Fast Optimization of Nano-CMOS Mixed-Signal Circuits Through Accurate Metamodeling

Oleg Garitselov[1], Saraju P. Mohanty[2], and Elias Kougianos[3]
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu)
University of North Texas, Denton, TX 76203, USA.
E-mail ID: omg0006@unt.edu[1], saraju.mohanty@unt.edu[2], and eliask@unt.edu[3]

***Abstract*—Design optimization methodologies for AMS-SoCs with analog, digital, and mixed-signal portions have not received significant attention, due to their high complexity. In mixed-signal circuit design, optimization and simulation are still important issues as they make the design cycle longer. This paper presents a new approach to reduce design optimization time. The approach relies on the fact that optimization carried out over a metamodel (which is an abstracted representation of the circuit model) instead of the actual circuit will allow fast design space exploration and reduce the design cycle time. In this paper three different optimization algorithms are compared: exhaustive search, tabu search and simulated annealing algorithms are analyzed to determine their suitability for metamodeling-based optimization. A ring oscillator is designed for a 45 nm nano-CMOS technology and the post-layout parasitic netlist is used as a test case for a comparative study. It is observed that the metamodel-based simulated-annealing optimization algorithm achieved $\sim$9000$\times$ speed-up over the actual circuit-based optimization.**

I. Introduction

Modern consumer electronic systems are essentially built as heterogeneous components which consist of diverse analog and digital circuits and their interfaces on a single board or die. Typically, they are Analog/Mixed-Signal Systems-On-Chip (AMS-SoC) where analog, digital and RF portions are integrated on the same die for cost-performance tradeoffs [1], [2], [3]. Present day AMS-SoCs are of gigascale complexity and consist of CMOS transistors of nanoscale process technologies. To compound the problem for such highly complex systems, the time-to-market has been reduced significantly and specifications have been tightened, especially for products related to an already crowded wireless market. In such a situation, efficient design approaches are more important than ever in order to produce error free AMS-SoCs on time.

Existing tools predominately focus on digital SoCs possibly with hardware-software co-design [4], [5], [6]. However, design approaches for AMS-SoCs with analog, digital, and mixed-signal portions have not received much attention, due to the increased complexity of the designs. For such large and complex nanoscale technology based mixed-signal circuits **the two most important issues are**: How fast can design space exploration be performed? (2) How fast can layout generation and optimization be performed? This paper presents research in that direction by introducing a metamodeling-based design flow in which the design space exploration is performed on the metamodels of circuit models instead of the actual circuit model to obtain fast and accurate designs with minimal effort.

This paper introduces an approach called metamodeling based optimization to acquire accurate physical design (layout) with minimal simulation time. The creation of an accurate metamodel provides designers with a simple and less computationally expensive model which is accurate enough to produce a close optimized result for the given parametric problem.

The **novel contributions of this paper** are as follows:

1) This paper proposes a metamodeling-based design flow for fast and accurate optimization of nano-CMOS complex mixed-signal circuits.
2) The different phases of the design flow are identified which will be undertaken for future research, such as statistical sampling for metamodeling, mixed-signal optimization algorithms, and optimization of metamodels.
3) As a step towards optimization, three distinct optimization algorithms are discussed. They include exhaustive search, tabu search and simulated annealing, and are applied to nano-CMOS technology.
4) A 4 5nm ring oscillator (RO) is designed and characterized, including the layout, and used as a case study. The oscillator is characterized for frequency, and power.
5) The full RLCK (resistance, capacitance, and self and mutual inductance) parasitic extraction is performed and compared to the schematic of the oscillator. The metamodel is generated on the parasitic netlist.
6) The use of optimization techniques with and without metamodeling are compared.

The rest of the paper is organized as follows. Discussion of related research is included in Section II. A case study mixed-signal circuit, a ring oscillator, is presented in Section III. Section IV introduces three different optimization algorithms which are applied on the sample circuit. Section V proposes the metamodeling optimization approach. Section VI discusses conclusion and future research.

