diagnosing errors in sequential circuits is much more difficult than that in combinational circuits because circuit unrolling is used. For example, the bug trace for the last benchmark has 77 cycles, and it produces an unrolled circuit containing more than one million standard cells. The characteristics of the benchmarks and their results are summarized in Table I. For each benchmark, 32 traces were provided, and the goal was to repair the circuit so that it produces the correct output responses for those traces. Since our algorithm processes all the traces simultaneously, only one iteration will be required. For the computation of more representative runtimes only, we deliberately processed the traces one by one and failed all verification so that all the benchmarks underwent 32 iterations. All the bugs were injected at the Register Transfer Level (RTL), and the designs were synthesized using Cadence RTL compiler 4.10. In the table, "Err. Diag. time" is the time spent on error diagnosis, "#Fixes" is the number of valid fixes returned by CoRé, and "DPS time" is the runtime of DPS. The minimum/maximum numbers of support variables and gates used in the returned fixes are shown under "Resynthesis netlist." Note that implementing any valid fix is sufficient to correct the circuits' behavior, and we rank the fixes based on the logic depth from primary inputs: Fixes closer to primary inputs are preferred. Under "Err. diag. time," "1st" is the runtime for diagnosing the first bug trace, whereas "Total" is the runtime for diagnosing all 32 traces. The comparison between the first and total diagnosis time shows that diagnosing the first trace takes more than 30% of the total diagnosis time in all the benchmarks. The reason is that the first diagnosis can often localize errors to a small number of sites, which reduces the search space of further diagnoses significantly. Since CoRé relies on iterative diagnosis to refine the abstraction of signatures, this phenomenon ensures that CoRé is efficient after the first iteration. As Table I shows, error diagnosis is still the bottleneck of the CoRé framework. We also observe that fixing some bugs requires a large number of gates and support variables in their resynthesis netlists because the bugs are complex functional errors at the RTL.

VI. Conclusion

In this paper, we propose a framework, called CoRé, to correct functional errors in digital circuits relying only on error traces. This framework exploits both satisfiability and observability DCs, and it uses an abstraction-refinement scheme to achieve better scalability. To support the resynthesis task required in the framework, we propose an encoding of resynthesis information, called PBDs, and use it in our innovative resynthesis techniques. Because CoRé does not rely on specific error models, it offers more error-correction capabilities than many previous solutions. The experimental results show that CoRé can produce a modified netlist which eliminates erroneous responses while maintaining correct responses. In addition, CoRé supports combinational and sequential error repairs, and it can be easily adopted in most verification flows.

References

[1] M. F. Ali, A. Veneris, S. Safarpour, R. Drechsler, A. Smith, and M. Abadir, "Debugging sequential circuits using Boolean satisfiability," in *Proc. ICCAD*, 2004, pp. 44–49.

[2] K.-H. Chang, I. L. Markov, and V. Bertacco, "Fixing design errors with counterexamples and resynthesis," in *Proc. ASPDAC*, 2007, pp. 944–949.

[3] N. Eén and N. Sörensson, "An extensible SAT-solver," in *Proc. Theory Appl. Satisfiability Testing, (SAT)*, 2003, pp. 502–518.

[4] A. Mishchenko, J. S. Zhang, S. Sinha, J. R. Burch, R. Brayton, and M. Chrzanowska-Jeske, "Using simulation and satisfiability to compute flexibilities in Boolean networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 5, pp. 743–755, May 2006.

[5] R. Rudell and A. Sangiovanni-Vincentelli, "Multiple-valued minimization for PLA optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 5, pp. 727–750, Sep. 1987.

[6] S. Sinha, "SPFDs: A new approach to flexibility in logic synthesis," Ph.D. dissertation, Univ. California, Berkeley, CA, May 2002.

[7] A. Smith, A. Veneris, and A. Viglas, "Design diagnosis using Boolean satisfiability," in *Proc. ASPDAC*, 2004, pp. 218–223.

[8] S. Staber, B. Jobstmann, and R. Bloem, "Finding and fixing faults," in *Proc. CHARME*. New York: Springer-Verlag, 2005, vol. 3725, pp. 35–49.

[9] A. Veneris and I. N. Hajj, "Design error diagnosis and correction via test vector simulation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 12, pp. 1803–1816, Dec. 1999.

[10] S. Yamashita, H. Sawada, and A. Nagoya, "SPFD: A new method to express functional flexibility," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 8, pp. 840–849, Aug. 2000.

[11] Y.-S. Yang, S. Sinha, A. Veneris, and R. E. Brayton, "Automating logic rectification by approximate SPFDs," in *Proc. ASPDAC*, 2007, pp. 402–407.

