Aggregate Objective Function). While the first option takes into account the whole problem nature and has as a result the Pareto set (best solutions) it also takes much more time to be performed. The approach using AOF is usually faster and uses less memory as it can discard most of the solutions as it uses only one metric to compare them. However, as it simplifies the nature of the problem, it returns only one best solution and it is also more susceptible to be trapped in local minimums.

This paper proposes an optimization-based analog synthesis method that combines the use of Simulated Annealing/Quenching (SA/SQ) with AOF to create a single objective function together with a technique to escape local minimums through the use of multi-objective information, crossovers and weight adjustment. When a local minimum is found, the algorithm selects one of the specifications in which the present solution is not performing well and executes a crossover between it and a previous anchor solution that performs well in this specification. A modification on the weight of the chosen specification is made to allow SA algorithm to search new places in the design space. After all specifications are met, the exploration of the Pareto Front is done using Particle Swarm Optimization.

The paper is organized as follows. In section 2, the basic background on optimization problems is introduced. In section 3, the proposed method for synthesis using SA with crossovers is presented and also the exploration of the Pareto Front using Particle Swarm Optimization. The results of the design of a miller amplifier on a 0.35 µm technology are discussed in section 4 and the conclusion of this work is presented in section 5.

2. BACKGROUND

The synthesis of analog circuits through the use of optimization-based approach with simulation-based evaluation can be represented as in figure 1. Several recent works use this approach [4],[5],[6]. This section will explore the basic knowledge related to optimization concepts and algorithms.

A. Optimization

Analog design synthesis can be seen as an optimization problem with multiple specifications/objectives to address. The design space, which consists of all possible values for transistor lengths and widths, capacitances, resistances and other elements in the design, can be seen as continuous.

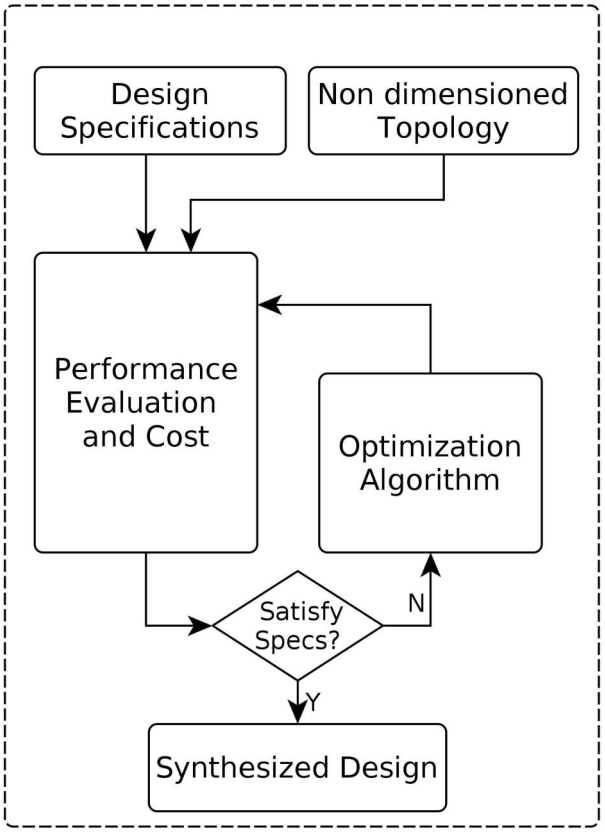

Figure 1. Optimization based approach

The formulation of this type of problem can be seen as follows:

$$\text{minimize } f(\boldsymbol{x}) := [\, f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}) \,] \tag{1}$$

subject to:

$$g_i(\boldsymbol{x}) \leq 0 \quad i = 1,2,\ldots m \tag{2}$$

$$\mathrm{h}i\,(\boldsymbol{x}) = 0 \quad j = 1,2,\ldots p \tag{3}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is a vector containing the decision variables, f_I: $\Re^n \rightarrow \Re$, i=1,...,k are all the objective functions g_i, h_i: $\Re^n \rightarrow \Re$, i =1,...,m, j=1,...,p are the constraint functions of the problem. The number of objective functions in a design is equal to the number of specifications of the design which usually are several in an analog design.

In multi-objective optimization problems is not always possible to say that one solution is better than the other. This is straightforward to see as one solution could be better in one specification and the other solution better in another one. However, there are cases in which one solution outperforms another in all specifications. In this case, we say that one solution dominates the other. The set of solutions of a design that are not dominated by any of the other solutions is called the Pareto set. This set is used to form the Pareto Front, which ultimately represent all the trade-offs involved in the synthesis.