II. Related Prior Research

In [7], the comparative analysis of multiple optimization methods is provided. The optimization is used for the calibration of TCAD simulators and is not used for optimization on a circuit. A heuristic tabu search optimization algorithm is proposed in [8] and compared with simulated annealing for

[0]This research is supported in part by SRC award P10883 and NSF awards CNS-0854182 and DUE-0942629.

an operational amplifier. The optimization is not performed on an actual SPICE netlist and analytical formulas are used for characterizing the optimization. In [9], an automated analog circuit design was proposed and used to optimize a two-stage CMOS op-amp for 2μm and 1.2μm CMOS processes. The general theory of metamodeling, sampling techniques, and computer experiments can be found in [10], but do not address nano-CMOS technologies. Design characterization of ring oscillators including jitter, power and frequency can be found in [11], [12] which cover the design of the ring oscillator but do not cover optimization or metamodeling.

In [13], a geometric program formulation for transistor sizing and optimization is investigated but it is not parasitic aware and slow to converge. Other geometric programming based methodologies include ORACLE [14], which is applicable to metal-mask configurable circuits only, applications to inductor optimization [15] and automatic synthesis of CMOS LDO regulators [16]. OPERA, a framework based on ellipsoidal uncertainty is presented in [17]. Macromodeling approaches are given in [18], [19], [20] and [21]. Design space exploration approaches from high-level descriptions of analog circuits are given in [22]. The use of neural networks in the automatic synthesis of op-amps is explored in [23].

RF-specific transistor sizing with explicit parasitic estimates is given in [24]. A layout-aware modeling approach for analog synthesis is given in [25] and [26].

To the best of the authors' knowledge, this is *the first application of pure metamodeling to parasitic-aware netlists* (refer Section IV-A) for analog/mixed-signal design optimization.

III. A 45 nm CMOS Ring Oscillator: The Case Study Circuit

A Ring Oscillator (RO) consists of an odd amount of inverters connected in series with positive feedback, as shown in figure 1. The initial design parameters are: length of transistors $L = 45$ nm, width of NMOS $W_n = 120$ nm and width of PMOS $W_p = 240$ nm, as shown in figure 1, with a nominal operating voltage of $V_{dd} = 1$ V. Assuming equal fall and rise times of each inverter transistors, the frequency of oscillations for ring oscillator is calculated using $f = 1/(2Nt_p)$, with N the number of inverters and t_p the inverter's propagation delay [27].

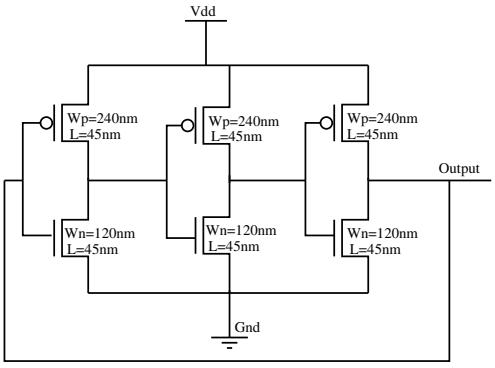

Fig. 1. Logical design of the ring oscillator for 45nm CMOS technology.

It is difficult to estimate the effect of parasitics without doing the layout. The creation of the layout, figure 2, provided the full parasitic SPICE netlist.

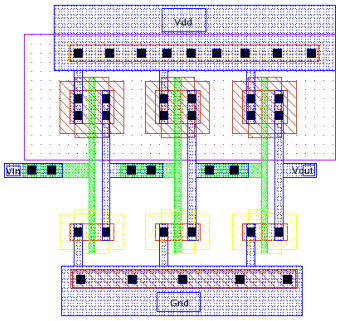

Fig. 2. Physical design of the ring oscillator for 45nm CMOS technology.

The circuit netlist with post-layout parasitics demonstrates a dramatic decrease in frequency versus schematic simulations. The presence of parasitics in this simple circuit increases the run time of simulation by 5 seconds. In complex circuits with hundreds or even thousands of transistors, the simulation time will be in the order of days if not weeks, depending on the complexity of the circuit. Table I shows a comparison of the number of components between the regular schematic and the parasitic netlist.

TABLE I
NUMBER OF PARASITIC ELEMENTS OF THE 45NM RING OSCILLATOR.

