Bee Colony Inspired Metamodeling Based Fast Optimization of a Nano-CMOS PLL

Oleg Garitselov[1], Saraju P. Mohanty[2], Elias Kougianos[3], and Priyadarsan Patra[4]
Nano-Systems Design Laboratory (http://nsdl.cse.unt.edu)[1,2,3]
University of North Texas, Denton, USA.[1,2,3]
Intel Architecture Group, Intel Corporation, USA.[4]
Email-ID: omg0006@unt.edu[1], saraju.mohanty@unt.edu[2],
eliask@unt.edu[3], and priyadarsan.patra@intel.com[4].

Abstract— **The design and optimization complexity of analog/mixed-signal (AMS) components causes significant increase in the design cycle as the technology progresses towards deep nanoscale. This paper presents a two-tier approach to significantly reduce the design cycle time by combining accurate metamodeling and intelligent optimization. The paper first presents metamodeling which is a surrogate model of a parasitic-aware SPICE model of the circuit in order to simplify the optimization calculations and minimize the design space exploration time. The paper then introduces the Bee Colony Optimization (BCO) algorithm for nano-CMOS AMS circuit optimization. To best of the authors' knowledge, this is the first research combining metamodel and BCO for AMS design space exploration. The proposed design optimization flow is used on 5 metamodels with 21 design parameters each, corresponding to 5 distinct Figures of Merit (FoMs) to conduct multi objective optimization. A 180 nm LC-VCO PLL frequency generation circuit is used as case study. The optimization achieved approx. 90% power and 52% jitter reduction while keeping locking time constraints on the system. In comparison to an exhaustive simulation approach, metamodeling is 10^{20} times faster.**

I. Introduction and Contributions

Analog/Mixed-Signal (AMS) system design is a complex and time consuming process especially at the physical-design level. The optimization at this level is intensive and even infeasible as it is hard to predict the output of an actual circuit due to the complexity of computations involved in simulation. In addition, the presence of parasitics after the physical layout stage has a very dramatic effect on the output, hence making the numerical methods inefficient [1], [2]. For example, the simulation time for PLL lock on a full parasitic netlist is of the order of many hours to days. Thus, there is a pressing need for design methodologies that provide: (1) Fast simulation of nano-CMOS AMS systems and circuits for verification and characterization. (2) Fast layout optimization of complex nano-CMOS AMS systems. (3) Fast design space exploration and optimization convergence to reduce design cycle time in the current short time-to-market constraints.

To address the complexity of design optimization, numerous research works have been presented in the current literature which deal with the actual circuit netlist. Design space exploration approaches from high level descriptions of analog circuits are given in [3]. The use of neural networks in the automatic synthesis of op-amps is explored in [4]. In [5], a parasitic-aware LC-VCO is presented. Low-power LC-VCOs are presented in [6]. In [7], a current-controlled oscillator is subjected to process variations. The process mismatch of an ADC is discussed in [8]. A layout-aware modeling approach for analog synthesis is given in [9]. A single manual design iteration design flow is proposed in [1] for fast design optimization of VCOs.

An alternative approach is to use surrogate models of the actual circuit (metamodels) or simplified models (macromodels) for fast simulation, and design space exploration. A statistical wire-length estimation approach using surrogate modeling is proposed in [10]. A VCO parametric metamodeling approach is given in [11]. Posynomial modeling for gate sizing is done in [12]. Metamodeling is presented for IP reuse for SoC iteration and microprocessor design in [13]. Metamodeling is used in [14] for creating an inductor for CMOS circuits. In [15] support vector machine (SVM)-based machine learning is proposed as a surrogate for expensive circuit-level simulation. In [16], metamodeling has been used for small circuit optimization. Macromodeling is discussed in [17], [18], but it is not metamodeling since metamodels are continuous predictive equations and not simplified circuits.