A solution $\boldsymbol{x}^*$ is considered to be in the Pareto set if there is no other feasible solution that dominates $\boldsymbol{x}^*$. One solution $\boldsymbol{a}$ dominates a solution $\boldsymbol{b}$ (denoted as $\boldsymbol{a}>\boldsymbol{b}$) if $f_i(\mathbf{a}) \leq f_i(\mathbf{b})$ for all $i = 1, \ldots, m$ and $\mathbf{a} \neq \mathbf{b}$.

In multi-objective optimizations, we call anchor points the extreme optimal results of the specifications, while the other points in the Pareto front are usually trade-offs between several specifications. Following this definition, the anchor points in a two-dimensional optimization problem involving Gain and Unity Gain Frequency (UGF) would be the solution with greater gain of all others and the solution with greater UGF.

Despite of the several objectives present in an analog design synthesis problem, usually the synthesis methods use algorithms that can work with only one objective function when looking for a single solution. To convert the objectives into only one, an Aggregate Objective Function (AOF) is used. Each objective function is multiplied by a weight and the result is the sum of all weighted cost functions as it can be seen in eq. 4:

$$F = \sum_{i=1}^{n} w_i f_i \tag{4}$$

where F is the AOF function, i is the cost function index, n is the number of cost functions, w_i is the weight of each function and f_i is the cost function of each specification.

B. Optimization Algorithms

This work deals with two algorithms that are used in optimization called Simulated Annealing (SA) and Particle Swarm Optimization (PSO). These algorithms are used in different phases of optimization (as it will be explained in section 3) and their basics will be described in this section.

Simulated Annealing was proposed in [7] and it is based on thermodynamics process called annealing. The physical process consists of submitting a metal to a high temperature and then cooling it slowly in order to reorganize its crystalline configuration in its lowest energy.

SA has several variations [8] and it is one of the most popular optimization algorithms used in analog synthesis. The main advantages are that it can work with any type of cost functions, it is easy to implement and is usually faster than population-based approaches. After the evaluation of the initial solution, there is an optimization loop that makes small modifications on the solution, evaluates the new solution and then performs a decision to keep the old or to adopt the new solution. In analog design synthesis, each solution is evaluated by simulation, reading of the measurements and finally the conversion of these measurements into a cost function. What differentiates Simulated Annealing from Hill Climbing is the decision over the acceptance of the new solution. While in Hill Climbing only better solutions are accepted, in Simulated Annealing new solutions may be accepted even if they are worse than the previous. Equation 5 is used to select the next solution based on the energy costs:

$$S^{k+1} = \begin{cases} N & if\ \ \tau \leq P\,(t^k, N, S^k) \\ S^k & \text{otherwise} \end{cases} \tag{5}$$

where k is the solution index, N is the generated solution, S^k is the old solution, S^{k+1} is the next solution, t^k is the temperature, τ is a uniform distributed random value between 0 and 1 and $P(t^k, N, S^k)$ is the probability of a worse solution to be selected on a given temperature and solution values. The metropolis probability of accepting a worse solution is described in equation 6:

$$P\,(t^k, N, S^k) = min(\,1,\ e^{\frac{C(S^k)-C(N)}{t^k}}) \tag{6}$$

where $C(x)$ is the cost function that evaluates how far a given solution is from the designer objectives. The function used to generate new solutions and the cooling schedule define how effective and fast a SA algorithm is. The set of algorithms which employ the SA concepts with different generation and schedule functions than the classical SA in order to increase the speed at the cost of customizing the algorithm for a specific problem are often called Simulated Quenching (SQ), which is the case of the proposed algorithm.

Particle Swarm Optimization (PSO) is part of a field called swarm intelligence, in which population-based algorithms try to solve problems by using the decentralized intelligence found in groups. Particle Swarm Optimization was initially proposed to simulate movement social behavior, as it can be seen in birds within a flock. James Kennedy and Russell C. Eberhart [9] proposed the algorithm to be applied in optimization. The algorithm allows multiple particles (population) to navigate through a design space while exchanging some information about the places their already visited. The objective of the particles is to find minimum cost solutions, and at each iteration the particles update their positions based on their flight speed. This speed is calculated based on the experience of each particle and from the experience of the group as a whole or from the particle's neighborhood. Equation 7 shows how this calculation can be performed:

$$v_i(t) = \begin{matrix}[\, w v_i(t-1) + c_1 r_1 (x_{lbesti} - x_i(t)) \\ + c_2 r_{2i} \left(x_{gbest} - x_i(t)\right) \,]\end{matrix} \tag{7}$$

where w, c_1 and c_2 are constants that define the influence of each factor on the final speed, r_1 and r_2 are uniform random variables from 0 to 1, v is the velocity vector, t is the time (or iteration number), x_{lbesti} is the local best result and x_{gbest} is the global or neighborhood best result. In [10] it is shown that PSO can be used to analog design problems.