Simulation	Transistors	Capacitors	Resistors	Total
With Parasitics	6	82	19	107
Without Parasitics	6	0	0	6

IV. Fast and Accurate Physical Design Optimization in Nano-CMOS Mixed-Signal Circuits

A. Parasitic-Aware Accurate Physical Design Optimization

The proposed *parasitic-aware* physical design optimization flow is shown in figure 3. An initial physical design is performed once the logical design is complete and meets the specifications. The physical design is then subjected to Design Rule Check (DRC), Layout vs. Schematic (LVS) and parasitic (RLCK) extraction. If the specifications are not met, a parasitic parameterized netlist is then created with the design variables as parameters. *This netlist contains all the parasitics associated with the initial layout while the active devices are parameterized.* The resulting netlist is then used by an automated process to conduct the optimization phase, which is outlined by the dashed line and is discussed in this paper. The final physical layout is then created by using the optimized parameters. The physical layout creation, which is a very time consuming process, needs to be only conducted at most two times using this proposed approach, provided that the perturbation introduced to the layout by the resizing process does not have a large impact on the design behavior. This approach is justified by the results presented below.

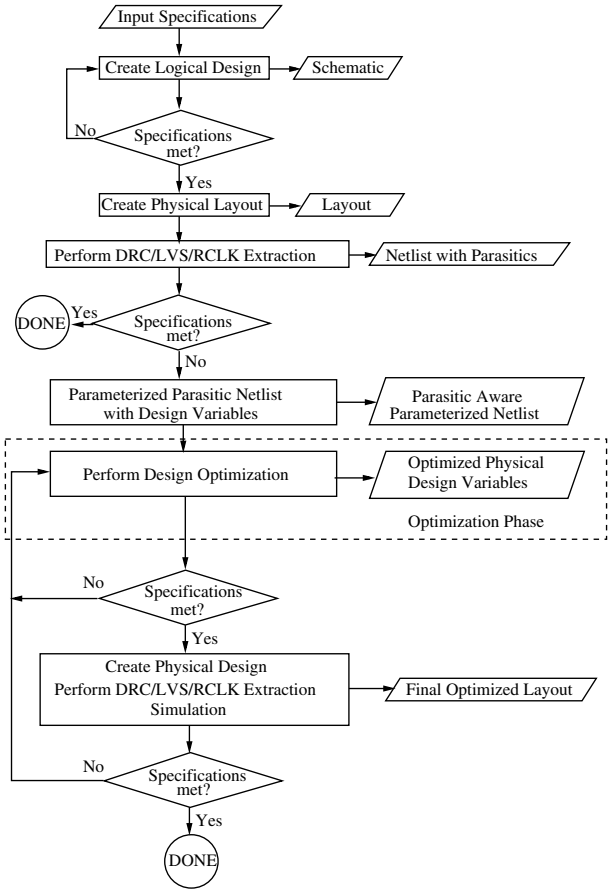

Fig. 3. Parasitic-aware accurate physical design Optimization Flow. This flow performs the mixed-signal design optimization in one layout iteration in which only two manual layouts are needed instead of multiple layouts as is the case of the traditional mixed-signal designs.

The most important phase of the design flow is the optimization algorithm. This paper analyzes three different optimization algorithms: exhaustive search, tabu search and simulated annealing. Each optimization problem uses two variables: W_n and W_p, for the width of NMOS and PMOS, respectively. The optimization is conducted to find a certain output, in the current case frequency (f), with 5% accuracy or better. The selected algorithms are analyzed and compared in terms of the following: (1) run and simulation time, (2) the amount of iterations each algorithm takes for each problem, and (3) accuracy. For the nano-CMOS based mixed-signal circuit under consideration, the objective is to reduce power dissipation and increase oscillating frequency. The optimization algorithms should find the global minimum power and the global maximum frequency. The conflicting objectives are combined to a single figure of merit, namely power to frequency ratio (PFR) which is defined as $PFR = (Power/Frequency)$.

To simplify the study of the algorithms the current paper uses W_n and W_p as the design variables However, the algorithms and the design flow are generic in nature and can accommodate any other design variable that the design engineers intend to use. In the following discussion, all algorithms use S_i for an objective function that holds the best result and S_i^* for a objective function that stores temporary results during the iterative optimal solution search process.