A metamodel is essentially a predictive mathematical formula for a given figure of merit (FoM) such as power, frequency, jitter, leakage, phase noise, etc. Each circuit can obviously have more than one metamodel if the optimization step is multi objective , which are then used by the optimization algorithm to bring the circuit to the needed specifications. During the optimization phase the savings by using the mathematical models are enormous, since sampling and recreation of the physical design of the circuit is not needed for each iteration. Another advantage of using metamodels is that they are reusable and *language and tool independent*, hence IP reuse speeds up the process if the designer has to create multiple designs that have close specifications. In other fields the metamodeling process is used consistently especially when the sampling is very costly or time consuming [19].

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

To fill the gap of the existing approaches to meet the need for shorter design cycle for nanoscale AMS circuits, this paper presents following **novel contributions** to the state-of-the art:

- Approaches to create accurate, parasitic-aware metamodels for complex nanoscale AMS circuits such as a PLL, are presented. Accurate models are created for 21 design parameters from only 100 SPICE-level simulations.
- The Bee Colony Optimization (BCO) algorithm is investigated for the first time for AMS circuit optimization. The multi-objective optimization results in the desired physical design the PLL, demonstrating the effectiveness of this approach.
- It is demonstrated that the BCO assisted, metamodeling based design flow is orders of magnitude faster than circuit-based approaches.

The rest of the paper is organized as follows: A brief overview of the PLL circuit that was used in this research is given in Section II. Section III describes the BCO algorithm. Section IV introduces and describes the metamodeling design flow. The optimization results are shown in Section V. The paper is concluded with directions for future research in Section VI.

II. Phase Locked Loop (PLL) Circuit

The Phase Locked Loop (PLL) shown in Fig. 1 is a closed loop feedback control system consisting of the phase detector, charge pump, loop filter, LC-VCO, and frequency divider. The logical and physical design of the PLL was performed for a 180 nm CMOS technology with a target frequency of 2.6 GHZ. The characterization of the target FoMs is presented in Table I. The following sections will briefly explain each component of the PLL system.

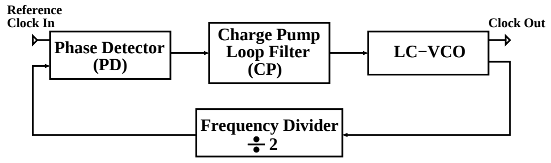

Fig. 1. Block diagram of a PLL.

A. Phase Detector

The phase detector enables the phase differences in the loop to be detected and the resultant error voltage to be produced. A proportional phase detector directs the charge pump to supply charge amounts in proportion to the phase error detected. Phase detectors range from a simple XOR gate to complex logic circuit consisting of flip-flops. Phase detector is an analog mixer [20] or an asynchronous sequential logic circuit functioning so as to detect mismatch between phase or frequency between two signals.The schematic representation of the phase detector using two D flip-flops and one AND gate is shown in Fig. 2.

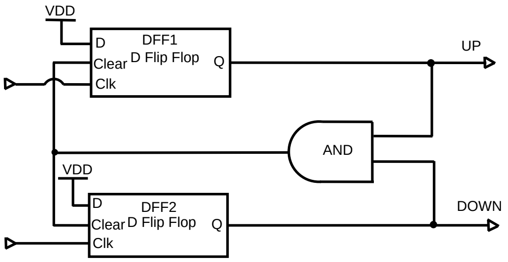

Fig. 2. Phase detector circuit.

B. Loop Filter and Charge Pump

The charge pump schematic is shown in Fig. 3. It stabilizes spurious fluctuations of currents and switching time to minimize the spurs in the VCO input.