Although the algorithm is simple for single-objective optimization, the use of PSO for multi-objective prob-

lems requires the storage of all non-dominated solutions in and external archive [11]. The evaluation of the new solutions no longer is based on a unified cost function, but based on the dominance among the solutions. The local best and global best are chosen from these lists based on the density of solutions in the neighborhood of each Pareto solution. Therefore, the leaders will always be the solutions that are on less explored regions of the Pareto Front, collaborating for a more uniform exploration.

3. PROPOSED ALGORITHM

The algorithm proposed in this paper for synthesis of analog circuits consists of looking for the global minimum through simulated annealing with crossovers and then further exploring the Pareto front through a combination of the simulated annealing with Particle Swarm Optimization [12]. In this section both the use of simulated annealing with crossovers and use of particle swarm optimization will be described

Integration of Simulated annealing with Genetic Algorithms concepts have been made on the optimization field [13] [14]. These attempts are usually population based and therefore keep several individuals performing SA and then at some point allow them to perform crossover among themselves.

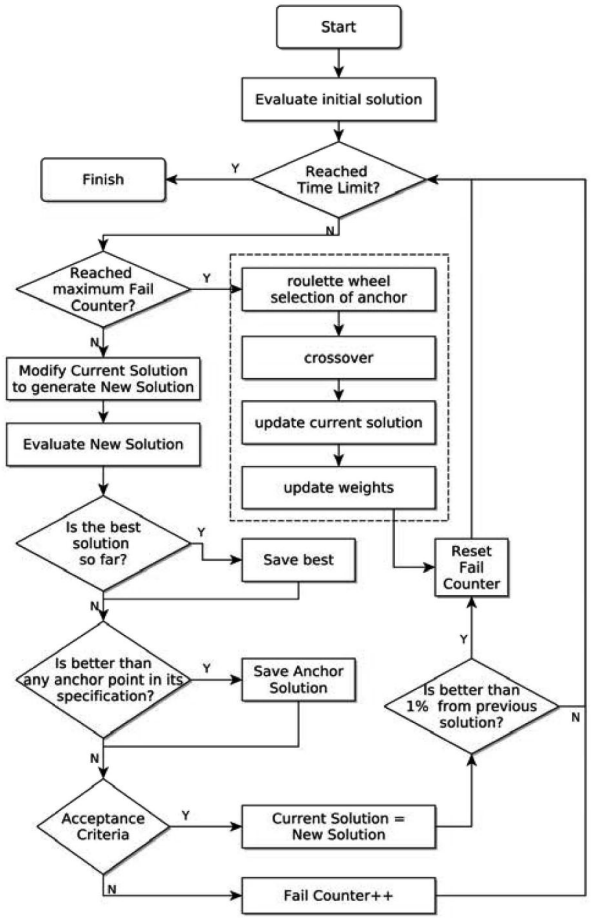

Figure 2. General flow of Simulated Annealing with Crossovers and Weight modification

The algorithm proposed in this paper uses one individual and selects solutions to crossover based on the memory of past promising solutions. Using these memories the algorithm can effectively escape from local minimums by using multi-objective information. When using an AOF, the specifications in which the present solution is performing worst are usually the responsible for creating the local minimum. Therefore, during the SA exploration of the design space the algorithm can collect enough information so that it can use them through crossover when it becomes necessary.

There is one solution memory slot for each specification. The best solution found so far for each specification is saved during the optimization process and used to perform crossover with the present solution when it is stuck in a local minimum. Figure 2 shows the flow diagram of the algorithm. A description of the procedures used is discussed following.

Choosing which solutions to save is an important step since they are going to be the responsible for taking the present solution away from the minimums. Anchor solutions are the ones that are the responsible for the extreme points in a Pareto front. They represent the best solution already achieved for each given specification. Although they are the best, selecting directly these anchor points to be kept in the memory is not interesting for the flow of Simulated Annealing. A solution can perform extremely well in power consumption, for instance, but perform terribly in all other specification. The crossover between the present solution and this anchor solution would result most likely in a not functional solution. A more interesting solution to be saved is one that is not too much far from the overall cost function achieved by the present solution. It should be better than the present solution on its worst specification however not much worse in the overall functionality. This trade-off can be accomplished through the computation of the weighted sum of the cost for the specification and the overall cost as it can be seen in the equation 8:

$$AV_j = C_j\ 0.95 + T\ 0.05 \quad (8)$$

where j is the index of the specification, AV_j is a value used to compare the present solution with the anchor values already in memory to see if the present can be substitute the previous anchor on the memory, C_j is the cost of the solution only for the specification j, T is the total cost of the solution. After each cycle, the present solution has its partial costs compared with the anchor points through this equation and the ones that are better are kept.