B. Exhaustive Search Optimization Method

The exhaustive search, which is described in Algorithm 1, should be used if the simulation time is not an issue.

Algorithm 1 Exhaustive Search Algorithm for W_n and W_p.

1: Determine the step size $Step$ of each variable between W_{nmax}, W_{nmin} and W_{pmax}, W_{pmin} for N simulations.
2: Initialize the result counter $result\_counter = 0$.
3: **for** ($i = W_{nmin}$ to W_{nmax} with $Step_{Wn}$) **do**
4: **for** ($j = W_{pmin}$ to W_{pmax} with $Step_{Wp}$) **do**
5: $S_{ij} = F(i, j)$
6: Calculate and record minimum and maximum.
7: Calculate the figure of merit under consideration, i.e. power to frequency ratio (PFR).
8: **if** (value is within the limit) **then**
9: $result[result\_counter] = S_{ij}$.
10: $result\_counter = resut\_counter + 1$.
11: **end if**
12: **end for**
13: **end for**
14: **return** $result$: minimum power, maximum frequency, and PFR.

In the current case, taking into consideration the widths of PMOS and NMOS as variables, the exhaustive search was conducted for different amount of iterations. Table I shows the results of the algorithm's performance. For example the simulation time for 10,000 points of this simple circuit the algorithm took approximately 32 hours to run the actual simulation *on the parasitic netlist* and found 42 points with the same minimum and maximum values. Table II shows the optimization data on the metamodel. It is evident from the table that the search time has been reduced dramatically; the same 10,000 point optimization took only 0.46 seconds. Of course the strength of this approach lies in the fact that an accurate metamodel can be generated by intelligent sampling, but this is not the focus of this work. Figure 4 shows the points that the algorithm found.

C. Tabu Search Optimization Method

Tabu search is described in algorithm 2. This is a metaheuristic algorithm that takes a more aggressive approach than most other search approaches. It skips inferior solutions other than the cases when it needs to exit out of the local optimum [6]. This algorithm uses the entire search constrained space and applies the divide and conquer approach [28].

Figure 5 shows the points that the algorithm has sampled to find the value within the search constrained space. The red dot shows the final result for the search for the 9 GHz frequency, which completed in 4 iterations of the algorithm loop and uses 30 sampling points to receive the result.

TABLE II
EXHAUSTIVE SEARCH OPTIMIZATION FOR FREQUENCY OF 10 GHZ WITH AN ACCURACY OF 1%.

Iterations	Points Found	Times	Min Power	Max Frequency	Min PFR
Parasitic Netlist Optimization (Without Metamodel)					
10000	42	32 hours	19.9μW	12.7 GHz	2.18e-15
2500	13	8 hours	19.9μW	12.7GHz	2.18e-15
625	2	2 hours	19.9μW	12.7GHz	2.18e-15
Metamodel Optimization					
1000000	4566	57.01 sec	19.9 μW	12.8 GHz	2.18e-15
250000	1142	21.73 sec	19.9μW	12.8 GHz	2.18e-15
10000	44	0.46 sec	19.9μW	12.7 GHz	2.18e-15
2500	13	0.04 sec	19.9μW	12.7GHz	2.18e-15
625	2	0.02 sec	19.9μW	12.7GHz	2.18e-15

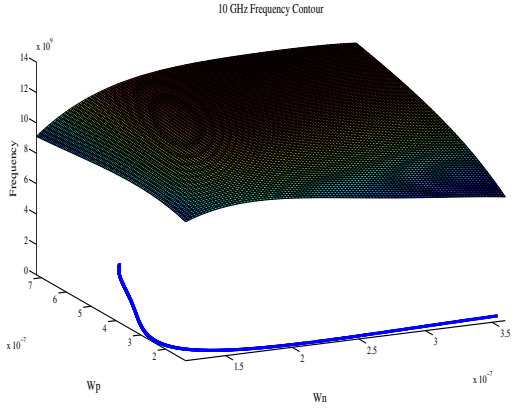

Fig. 4. 1000×1000 Search for 10 GHz design specification of the ring oscillator. The top surface is the golden surface for the complete design space exploration. The bottom curve is the search result.