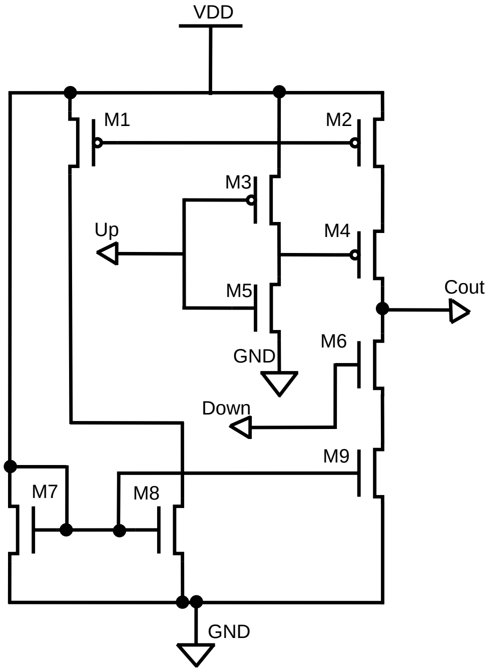

Fig. 3. Schematic of the charge pump circuit.

The output signal from the charge pump is applied to the loop filter, which is shown in Fig. 4. The loop filter determines the PLL's dynamic characteristics. A low-pass RC filter is used to pass frequency signals within the range of the VCO.

C. LC Voltage Controlled Oscillator

The LC-tank voltage-controlled oscillator (LC-VCO) for this design is shown in Fig. 5.The operating frequency of the LC-VCO can be mainly controlled by applying a DC input voltage.

D. Frequency Divider

The frequency divider shown in Fig. 6 is implemented using true single phase logic. When a continuous train of pulse waveforms at fixed frequency is fed to it as an input signal,

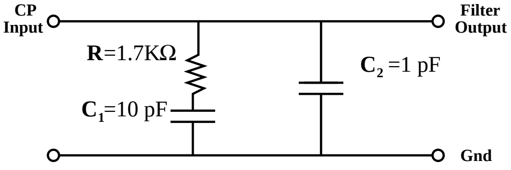

Fig. 4. Schematic of the loop filter circuit.

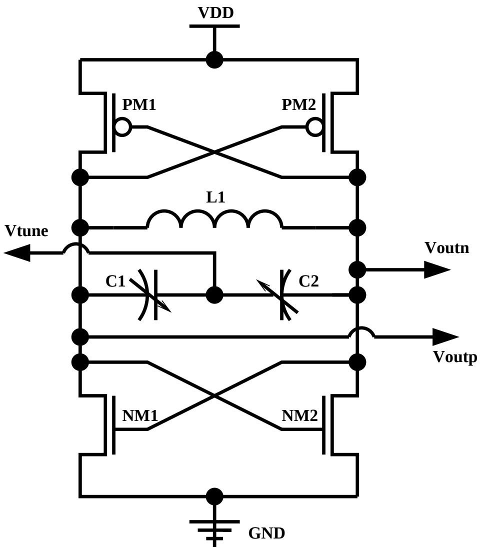

Fig. 5. Schematic diagram of the LC-VCO.

an output signal of approximately half the frequency of the input signal can be obtained.

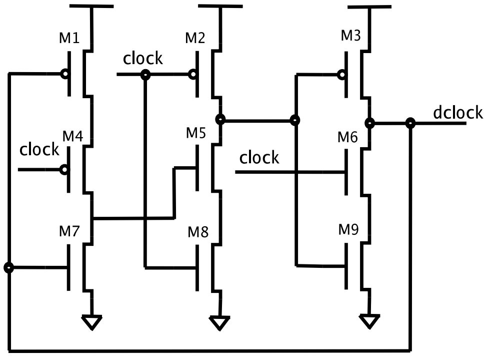

Fig. 6. Schematic of divide by 2 circuit.

III. Bee Colony Optimization Algorithm

The two crucial phases in a metamodel assisted design is the creation of the metamodels and the optimization algorithm. In this section we concentrate on the optimization algorithm and defer discussion of the metamodel generation to Section IV.