The detection of local minimums is based on the number of consecutive failures. If the SA algorithm is not able to find a solution within the maximum number previously defined, it is considered that it is in a local minimum. It is important to avoid that minimal improvements reset the failure counter, which would result in cases when a local minimum is never detected due to small refinements on the solution. Therefore, only solutions that improve the results in more than 1% of the cost func-

tion are considered real improvements. Equation 9 shows the behavior of the fail counter:

$$FC^{K+1} = \begin{cases} 0 & if \ \frac{c(s^k) - c(N)}{c(s^k)} > 0.01 \\ FC^k & if \ 0 < \frac{c(s^k) - c(N)}{c(s^k)} < 0.01 \\ FC^k + 1 & otherwise \end{cases} \quad (9)$$

where FC^{k+1} is the next fail counter value and FC^k is the present fail counter value. The selection of which of the previous solutions will be mixed with the present one is based on how far each of the objectives is from being achieved. The worse present objectives will grant more chance to their respectives anchor solutions to be selected to crossover with the present solution. This is performed through a roulette selection among the anchor points as it can be seen in figure 3 for a 5 objective e case.

The crossovers are the responsible for taking a solution out of a local minimum. This process produces a child from two parents, which in this case are the present solution and the selected anchor solution. It is expected that the mixture between these solutions will create a child that has characteristics of the anchor (which will take the solution away from the local minimum) as well as some characteristics of the present solution.

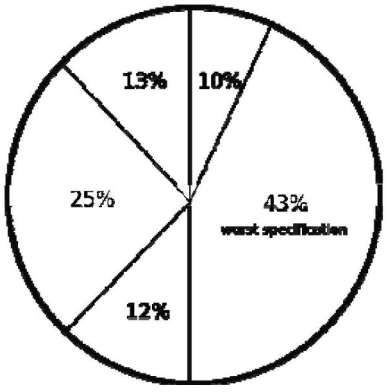

Figure 3. Example of roulette selection based on specifications. The worst specifications have more chance to be selected randomly

A popular crossover operator used for analog design synthesis is the one-point crossover. In this work, however, we used a different crossover based on conditional probability that has yielded better results. The probability of a gene to be from one of its parent depends on from which parent was the previous gene. This type of crossover allows more degrees of freedom than the one-point or two-point crossover while keeping a more structured child than the uniform crossover. We used the conditional probabilities displayed on equation 10:

$$\begin{aligned} P(g_i = present \,|g_{i-1} = present) &= 0.5 \\ P(g_i = anchor|g_{i-1} = anchor) &= 0.6 \end{aligned} \quad (10)$$

where g is the gene, i is the gene index, *present* is a gene from the present solution and *anchor* is a gene from the anchor solution.

Besides directing the solution to a better spot by applying crossover to a previous anchor point, it is also interesting to change the weights to prevent Simulated Annealing from going to the same local minimum once again. This can be made by increasing the weight of the selected specification after it is sorted on the roulette wheel. There will be a higher cost involved in getting worst solutions in that specification once the weight is increased and SA will try to preserve it while searching for a better solution. However, as we are dealing with memory of previous solutions, it is important to keep the same scale when comparing solutions collected with different weights for the specifications. There are two possible ways of performing that: by updating all costs of previous saved solutions when one modification on weight occurs, or by using these weight modifications as penalty factors. The last option keeps all memories preserved and only requires that the general SA algorithm add an extra penalty factor to the overall cost of a solution. By using a penalty factor it is also easy to separate the original cost from the modified one and save anchor points only based on the original weights.

The equation for the general Aggregate Objective Function used can be seen in equation 11:

$$C(x) = \sum_{i=1}^{n} w_i f_i + \sum_{j=1}^{m} p_j g_i \quad (11)$$

where $C(x)$ is the cost function, i is the index of specifications, n is the total number of specifications, w_i is the weight for each specification, f_i is the individual cost of each specification, j is the index of penalty factors, m is the total number of penalty factors, p_j is the weight of each penalty factor and g_i is the individual cost of each penalty. The penalties described in this equation are the ones related to transistors out of the desired region of operation.