Algorithm 2 Tabu search algorithm for W_n and W_p.

1: Initialize iteration counter $Counter = 0$.
2: Perform design of experiments analysis for W_n and W_p.
3: Generate initial feasible solution S_i.
4: **while** ($Counter < Max\_Counter$) **do**
5: Generate the next feasible solution S_i^*.
6: $Counter = Counter + 1$
7: **if** (S_i is not visited in the previous iterations) **then**
8: **if** (S_i^* is better solution than S_i) **then**
9: **if** (result is found) **then**
10: break while loop.
11: **end if**
12: $S_i = S_i^*$.
13: **else**
14: Discard the Solution S_i^*.
15: **end if**
16: **end if**
17: **end while**
18: **return** Result or mid point of S_i: minimum power, maximum frequency, and PFR.

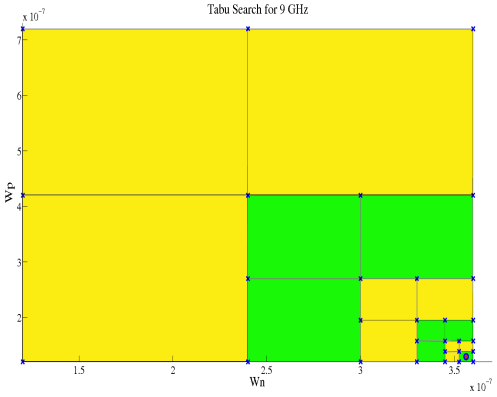

Fig. 5. Tabu search optimization algorithm for 9 GHz. The search space is recursively divided into rectangles and each time the rectangle with superior result is selected, the other 3 with inferior solutions are discarded.

TABLE III
TABU SEARCH OPTIMIZATION FOR FREQUENCY.

Number of Simulations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
32	9 GHz	9.38 GHz	4.22%	6.25 min
7	9.5 GHz	9.4 GHz	1.05%	1.37 min
12	10 GHz	9.94 GHz	0.62%	2.34 min
18	10.5 GHz	10.5 GHz	0.32%	3.52 min
10	11 GHz	11.1 GHz	0.84%	1.95 min
19	11.5 GHz	11.4 GHz	0.71%	3.71 min
30	12 GHz	11.8 GHz	1.92%	5.86 min
4	12.5 GHz	12.6 GHz	0.96%	0.78 min
Metamodeling Optimization				
30	9 GHz	9.4 GHz	4.41%	8.6 ms
7	9.5 GHz	9.41 GHz	0.94%	6.05 ms
12	10 GHz	9.93 GHz	0.74%	7.18 ms
24	10.5 GHz	10.5 GHz	0.32%	7.38 ms
10	11 GHz	11.1 GHz	0.84%	6.41 ms
19	11.5 GHz	11.4 GHz	0.71%	7.11 ms
30	12 GHz	11.8 GHz	1.92%	9.3 ms
4	12.5 GHz	12.6 GHz	0.96%	6.3 ms

D. Simulated Annealing Search Method

Simulated annealing optimization is an extension of the Monte Carlo algorithm and simulates the annealing process [29]. Hence it has a random component; two successive runs will produce different results. The steps of simulated annealing based search are presented in algorithm 3.

Figure 6 shows the algorithm in action as it searches the constraint space. We chose the starting point to be in the middle. From table IV it is observed that the algorithm reaches the first optimal solution in 10 iterations which is within 0.48% of the needed result within 0.77 ms.

V. METAMODELING-BASED OPTIMIZATION FOR SIMULTANEOUS SPEED AND ACCURACY

We propose to substitute the "Optimization Phase" step in figure 3, contoured by the dashed line, with the steps shown

Algorithm 3 Simulated annealing algorithm for W_n and W_p.