The bee colony approach has been used for job shop scheduling optimization in [21]. A powerful and efficient algorithm for numerical function optimization, the artificial bee colony algorithm, was proposed in [22]. A multi-objective heuristic algorithm which is based on bee colony process is proposed in [23]. A bee colony optimization metaheuristic algorithm is introduced in [24] and applied to different transportation problems. Multiple heuristics for bee colony algorithm have been proposed in [25] and were used in multi-variable function optimizations. In this paper the algorithm is investigated for nanoscale PLL design as the parasitic-aware netlist simulation time is excessive for efficient design space exploration.

The BCO algorithm is based on the natural behavior of honey bees for finding the best food source. The artificial bee colony divides bees into three categories: onlookers, scouts, and workers. The algorithm starts with bees being divided equally between onlookers and worker bees only. An initial random solution is assigned to worker bees. Worker bees then search for food at the known random location. When bees return to the hive, the information is shared among the bees by performing a wiggly dance on the hive floor. The unemployed onlooker bees then choose the best food source and employ themselves to go search for more food around the area of the food source. The worker bees then become scout bees and start searching for food randomly again.

The internals of the meta-heuristic BCO algorithm which is used for fast design space exploration of the PLL in this paper is described in Algorithm 1. The proposed algorithm is a maximization approach. Due to the random behavior of bees it can leave local maxima and potentially find the global maximum, provided a sufficiently large number of iterations is performed. The Figures-of-Merit (FoMs) used in this algorithm are presented in Section V.

IV. The Proposed Bee-Colony Inspired Metamodel-Based Design Flow

In order to create accurate metamodels, the designer needs to take into account how many design parameters are in the systems, and what is the maximum number of samples that will be used to create the metamodel. The sampling stage is the slowest part of the metamodeling process. The accuracy of the metamodel is dependent on the amount of simulations which is limited by the available simulation budget (time-wise). The final accuracy also depends on the form of the metamodel. In this paper we consider polynomial models hence the accuracy is directly related to the maximum number of coefficients that can be fit in the model. To maximize the accuracy of the model the right sampling technique should be identified. Following [26], we use Latin Hypercube Sampling (LHS) in this paper.

For the PLL circuit under study, the sampled data is fit into partial polynomial equations. Since the full polynomial function would result in a very large amount of coefficients for 21 variables, partial polynomial functions of order 1 through 6 are considered. A stepwise regression method [27] is used to filter out the coefficients that do not contribute to the function's outcome. Stepwise regression starts with an initial model and then compares the explanatory power of incrementally larger and smaller models. At each step, the P-value of an F-statistic is computed to test the model with and without a potential

Algorithm 1 Proposed Bee Colony Optimization Algorithm.

```
**Initialize** maximum iterations $\leftarrow max_i$.
**Set** the boundaries for each parameter $P(i) \leftarrow [min, max]$.
$NumberBees \leftarrow$ **Define** the number of bees.
$buffer \leftarrow$ Number of close worker bees dispersal.
**Initialize** a matrix as follow: $bee_{matrix}(3, NumberBees) \leftarrow [workers, onlookers, scouts]$.
**Set** $bee_{matrix}$ first half to be workers and other onlookers.
**Initialize** food sources.
**while** $(Counter \leq max_i)$ **do**
    **for** each $i$ from 1 to $NumberBees$ **do**
        **if** $(bee_{matrix}(1, i) == 1)$ **then**
            (1) **Send** worker bee to a random known food source.
            **Calculate** Power(i), Jitter$_{h/v}$(i) using metamodels.
            **Calculate** the proposed FoM of the PLL.
            **if** (current FoM is better than the previous FoM) **then**
                **Update** result and location.
            **else**
                **Convert** bee to onlooker.
            **end if**
        **else**
            **if** $(bee_{matrix}(1, i) == 1)$ **then**
                (2) **Send** onlooker bee.
                **Calculate** probability that the food source is good
                **if** (probability is high) **then**
                    **Send** onlooker to random location for each design parameter $P$.
                    Calculate the FoM.
                    **if** (current FoM is better than the previous FoM) **then**
                        **Update** result and location.
                        **Convert** bee to worker.
                    **else**
                        **Convert** bee to scout.
                    **end if**
                **end if**
            **else**
                (3) **Send** scout bee.
                **Pick** the best result as $best_r$.
                **Send** the scout to random location for each $P$.
                **if** (current FoM is better than the previous FoM) **then**
                    **Update** the result.
                    **Convert** bee to worker.
                **end if**
            **end if**
        **end if**
        **if** (current FoM is better than previous FoM) **then**
            **Update** result and location.
        **end if**
    **end for**
    $Counter \leftarrow Counter + 1$.
**end while**
**Return** result and location.
```