Equation 12 describes the cost to be used after one or more local minimums are found. A vector *sp* containing all specifications previously sorted on the roulette selection is used to keep track of the modifications on the weights.

$$C_m(x) = C(x) + k \sum_{i=\max(1,li-5)}^{li} w_{sp(i)} f_{sp(i)} \quad (12)$$

Where $Cm(x)$ is the total cost after modification by adjustment, k is the multiplication factor applied to local minimums (in this work we used $k = 5$), l_i is the index of the last value on the *sp* vector, and finally $C(x)$, w and f are the same as in equation 11.

The only update required after the weight modification is from the old solution cost. This is necessary so the following comparisons do not reject the solution after crossover based on its extra weight on the selected specification.

To perform the exploration of the Pareto front, two subsequent phases are added after the synthesizer finds what it considers the global optimum. The second phase

is the exploration of the anchor points and the third phase is the use of Particle Swarm Optimization to further explore the front [12].

The process described on this section so far explored what is considered phase 1, in which all objectives are combined into one through weighted sum and solved using Simulated Annealing with crossovers. The final of this phase occurs when a solution that respect all specifications defined by the user is found and also no further improvements are being accomplished by Simulated Annealing. Therefore, if the designer is only interested in the best solution and not in exploring the Pareto Front, this phase is enough to provide him with a good answer.

The solution from phase 1 is used as a seed to start several optimizations each directed specifically to one of the specifications. Each optimization is directed by setting to zero all specification weights except the one of interest. However, in order to keep the new solutions within the minimal specifications, only the weights of the measurements after they achieved the minimum specifications are set to zero. Therefore, if one of the specifications goes out of the minimal, it will have a strong cost associated to it and the optimizer will probably return to the previous solution. In the path from the seed to each anchor point, all data are collected in order to aid the Particle Swarm Optimization.

With all anchor points explored and the collected data, Particle Swarm Optimization for multi-objective optimization is performed. It can use the anchor solutions as particles and the history from the seed to each of them as experience to each particle. This is one of the major reasons for selecting PSO as the population-based algorithm in this stage, as it can fully use the information extracted from the two first phases. The disadvantage of selecting PSO to explore the Pareto front is the time spent on updating the external archive after each population evaluation.

In order to plot the Pareto front, two specifications are chosen by the designer as fixed axes and multiple graphs of the Pareto front are plotted using the other specifications as the third axis of each graph. A filter procedure for each graph needs to be done to select from the n-dimensional Pareto solutions which ones are non-dominated when considering the new 3-dimensional objective.

In relation to the state-of-the-art, the proposed algorithm differentiates from the current approaches by using multi-objective information in the search for better solutions while looking for the best solution for the AOF. In literature, multi-objective information (and not only the AOF result) is used in synthesis only in cases in which the search for a pareto front is explored, while in this algorithm this information is used to perform crossover and adjust weights of the AOF. Pareto exploration is restricted to a final stage using a population-based algorithm that uses as seed information collected during the initial stages.

4. RESULTS

To check the performance of the simulated annealing algorithm with crossover, the synthesis of a Miller amplifier (figure 4) in the AMS 0.35 µm CMOS technology was performed. Two different performance tests were used to evaluate the algorithm. All tests were performed using an Intel(R) Core(TM)2 Duo CPU with a 1.80GHz frequency and with a RAM memory of 3GB.

The first test compares three algorithms: the Adaptive Simulated Annealing (ASA) [15], a Simulated Annealing/Quenching without crossovers using the same structure as our algorithm however without crossovers, and finally the Simulated Annealing/Quenching with crossovers proposed in this paper. The objective is to achieve the specifications described in Table I in a 20-minute bounded synthesis. As the techniques used in this work are stochastic procedures, the evaluation is done through 40 syntheses procedures and further each pair of techniques is compared through the non-parametric Wilcoxon-Mann-Whitney test [16]. The null hypothesis is that the probability of a sample from one technique to be greater than the other is equal 0.5. Rejecting this hypothesis means that one technique has more chances of yielding a better answer than the other. Equation 13 shows the hypotheses:

$$H_0: P(x_i > y_i) = 1/2$$
$$H_1: P(x_i > y_i) \neq 1/2 \quad (13)$$

where H_0 is the null hypothesis, H_1 is the alternative hypothesis, x_i is an observation of one technique and y_i is an observation of another technique.