[12] MVSIS. [Online]. Available: http://embedded.eecs.berkeley.edu/Respep/Research/mvsis/

ANN- and PSO-Based Synthesis of On-Chip Spiral Inductors for RF ICs

Sushanta K. Mandal, Shamik Sural, and Amit Patra

***Abstract*—This paper presents an efficient layout-level synthesis approach for RF planar on-chip spiral inductors. A spiral inductor is modeled using artificial neural networks in which the layout design parameters, namely, spiral outer diameter, number of turns, width of metal traces, and metal spacing, are taken as input. Inductance, quality factor (Q), and self-resonance frequency (SRF) form the output of the neural model. Particle-swarm optimization is used to explore the layout space to achieve a given target inductance meeting the SRF and other constraints. Our synthesis approach provides multiple sets of layout parameters that help a designer in the tradeoff analysis between conflicting objectives, such as area, Q, and SRF for a target-inductance value. We present several synthesis results which show good accuracy with respect to full-wave electromagnetic (EM) simulations. Since the proposed procedure does not require an EM simulation in the synthesis loop, it substantially reduces the cycle time in RF-circuit design optimization.**

***Index Terms*—Artificial neural networks (ANNs), layout synthesis, on-chip inductor, particle-swarm optimization (PSO).**

I. Introduction

Continuous growth in wireless-communication systems has stimulated research in low-cost, low-power, and high-performance CMOS RF integrated-circuit (IC) components for system-on-chip solutions. On-chip spiral inductor is one of the major components of the RF ICs that dominates circuit performance. In an RF IC, the operating frequency of on-chip inductors is much lower than the first self-resonance frequency (SRF). For example, a voltage-controlled oscillator (VCO) operating at 2.5 GHz requires an inductor with an SRF of at least 6 GHz with a high-quality factor (Q). To fulfill this type of high-SRF

Manuscript received February 21, 2007; revised May 2, 2007. This paper was recommended by Associate Editor H. E. Graeb.

S. K. Mandal and S. Sural are with the School of Information Technology, Indian Institute of Technology (IIT), Kharagpur 721 302, India (e-mail: skmondal@sit.iitkgp.ernet.in; shamik@sit.iitkgp.ernet.in).

A. Patra is with the Department of Electrical Engineering, Indian Institute of Technology (IIT), Kharagpur 721 302, India (e-mail: amit.patra@ieee.org).

Digital Object Identifier 10.1109/TCAD.2007.907284

requirements in the RF IC design, a long running EM simulation is required. Conventional techniques such as numerical approaches that include solution of algebraic and differential equations are computationally expensive. Other methods like analytical and empirical modeling techniques are time efficient but are not sufficiently accurate.

Artificial neural network (ANN) has emerged as an efficient alternative to these conventional modeling techniques. It is popular due to its capability of learning any arbitrary nonlinear input–output relationship from corresponding data and also because it produces smooth approximation results from discrete data. Once a neural network is trained, the model parameters such as weights and biases remain fixed. Thus, the relationship between the model input and output becomes a closed-form expression, and the trained network gives an almost instant output due to low latency. Neural models are, therefore, much faster than physics/EM models and have a higher accuracy than analytical and empirical models. Furthermore, they are easy to develop for a new device or technology [1], [2].

The optimization goals while designing spiral inductors depend on the application. Several efforts have been made for finding the optimal inductor-layout geometries that enhance the quality factor at a particular operating frequency and inductance value within a given design space. The most common approach is the enumeration technique [3] which uses discretized design parameters for simulations and selects the geometry parameters corresponding to the highest Q value for design. Inefficiency of this method becomes apparent with the increase in the number of design variables since the complexity is exponential. Geometric programming has been applied to the inductor-optimization problem in [4] which requires a specific model formulation based on curve fitting. In [5], a sequential-quadratic-programming-based-optimization technique has been developed which improves the speed over enumeration. However, it may get trapped in the local minima. In [6], Q-contour is used for the optimization which is both time consuming and technology-parameter dependent. Different search methods like incremental search [7], binary search [8], and genetic algorithm [9] are used for the optimization that search over all the geometry parameters satisfying a set of constraints.

In this paper, we present an efficient synthesis procedure for the spiral inductor based on the constraints imposed by design specifications. We have developed a multilayer perceptron-based (MLP) ANN model of on-chip spiral inductors and use the particle-swarm-optimization (PSO) algorithm [10]–[13] to explore the layout space. During exploration, the ANN model is used to compute the inductance (L), Q, and SRF of each spiral. The proposed synthesis procedure provides multiple sets of layout parameters for a given inductance value within acceptable error limits. Synthesis results facilitate the designer with more freedom for tradeoff analysis between objectives, such as area, Q, and SRF for inductors.