term. If a term is not currently in the model, the null hypothesis is that the term would have a zero coefficient if added to the model. If there is sufficient evidence to reject the null hypothesis, the term is added to the model otherwise, if a term is currently in the model, the null hypothesis is that the term can be ignored, so if there is insufficient evidence to reject the null hypothesis, the term is removed from the model. The method concludes when no more improvements can be made to the model. Stepwise regression may build different models from the same set of potential terms depending on the terms that were initially included in the model which changes the order in which terms are moved in and out.

A. *Design Parameter Selection and Their Ranges*

The design parameters are selected among the key components of the PLL circuit. LC-VCO transistor NM1 and NM2 (Fig. 5) widths are set to W_{nLC} and PM1 and PM2 widths are set to W_{pLC}. The divider (Fig. 6) transistor widths are parameterized each separately: W_{n1Div} for M5, W_{n2Div} for M6, W_{n3Div} for M7, W_{n4Div} for M8, W_{n5Div} for M9, W_{p1Div} for M1, W_{p2Div} for M2, W_{p3Div} for M3, and W_{p4Div} for M4. Since the phase detector has too many transistors, this component circuit is parameterized after dividing it into three logical portions. The parameters are distributed between two flip-flops and the AND gate. D-flip-flop DFF1: W_{npd1} and W_{ppd1}, D-flip-flop DFF2: W_{npd2} and W_{ppd2}, and AND gate: W_{npd3} and W_{ppd3} (Fig. 2). The charge pump (Fig. 3) is also divided into two different portions. Since the current mirror transistors need to be larger size than the logic transistors of the circuit, the charge pump inverter 1&2 transistors M1, M2, M3, and M4 are set with W_{nCP1} and W_{pCP1}, and the current mirror transistors M5, M6, M7, and M8 are set to W_{nCP2} and W_{pCP2}. This results in a total of 21 parameters. The ranges for each parameter are shown in Table II.

B. *PLL Circuit Characterization*

The PLL circuit is characterized for output frequency, power, vertical and horizontal jitter (to simplify the phase noise calculations), and locking time. A separate metamodel is created for each FoM from the same sample set. Each single transient simulation calculates all FoMs so the number of simulations that are needed does not depend on the number of metamodels than need to be generated. In general, the more the particular circuit is characterized the better, since the metamodels are reusable and they can be used for different optimizations and verification later without rerunning the simulations.

C. *Selecting the Right Metamodel*

There may be numerous forms of metamodels that can be created from the same sampled set. The Root Mean Square Error (RMSE) and coefficient of determination R^2 are the metrics used for goodness of fit. The RMSE is derived from the sum of square errors (SSE):

$$RMSE = \sqrt{\frac{1}{N} SSE} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(y(x_k) - \hat{y}(x_k)\right)^2}, \quad (1)$$

where N is the number of simulation points, y is the actual simulation result values and $\hat{y}$ are the results of the metamodel at the same location as the simulation point. R^2 predicts the probability that a future result is accurately predicted by the model. R^2 ranges from 0 to 1, where 1 is the best value. However, R^2 cannot account for over-fitting of the model. Hence, the adjusted R^2, R^2_{adj} is used since it accounts for the number of explanatory terms in a model [27]. Both R^2 and R^2_{adj} for different orders of the polynomial metamodel for settling time are shown in Fig. 7. The number of coefficients that are generated for each order of the polynomial metamodel is shown in Fig. 8. In that figure the R^2 value and R^2_{adj} are nearly equal to 1 when the order reaches 5. The number of coefficients that represent the metamodel at those orders is equal to the number of simulation data points (100). This means that the model is over fitted, therefore for the metamodel that represents settling time, a polynomial order of 4 will be used.