Table 1. Specifications of the Miller Amplifier for AMS 0.35 µm

Parameter	Specification
Vdd	1.65 [V]
Vss	-1.65 [V]
UGF	> 15 M [Hz]
DC Gain	> 80 dB [V]
Phase Margin	> 60º
Slew Rate (pos.)	> 20 M [V/s]
Slew Rate (neg.)	< -20 M [V/s]
ICMR	> 2 [V]
Output Swing	> 2 [V]
CMRR	> 80 dB [V]
PSRR	> 70 dB [V]
Isupply	< 300 µ [A]
Area	< 1000 [µm]²

The results of the first test can be seen in Table 2. The mean and median of the SA/SQ with crossovers technique were better than the ones from ASA and SA/SQ. To verify this difference, the Wilcoxon-Mann-Whitney comparison test with a significance of 0.05 was performed and the result shows that both ASA and SA/SQ are different from SA/SQ with crossovers (Table 3). This means that a synthesis using SA/SQ with crossovers have greater change to return a solution close to the specifications than the other algorithms.

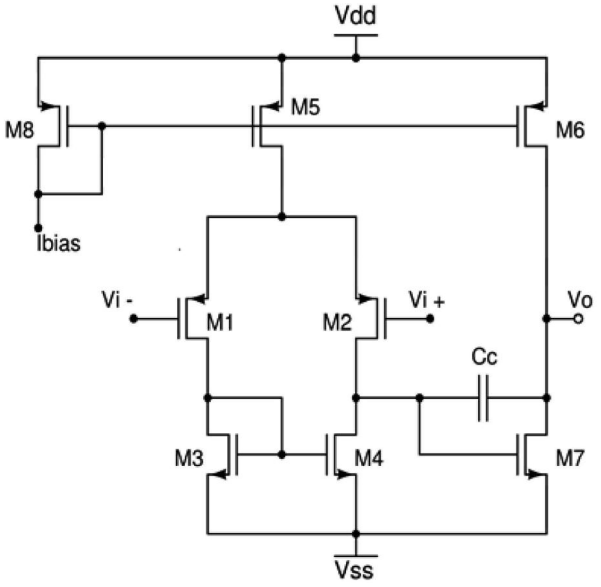

Figure 4. Miller amplifier topology

Table 2: Comparison between Simulated Annealing (SA) and SA with Crossovers for 40 synthesis procedures bounded by 20 minutes each. Values are the results of the cost function (equation 11).

Measure	ASA	SA/SQ	SA/SQ with Crossovers
Mean	13.68	17.05	3.3
Median	4.96	0,83	0.45
Variance	446.11	687.85	37.03
Std. Deviation	21.12	26.23	6.09
Minimum	0.0735	-0.0017	-0.005
Maximum	93.83	99.02	25.00

The second test was the synthesis of a Miller Amplifier not bounded by time, but by number of iterations without improvement. The algorithms tested were the SA/SQ and the SA/SQ with crossovers, as these were the algorithms that performed better in the first test (using the median as comparison), which were also compared with a manual design found in literature [17].

Table 3: Results of the Wilcoxon-Mann-Whitney test for the comparison among the ASA, SA/SQ and SA/SQ with crossovers techniques.

Technique 1	Technique 2	P	Different (P>0,05)
ASA	SA/SQ	0.904	0
ASA	SA/SQ with crossovers	0.003	1
SA/SQ	SA/SQ with crossovers	0.002	1

Results of the second test can be seen in Table 4 and show that the crossovers effectively improved the final result, although it took 27% more time than the regular SA/SQ. The results also show that the algorithm can perform well when compared with manual designs, as the solution had similar measurements although some specifications were better achieved by the manual design and some by the synthesized version. The final dimensions are in Table 5.

These same results of the version with crossovers were used as seed to the Pareto set exploration. The results for the search for the anchor solutions and the time spent on each optimization can be seen in Table 6.

Table 4: Optimized Results of the Miller Amplifier Synthesis for AMS 0.35 µm using only Simulated Annealing, using Simulated Annealing with Crossovers and a result found in literature for the same technology [17]

Parameter	SA/SQ	SA/SQ with Crossovers	Manual Design found in [17]
UGF M[Hz]	14.96	15.60	15.19
DC Gain dB[V]	75.91	82.63	85.42
Phase Margin [º]	60.02	61.23	60.07
Slew Rate (pos.) M[V/s]	19.22	22.18	9
Slew Rate (neg.) M[V/s]	-25.99	-31.55	
CMRR dB [V]	73.33	82.03	89.9
PSRR dB[V]	79.78	85.10]	-
Current Supply µ [A]	299.31	296.88	259.9
Gate Area [µ m]²	255.00	980.47	925*
Crossovers	-	19	-
Synthesis time [min.]	66	84	-

* we did not consider the M9 transistor found in the reference for calculation since it does not exist in the design used for this work