The rest of the paper is organized as follows: Section II presents the spiral-inductor-synthesis procedure. It also gives a brief overview of ANN modeling and PSO. In Section III, modeling and synthesis results are discussed. Finally, the conclusions are drawn in Section IV.

II. Spiral-Inductor Synthesis

A. ANN-Model Development

Multilayer perceptron feedforward network is one of the most effective and widely used neural network structures. Typically, it consists of an input layer, one or more hidden layers, and an output layer. The neurons in each layer are connected to those in the next layer by weighted edges. We consider four inductor-layout parameters, namely, outer diameter (d), number of turns (N), metal width (W), and spacing between metal traces (s), as the input to the neural model.

TABLE I
Range of Input Parameters

	d (μm)	W (μm)	N	s (μm)
Min	100	4	2	1
Max	340	32	8	5
Step	20	4	2	2

Technology parameters are not included in the input parameters since, for a given fabrication process, the designer cannot control them. The output neurons represent electrical attributes of the inductor which are L, Q, and SRF. We use two hidden layers with 20 neurons each, hyperbolic-tangent activation function for hidden layers and linear activation function for output neurons.

In order to generate training and testing data sets, planar octagonal spiral inductors were constructed in the range of geometric dimensions shown in Table I for the 0.18-μm CMOS technology. This range covers potential inductor dimensions for a typical wireless-communication application. Based on uniform grid distribution sampling strategy, each input parameter is sampled at an equal interval using the step sizes given in Table I. For this kind of nonlinear problem, further fine graining of data samples is expected to improve the input–output-mapping accuracy but will also increase the number of spirals to be simulated, thus requiring higher training-time investment. Out of all the theoretically possible combinations, we have considered 500 realizable spirals and simulated them using IE3D [14], a full-wave electromagnetic (EM) solver. The inductance and quality factor for each inductor were extracted from the IE3D data as follows:

$$L = \frac{\mathrm{Imag}(1/Y_{11})}{2\pi f} \quad Q = \frac{\mathrm{Imag}(1/Y_{11})}{\mathrm{Real}(1/Y_{11})}. \tag{1}$$

Here, Y_{11} is the input admittance of the two-port Y-parameters. The SRF was measured from the Q plot at the frequency point where the Q-value becomes zero. Although various neural-network-training methodologies like k-fold cross validation which is followed by ensemble analysis could have been done, we use the hold-out method for building the neural model to reduce the total cycle time. There is no perceptible loss in accuracy. Out of 500 spirals, 80% were used for training, and the remaining 20% were used for testing the neural network. We selected training and test data in a regular interval of five units so that both cover the complete range and adequately represent the original inductor behavior. As shown in Table I, the input parameters for building the neural model vary over a wide range. The corresponding output-parameter values of the inductors are also quite different. As a preprocessing step, input and output data were normalized to $[-1\ 1]$ with respect to the minimum and the maximum of the data range by means of linear scaling.

During neural-network training, the weight and bias values are adjusted to minimize the training error which is a measure of the correlation between the ANN-model output and the training data. In this work, we have used the Levenberg–Marquardt method as the training algorithm in MATLAB's neural-network tool [15]. The training error goal was set to 0.001. Further lowering of the error limit reduces the generalization capability of the model. On the other hand, setting it too high would lead to lower mapping accuracy. The learning rate was taken as 0.01. If the learning rate is too large, it leads to oscillation, whereas a very small value results in a longer training time for reaching the same level of accuracy.

B. Particle-Swarm Optimization

In this subsection, we give a brief introduction to PSO. Fig. 1 shows the basic flowchart of PSO. It is an evolutionary technique based on