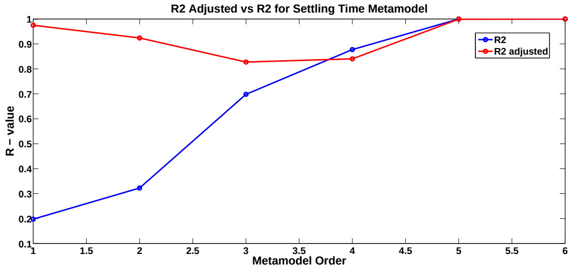

Fig. 7. Generated R^2 and R^2_{adj} for various orders of the polynomial metamodel for settling time.

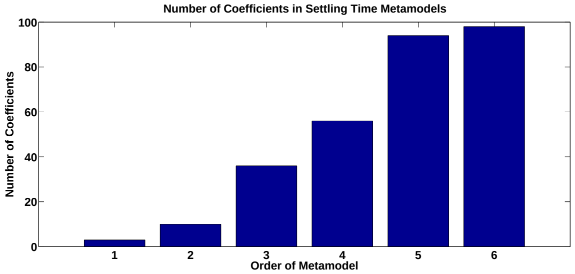

Fig. 8. The number of coefficients corresponding to the order of the generated metamodel for settling time.

V. Metamodel Optimization Results

The proposed optimization is performed on a constraint of locking time. The locking time is a metric that shows that the PLL circuit really works as expected. The optimization target is that the frequency is within 0.5% of the specification. The aim is for a 2.6 GHz output frequency of the PLL. The FoM that needs to be maximized is calculated for every arising combination of parameters varied for optimization. The paper introduced the following FoM for the PLL to ensure that mutual conflicting objectives are met during the optimization:

$$FoM_{\text{PLL}} = \left(\frac{1}{\text{Power} \times \text{Jitter}_h \times \text{Jitter}_v} \right), \qquad (2)$$

where P, J_h, and J_v are the power, horizontal jitter, and vertical jitter, respectively. The maximization of this FoM will lead to a PLL design that will have minimized power and jitter.

The BCO progression over the number of iterations is shown in the Fig. 9. Each run of the optimization is slightly different from the others due to the randomness of the bees involved in searching the solution space.

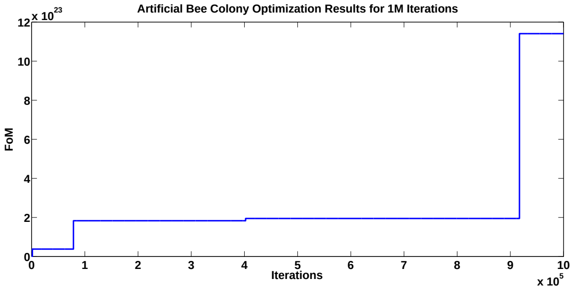

Fig. 9. Results of the BCO algorithm progression for the selected FoM.

The results of the BCO are shown in Table I. The final optimized responses of the PLL are shown in Table II. The final physical design of the PLL that uses the optimized parameters is shown in Fig. 10.

TABLE I
Power and jitter of the PLL before and after optimization.

Metric	Before Optimization	After Optimization	Improvement
Power	9.29 mW	0.87 mW	90.6%
Jitter Vertical	168.35μV	3.28 nV	∼100%
Jitter Horizontal	189 ps	180 ps	4.8%

TABLE II
PLL circuit parameters with the parameter constraints. Final column also shows the optimized values for each parameter.