Table 5: Dimensions of the Miller Amplifier after Synthesis

Device	Value (W/L)
M1	69.2 µm / 5.92 µm
M2	69.2 µm / 5.92 µm
M3	14.1 µm / 2.23 µm
M4	14.1 µm / 2.23 µm
M5	136 µm / 0.4 µm
M6	100 µm / 0.4 µm
M7	100 µm / 0.4 µm
M8	136 µm / 0.4 µm
C_c	1 pF
I_{bias}	27 µ [A]

Table 6: Time and best values for each optimization of the anchor point search for the Miller amplifier (these values are not all achieved in a same solution)

Parameter	Best value	Time (minutes)
UGF	20.27 M [Hz]	14.4
DC Gain	86.08 dB [V]	19.2
Phase Margin	63.28 º	5.5
Slew Rate (pos.)	22.71 M [V/s]	2.0
Slew Rate (neg.)	-32.75 M [V/s]	6.9
ICMR	2.99 [V]	34.0
Output Swing	2.95 [V]	3.2
CMRR	82.69 dB [V]	1.8
PSRR	87.18 dB [V]	10.8
Current Supply	296.88 µ [A]	1.73
Area	352.3 [µm] ²	20.52

After applying the Particle Swarm Optimization, area and current supply specifications were chosen to be the fixed axes of the Pareto front graphics, which can be seen in figure 5. This phase of the optimization took 182 minutes (including the anchor solutions search).

The 3D graphs show the trade-offs of the circuit, such as the need of greater area and current supply in order to achieve unity gain frequencies as high as 20 MHz. The solutions showed in Table 3 as well as in the graphs indicate valuable information to the designer, such as how far can the solution can increase one objective while still keeping the other measurements within specifications (e.g., the gain in the design can go to 86 dB while all other specifications are maintained).

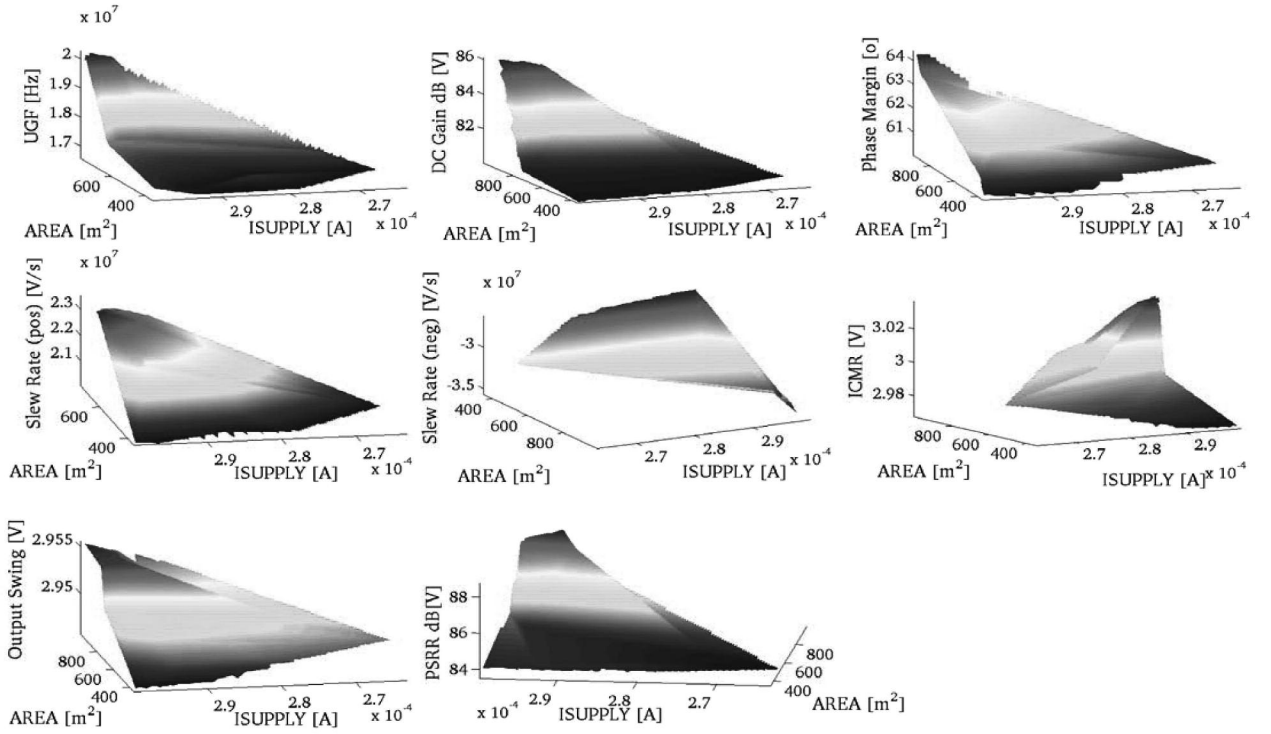

Figure 5. Pareto Front for Miller Amplifier in AMS 0.35 µm using Area and Current Supply Specifications as Fixed Axes