Initialize iteration counter $Counter = 0$.
Initialize first feasible solution S_i=F(mid(W_n),mid(W_p)).
Determine initial $Cost_i$ for the solution S_i.
Initialize temperature T as T_i.
while ($Cost$ is varying) **do**
  $Counter$ =Maximum number of iterations.
  while ($Counter > 0$) **do**
    Generate random transition from S_i to S_i^*.
    if (S_i^* is acceptable solution) **then**
      $result = S_i^*$
      break both while loops.
    else
      Calculate change as: $\Delta Cost = Cost_S - Cost_i^*$.
      if ($\Delta Cost < 0$ random(0,1) $< e^{\frac{\Delta Cost}{T}}$) **then**
        Update the solution with new solution: $S \leftarrow S_i^*$.
      end if
    end if
    $Counter = Counter - 1$.
  end while
  Decrease temperature as: $T = T \times Cooling\_Rate$.
end while
return $result$: min. power, max. frequency, and PFR.

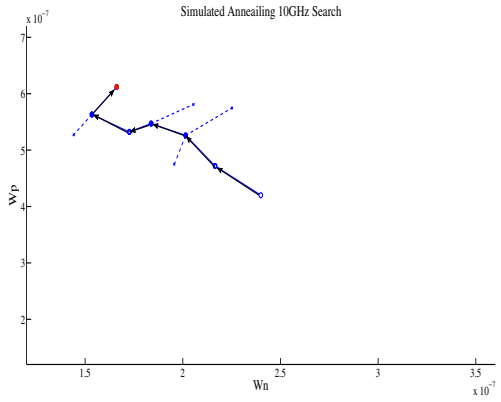

Fig. 6. Simulated annealing for 10 GHz objective. The search progressed in the direction of the arrow.

in figure 7. We used a polynomial metamodel:

$$y = \sum_{i,j=0}^{k} (a_{ij} x_1^i x_2^j), \tag{1}$$

where y is the response (frequency f, power P, or PFR). $x = [W_n, W_p]$ is the vector of design variables. aij are the coefficients determined by the polynomial regression which we chose to be with $k = 4$. From our previous study we chose the LHS sample model of the circuit with 50 sampling points to create an accurate metamodel with root mean square error (RMSE) of 20 MHz with the highest variance for error of 10.7 MHz. With the output frequency being the 10 GHz range, the accuracy of the metamodel is around 99.8%.

Once the metamodel is created, designers can conduct more extensive optimization of the circuit and use the same metamodel for different criteria to be optimized. As can be seen from the example in section IV, the simulation time is a lot more time consuming than using the metamodel.

TABLE IV
SIMULATED ANNEALING OPTIMIZATION FOR FREQUENCY.

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization (Without Metamodel)				
35	9 GHz	8.97 GHz	0.33%	6.84 min
14	9.5 GHz	9.44 GHz	0.63%	2.73 min
15	10 GHz	10.07 GHz	0.31%	2.93 min
24	10.5 GHz	10.40 GHz	0.97%	4.69 min
16	11 GHz	10.96 GHz	0.36%	3.12 min
5	11.5 GHz	11.46 GHz	0.34%	0.98 min
3	12 GHz	11.99 GHz	0.08%	0.59 min
10	12.5 GHz	12.47 GHz	0.24%	1.95 min
Metamodeling Optimization				
32	9 GHz	8.96 GHz	0.48%	1.8 ms
18	9.5 GHz	9.41 GHz	0.94%	1.05 ms
10	10 GHz	10.05 GHz	0.48%	0.77 ms
19	10.5 GHz	10.40 GHz	0.96%	1.16 ms
13	11 GHz	10.95 GHz	0.49%	0.85 ms
4	11.5 GHz	11.48 GHz	0.22%	0.38 ms
2	12 GHz	11.98 GHz	0.16%	0.16 ms
12	12.5 GHz	12.42 GHz	0.63%	0.95 ms

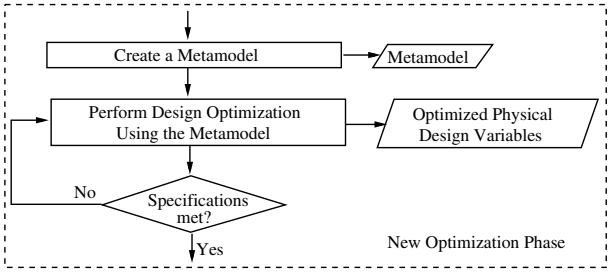

Fig. 7. New optimization phase that replaces optimization phase of figure 3.