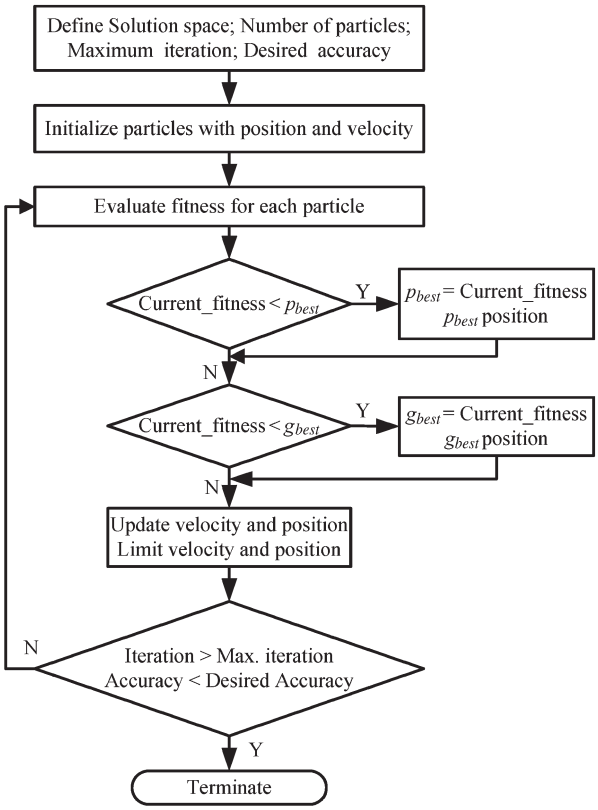

Fig. 1. PSO flowchart.

the social behavior, movement, and intelligence of swarms searching for an optimal location. PSO works on a population of potential solution candidates referred to as particles. Each particle in a swarm is represented by a position and velocity vector.

Like other evolutionary algorithms, PSO uses a fitness function to search for the best position. Each particle is initialized with a random position and velocity. In every simulation run, the fitness function is evaluated by taking the current position of the particle in the solution space. The particles keep track of two best values. The first one is the best fitness value obtained so far by the particle, the corresponding position being termed as personal best ($p_{\rm best}$). The other is the best fitness value achieved so far considering all the particles in the swarm. The location of the best fitness value in a whole swarm is called global best ($g_{\rm best}$). At each run, there is only one $g_{\rm best}$, and all the particles are attracted toward $g_{\rm best}$. In an iteration, particle velocity and position are updated based on $p_{\rm best}$ and $g_{\rm best}$ positions as follows:

$$v_{id}^{n+1} = w \times v_{id}^{n} + c_1 \mathrm{rand1}() * (p_{\rm best}^{n} - x_{id}^{n}) + c_2 \mathrm{rand2}() * (g_{\rm best}^{n} - x_{id}^{n}) \quad (2)$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1}. \quad (3)$$

Here, $d = 1, 2, \ldots, D$; $i = 1, 2, \ldots, Z$; D being the number of design parameters, Z the swarm size, and n the iteration number. The acceleration factors c_1 and c_2 in (2) indicate the relative attraction toward $p_{\rm best}$ and $g_{\rm best}$, respectively. The functions $\mathrm{rand1}()$ and $\mathrm{rand2}()$ generate random numbers that are uniformly distributed between zero and one. To assign equal weight to the relative pulls of $p_{\rm best}$ and $g_{\rm best}$, each of $c_1\mathrm{rand1}()$ and $c_2\mathrm{rand2}()$ was constructed to have an average value of one by making $c_1 = c_2 = 2$. The inertia-weight parameter w controls the tradeoff between the global and local search capabilities of the swarm. We start with a large inertia weight of 1.0 for an initial bias toward the global search and decrease it linearly to a minimum value of 0.4 through different iterations to facilitate more local explorations [16].

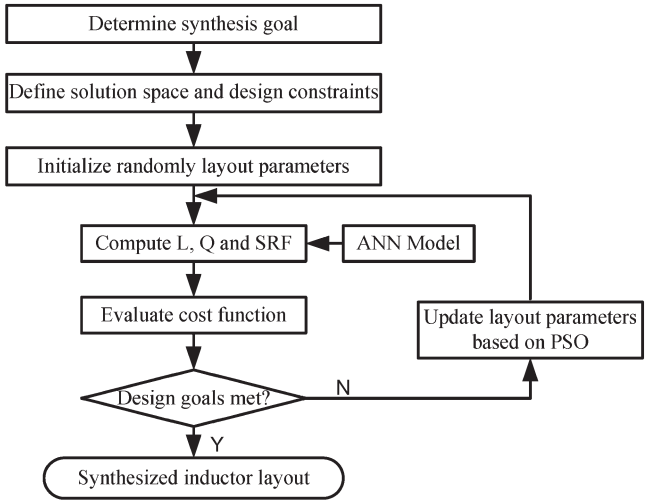

Fig. 2. Proposed spiral-inductor-synthesis methodology.