5. CONCLUSION

A method for analog synthesis with Simulated Annealing using crossovers with past anchor solutions and auto weight adjustment was presented. The method uses the speed of Simulated Annealing aided by multi-objective information to escape local minimums. The performance of the algorithm was tested through the synthesis of a Miller amplifier in AMS 0.35 µm CMOS technology. The final design had 15.6 MHz of UGF, 82.6 dBV, 61º phase margin, 26 MV/s slew rate, area of 980 um² and current supply of 297 µA. The design was performed in 84 minutes and the Pareto front was explored resulting in 3D plots of the design space in 182 minutes. Comparisons with the Adaptive Simulated Annealing, a general Simulated Annealing/Quenching algorithm and the proposed one in a 20 minutes bounded synthesis were performed and indicate the proposed algorithm has more probability of achieving better results.

ACKNOWLEDGEMENTS

The research work was partially supported by the Instituto Nacional de Ciência e Tecnologia de Sistemas Micro e Nanoeletrônicos (INCT/NAMITEC) under CNPq process no.573738/2008-4 and under FAPESP process no. 2008/57862-6.

REFERENCES

[1] M.G.R. Degrauwe et al., “IDAC: an interactive design tool for analog CMOS circuits,” *IEEE Journal of Solid-state Circuits*, vol. 22, pp. 1106–1116, 1987.

[2] W.Nye, D.C. Riley, A. Sangiovanni-Vincentelli, and A.L. Tits, “DELIGHT.SPICE: An optimization-based system for the design of integrated circuits,” *IEEE TCAD, 501,7*, 1988.

[3] G.G.E. Gielen, H.C.C. Walscharts, and W.M.C. Sansen, “Analog circuit design optimization based on symbolic simulation and simulated annealing,” *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, Jun., 1990, pp. 707-713.

[4] B. Liu et al., “Analog circuit optimization system based on hybrid evolutionary algorithms,” *Integration, the VLSI Journal*, vol. 42, no. 2, April, 2009, pp. 137–148.

[5] M. Barros, J. Guilherme, and N. Horta, “Analog circuits optimization based on evolutionary computation techniques,” *Integration, the VLSI Journal*, vol. 43, Jan., 2010, pp. 136–155.

[6] B. Liu, F.V. Fernandez, and G.G.E. Gielen, “Efficient and Accurate Statistical Analog Yield Optimization and Variation-Aware Circuit Sizing Based on Computational Intelligence Techniques,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, Jun., 2011, pp. 793-805.

[7] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, May, 1983, pp. 671-680.

[8] A. I. Pereira and E. M. G. . Fernandes, “A study of simulated annealing variants,” in *Proceedings of XXVIII Congreso de Estadística e Investigación Operativa*, 2004.

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

[10] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog Integrated Circuits and Signal Processing*, vol. 63, April, 2010, pp. 71–82.

[11] M. Reyes-Sierra and C. A. C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, 2006, pp. 287-308.

[12] T.O. Weber and W.A.M. Van Noije, "Analog design synthesis method using simulated annealing and particle swarm optimization," in *Proceedings of the 24th symposium on Integrated circuits and systems design*, 2011, pp. 85–90.

[13] M. Ohlídal, "Hybrid parallel simulated annealing using genetic operations," in *Proceedings of Mendel 2004 10th International Conference on Soft Computing,* 2004, p. 89-94.

[14] S.K. Tiwary, P.K. Tiwary, and R.A. Rutenbar, "Generation of yield-aware Pareto surfaces for hierarchical circuit design space exploration," in *Design Automation Conference*, 2006, pp. 31–36.

[15] H. B. Mann, D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," in *The Annals of Mathematical Statistics*, v.18, n.1, 1947, p.50–60.

[16] A.L. Ingber, "Very Fast Simulated Re-Annealing", *Mathematical and Computer Modelling*, v. 12, 1989, pp. 967-973.

[17] F.P. Cortes and S. Bampi, "Miller OTA Design using a Design Methodology Based on the gm/Id and Early-Voltage Characteristics: Design Considerations and Experimental Results," in *Proceedings of XII Workshop Iberchip*, 2006, pp. 10-13.