The three algorithms explored in section IV are now compared in terms of their running time when they performed optimization with and without the metamodel. The lower half of each of the results presented in Tables II, III, and IV has the running time information. The comparative results are presented in figure 8 in order to provide a visual perspective. The exhaustive search based optimization with metamodel is 2650× faster than without metamodel (on the actual circuit). The tabu-search based optimization with metamodel is 390× faster than without metamodel. The simulated-annealing based optimization with metamodel is 8900× faster than without metamodel. For the **optimization without metamodels the tabu-search based optimization is faster** by 1077× than the exhaustive search and 3.8× faster than the simulated-annealing based optimization. For the **optimization with metamodels the simulated annealing based optimization is faster** by 951× than the exhaustive search and 6× faster than the tabu-search based optimization.

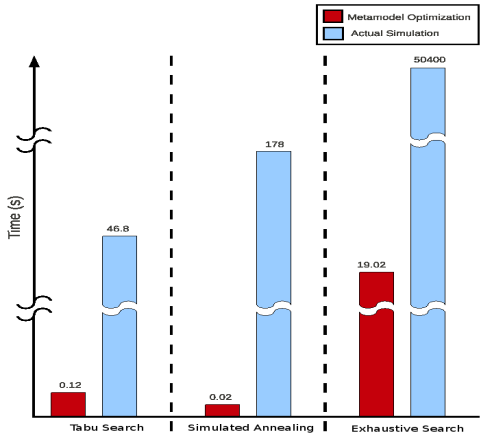

Fig. 8. Comparison of the running time of the three algorithms investigated.

VI. Conclusions and Future Research

A novel design flow using metamodels is proposed to minimize the amount of time of a circuit's optimization. The fast analysis of three different optimization algorithms demonstrated that simulated annealing and tabu search compute the result with fewer iterations than exhaustive search, even though they both stop computing on the first available result within the search criteria, while the exhaustive search computes all available results. It is observed that the metamodeling approach is competitively close to the one time optimization in the number of iterations and the accuracy of the result. Optimization techniques that are not computationally tractable, such as exhaustive search can be used on a metamodel. If the designers need to optimize the circuit more than once or need to generate all the optimal solutions for the problem, the metamodeling approach dramatically reduces the design optimization time. Future research will include a more complex analog nano-CMOS circuit metamodel optimization, with a larger number of variables.

References

[1] E. Y. Chou and B. Sheu, "System-on-a-Chip Design for Modern Communications," *IEEE Circuits and Devices Magazine*, vol. 17, no. 6, pp. 12–17, Nov 2001.

[2] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig, "Design of Mixed-Signal Systems-on-a-Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1561–1571, Dec 2000.

[3] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided Design of Analog and Mixed-Signal Integrated Circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.

[4] F. Wang, C. Nicopoulos, X. Wu, Y. Xie, and N. Vijaykrishnan, "Variation-aware task allocation and scheduling for MPSoC," in *Proceedings of the International Conference on Computer-Aided Design*, 2007, pp. 598–603.

[5] K. Srinivasan and K. S. Chatha, "Integer linear programming and heuristic techniques for system-level low power scheduling on multiprocessor architectures under throughput constraints," *Integration*, vol. 40, no. 3, pp. 326–354, 2007.

[6] S. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra, *Low-Power High-Level Synthesis for Nanoscale CMOS Circuits*. New York, NY 10013, USA: Springer Science+Business Media LLC, 2008.

[7] T. Binder, C. Heitzinger, and S. Selberherr, "A study on global and local optimization techniques for tcad analysis tasks," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 814 – 822, june 2004.

[8] M. Aguirre, J. Chavez, A. Torralba, and L. Franquelo, "Analog design optimization by means of a tabu search approach," in *Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on*, vol. 1, 30 1994, pp. 375 –378 vol.1.