Circuit	Parameter	Min (m)	Max (m)	Optimal Value (m)
Phase Detector	W_{ppd1}	400n	2μ	1.66μ
	W_{npd1}	400n	2μ	1.11μ
	W_{ppd2}	400n	2μ	784n
	W_{npd2}	400n	2μ	689n
	W_{ppd3}	400n	2μ	1.54μ
	W_{npd3}	400n	2μ	737n
Charge Pump	W_{nCP1}	400n	2μ	1.24μ
	W_{pCP1}	400n	2μ	1.35μ
	W_{nCP2}	1μ	4μ	1.35μ
	W_{pCP2}	1μ	4μ	2.88μ
LC-VCO	W_{nLC}	3μ	20μ	18.62μ
	W_{pLC}	6μ	40μ	37.48μ
Divider	W_{p1Div}	400n	2μ	1.65μ
	W_{p2Div}	400n	2μ	1.54μ
	W_{p3Div}	400n	2μ	1.38μ
	W_{p4Div}	400n	2μ	1.96μ
	W_{n1Div}	400n	2μ	1.09μ
	W_{n2Div}	400n	2μ	1.17μ
	W_{n3Div}	400n	2μ	1.29μ
	W_{n4Div}	400n	2μ	1.95μ
	W_{n5Div}	400n	2μ	536n

VI. Conclusion and Future Research

This paper investigated the use of metamodeling and an intelligent Bee Colony Optimization algorithm to speed up the design-space exploration for AMS circuits. On the case

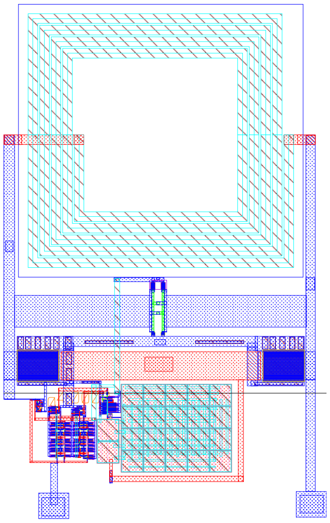

Fig. 10. Final layout of the optimized PLL for 180nm nano-CMOS.

study of a 180nm PLL, the circuit was parameterized with 21 parameters and optimized using the BCO algorithm. The algorithm is proven to be suitable for nanoscale AMS circuit optimization and performed very well in convergence. The final outcome of the design flow was 90% power savings and and average of 52% jitter minimization which have been achieved with a minimal time of 100 simulations to generate polynomial metamodels. In comparison, an exhaustive search of the design space of 21 parameters with 10 intervals per parameter would require 10^{21} simulations. The time savings are enormous ($\approx 10^{20}\times$ simulation time). In future research we will investigate metamodels other than polynomial. Since the accuracy of the metamodels is essential, it would be interesting to see the behavior of different kinds of equations and their accuracy applied to complex circuits with large parameter sets.

References

[1] D. Ghai, S. P. Mohanty, and E. Kougianos, "Design of Parasitic and Process-Variation Aware Nano-CMOS RF Circuits: A VCO Case Study," *IEEE Trans. VLSI Syst.*, vol. 17, no. 9, pp. 1339–1342, 2009.

[2] J. Park, K. Choi, and D. J. Allstot, "Parasitic-Aware Design and Optimization of a Fully Integrated CMOS Wideband Amplifier," in *Proceedings of the Asia South Pacific Design Automation Conference*, 2003, pp. 904–907.

[3] 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.

[4] G. Wolfe and R. Vemuri, "Extraction and Use of Neural Network Models in Automated Synthesis of Operational Amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, February 2003.

[5] M. Chu, D. J. Allstot, J. M. Huard, and K. Y. Wong, "NSGA-based parasitic aware Optimization of a $5GHz$ Low-noise VCO," in *Proceedings of the Asia South Pacific Design Automation Conference*, 2004, pp. 169–174.

[6] K. Kwok and C. H. Luong, "Ultra-low-Voltage high-performance CMOS VCOs using transformer feedback," *IEEE Journal of Solid State Circuits*, vol. 40, no. 3, pp. 652–660, March 2005.