[9] P. Maulik and L. Carley, "Automating analog circuit design using constrained optimization techniques," in *Computer-Aided Design, 1991. ICCAD-91. Digest of Technical Papers., 1991 IEEE International Conference on*, 11-14 1991, pp. 390 –393.

[10] K.-T. Fan, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. London: Chapman and Hall/CRC, 2006.

[11] L. Rodoni, F. Ellinger, and H. Jackel, "Ultrafast cmos inverter with 4.7 ps gate delay fabricated on 90nm soi technology," *Electronic Letters*, vol. 40, no. 20, pp. 1251–1252, Sept. 2004.

[12] T. Weigandt, B. Kim, and P. Gray, "Analysis of timing jitter in cmos ring oscillators," in *Proc. Int. Symp. Circuits Systems*, 1994, pp. 4.27–4.30.

[13] S. D. Boyd and S. J. Kim, "Geometric programming for circuit optimization," in *Proc. Int. Symp. Physical Design, ISPD*, 2005, pp. 44–46.

[14] Y. Xu, L. T. Pileggi, and Bo, "ORACLE: Optimization with resource of analog circuits including layout extraction," in *Proc. Des. Aut. Conf., DAC*, 2004, pp. 151–154.

[15] M. d. M. Hershenson, S. S. Mohan, S. D. Boyd, and T. H. Lee, "Optimization of inductor circuits via geometric programming," in *Proc. Des. Aut. Conf., DAC*, 1999, pp. 994–998.

[16] S. DasGupta and P. Mandal, "An automated design approach for CMOS LDO regulators," in *Proc. Asia South pac. Des. Aut. Conf.*, 2009, pp. 510–515.

[17] Y. Xu, K.-L. Hsiung, X. Li, I. Nausieda, S. Boyd, and L. Pileggi, "OPERA: OPtimization with Ellipsoidal uncertainty for Robust Analog IC design," in *Proc. Des. Aut. Conf., DAC*, 2005, pp. 632–637.

[18] A. Agarwal, G. Wolfe, and R. Vemuri, "Accuracy driven performance macromodeling of feasible regions during synthesis of analog circuits," in *Proc. Great Lakes Symp. VLSI, GLSVLSI*, 2005, pp. 482–487.

[19] A. Agarwal and R. Vemuri, "Hierarchical performance macromodels of feasible regions for synthesis of analog and RF circuits," in *Proc. Des. Aut. Conf., DAC*, 2005, pp. 429–435.

[20] M. Ding and R. Vemuri, "Efficient analog performance macromodeling via sequential design space decomposition," in *Proc. Int. Conf. VLSI Des., VLSID*, 2006, pp. 553–556.

[21] S. Basu, B. Kommineni, and R. Vemuri, "Variation-aware macromodeling and synthesis of analog circuits using spline center and range method and dynamically reduced design space," in *Proc. Int. Conf. VLSI Des., VLSID*, 2009, pp. 433–438.

[22] A. Doboli and R. Vemuri, "Exploration-based high-level synthesis of linear analog systems operating at low/medium frequencies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 11, pp. 1556–1568, November 2003.

[23] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operationa amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, February 2003.

[24] A. Agarwal and R. Vemuri, "Layout-aware RF circuit synthesis driven by worst case parasitic corners," in *Proc. Int. Conf. Comp. Des., ICCD*, 2005, pp. 444–449.

[25] A. Pradhan and R. Vemuri, "A layout-aware analog synthesis procedure inclusive of dynamic module geometry selection," in *Proc. Great Lakes Symp. VLSI. GLSVLSI*, 2008, pp. 159–162.

[26] ——, "Efficient synthesis of a uniformly spread layout aware pareto surface for analog circuits," in *Proc. Int. Conf. VLSI Des.*, 2009, pp. 131–136.

[27] S. Kang and Y. Leblebici, *CMOS Digital Inegrated Circuits*, 3rd ed. New York: McGraw Hill, 2003.

[28] L. Guerra, M. Potkonjak, and J. Rabaey, "Divide-and-conquer techniques for global throughput optimization," in *VLSI Signal Processing, IX, 1996., [Workshop on]*, 30 1996, pp. 137 –146.

[29] N. Metropolis, A. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, p. 1087, 1953.