[7] D. Kim and et. al., "CMOS Mixed-Signal Circuit Process Variation Sensitivity Characterization for Yield Improvement," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2006, pp. 365–368.

[8] C. Svensson, S. Andersson, and P. Bogner, "On the power consumption of analog to digital converters," in *Proceedings of the 24th Norchip Conference*, 2006, pp. 49–52.

[9] A. Pradhan and R. Vemuri, "Efficient Synthesis of a Uniformly Spread Layout Aware Pareto Surface for Analog Circuits," in *Proceedings of the 22nd International Conference on VLSI Design*, 2009, pp. 131–136.

[10] J. L. Wong, A. Davoodi, A. Khanderwal, A. Srivastava, and M. Potkonjak, "A Statistical Methodology for Wire-Length Prediction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1327–1336, July 2006.

[11] W. Dong, Z. Feng, and P. Li, "Efficient VCO phase macromodel generation considering statistical parametric variations," in *Proceedings of the IEEE/ACM international conference on Computer-aided design*, 2007, pp. 874–878.

[12] S. Roy and C. C.-P. Chen, "SmartSmooth: A Linear Time Convexity Preserving Smoothing Algorithm for Numerically Convex Data with Application to VLSI Design," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2007, pp. 559–564.

[13] D. A. Mathaikutty and S. K. Shukla, *Metamodeling Driven IP Reuse for System-on-chip Integration and Microprocessor Design*. Artech House, 2007.

[14] A. Lamecki, L. Balewski, and M. Mrozowski, "Towards Automated Full-Wave Design of Microwave Circuits," in *Proceedings of the 17th International Conference on Microwaves, Radar and Wireless Communications*, 19-21 2008, pp. 1–2.

[15] R. Samanta, J. Hu, and P. Li, "Discrete Buffer and Wire Sizing for Link-Based Non-Tree Clock Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, pp. 1025–1035, July 2010.

[16] O. Garitselov, S. P. Mohanty, and E. Kougianos, "Fast optimization of nano-CMOS mixed-signal circuits through accurate metamodeling," in *Proceedings of the 12th International Symposium on Quality Electronic Design*, 2011, pp. 405–410.

[17] 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.

[18] 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.

[19] K.-T. Fan, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. 23-25 Blades Court, London SW15 2NU, UK: Chapman and Hall/CRC, 2006.

[20] Z. Wang, "An Analysis of Charge-Pump Phase-Locked Loops," *IEEE Transactions on Circuit and Systems-I: Fundamental Theory and Applications*, vol. 52, no. 10, pp. 2128–2138, October 2005.

[21] C. S. Chong, M. Y. H. Low, A. I. Sivakumar, and K. L. Gay, "A bee colony optimization algorithm to job shop scheduling," in *Proceedings of the Winter Simulation Conference*, December 2006, pp. 1954–1961.

[22] D. Karaboga and B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm," *J. Global Optim.*, pp. 39:459–471, 2007.

[23] R. Hedayatzadeh, B. Hasanizadeh, R. Akbari, and K. Ziarati, "A Multi-objective Artificial Bee Colony for Optimizing Multi-objective Problems," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010, pp. V5–277–V5–281.

[24] D. Teodorovic, P. Lucic, G. Markovic, and M. D. Orco, "Bee Colony Optimization: Principles and Applications," in *8th Seminar on Neural Network Applications in Electrical Engineering (NEUREL)*, September 2006, pp. 151–156.

[25] M. Salim and M. T. Vakil-Baghmisheh, "Discrete bee algorithms and their application in multivariable function optimization," *Artif. Intell. Rev.*, vol. 35, pp. 73–84, January 2011.

[26] O. Garitselov, S. P. Mohanty, and E. Kougianos, "Nano-CMOS Mixed-Signal Circuit Metamodeling Techniques: A Comparative Study," in *Proc. Int. Symp. Elec. Des.*, 2010, pp. 191–196.

[27] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Inc, 2006.