Another important PSO parameter is the maximum/minimum limit on particle velocity. Without any such limit, particles can go out of the solution space. Since there is no actual mechanism to control the velocity of a particle, an external condition is imposed. Ten percent of a particle's position value is set as the limit of its velocity. A position bound of the particles is also similarly imposed. The number of particles in PSO is less critical than in other population-based algorithms. The number typically varies from 10 to 40 depending on the number of design variables and the complexity of the optimization problem. For our problem, it was found that 30 particles were enough to have good convergence. It may be noted that we use the global best version of PSO. This was considered instead of other variants like local best version to improve the speed of processing. It has been shown in [17] that the PSO formulation of (2) is sufficient for avoiding being trapped in the local minima.

C. Synthesis Methodology

On-chip spiral-inductor synthesis is the process of determining the layout geometric parameters from electrical specifications. As it is obvious, a target-inductance value can be realized by many different combinations of layout parameters. Out of these, only the set of inductor-layout parameters that meet all the design constraints is considered.

We have developed a spiral-inductor-synthesis procedure that helps the designer to make a tradeoff analysis between the competing objectives, namely, Q, SRF, and outer diameter, for a given L. The synthesis flow is shown in Fig. 2. Inductor-synthesis goals are determined by the specific circuit application with the constraints imposed by the design specifications. Our synthesis procedure uses ANN and PSO. The PSO optimizer generates a swarm of particles, each representing a combination of layout parameters in the given design space. The ANN takes each combination of layout parameters and produces L, Q, and SRF as output. Cost function is computed using these electrical parameter values. Particles of the optimizer are then updated according to the minimum cost. This process continues until a desired cost-function objective is achieved or the maximum number of iterations is executed. Usually, the spiral-inductor-design-optimization problem is formulated to maximize the Q value for a target inductance subject to certain constraints. Since, in this synthesis procedure, our aim is to

TABLE II
ANN-MODEL ACCURACY

ANN Structure/ Operating frequency	Training Epochs	Time of Each Epoch (sec)	Type of Data Set	% Average Relative Error			Correlation Coefficients		
				L	Q	SRF	L	Q	SRF
4-20-20-3 1 GHz	25	1.6677	Training	3.9032	1.9651	2.3787	0.9998	0.9996	0.9998
			Test	3.7581	2.1646	1.7599	0.9995	0.9994	0.9995
4-20-20-3 2.5 GHz	31	1.7177	Training	4.5494	0.7956	1.4516	0.9997	0.9996	0.9998
			Test	3.9123	0.7460	1.3853	0.9998	0.9997	0.9998
4-20-20-3 3 GHz	47	1.8314	Training	3.7894	1.1072	1.5295	0.9998	0.9996	0.9999
			Test	2.9622	0.9799	1.6039	0.9998	0.9997	0.9998

find a set of layout parameters which will give the desired inductance value with in acceptable error, the cost function is formulated as

$$\begin{aligned} \text{minimize} \quad & L_{\mathrm{T}} - L_{\mathrm{ANN}} \\ \text{subject to} \quad & N_{\min} \le N \le N_{\max} \\ & d_{\min} \le d \le d_{\max} \\ & W_{\min} \le W \le W_{\max} \\ & s_{\min} \le s \le s_{\max} \\ & d \ge 2N(W+s) - 2s \\ & \mathrm{SRF} \ge \mathrm{SRF}_{\mathrm{given}}. \end{aligned} \tag{4}$$

Here, L_{T}, L_{ANN}, and $\mathrm{SRF}_{\mathrm{given}}$ are the target inductance, the inductance computed from the trained ANN, and the given minimum SRF, respectively. $N_{\min}$, $N_{\max}$, $d_{\min}$, $d_{\max}$, $W_{\min}$, $W_{\max}$, $s_{\min}$, and $s_{\max}$ are the minimum and the maximum bounds of the corresponding optimization variables.

During synthesis, PSO provides multiple solutions of layout parameters for a target-inductance value due to the random initialization of particles and the random variables associated with the velocity- and position-update process. The PSO-search process is terminated if the objective function is less than an acceptable error value or if the number of iterations reaches the maximum. In this paper, for the synthesis of spiral inductors, the error value is set to 0.0001 nH and the maximum number of iterations is taken to be 1000.

III. RESULTS AND DISCUSSION

A. ANN-Model Accuracy

In this paper, we used 400 inductor geometries for training and 100 inductors for testing the neural network. To verify the accuracy of the neural models, statistical measures, such as the average relative error and the correlation coefficient between the outputs and targets were calculated for each output parameter. The average relative error and the correlation coefficients are calculated as follows:

Average Relative Error

$$= \frac{\sum_1^n (x-y)}{ny} \tag{5}$$

Correlation Coefficient

$$= \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}}. \tag{6}$$

Here, n, x, and y are the number of samples in the data set, the ANN-model output, and the corresponding EM simulated value, respectively. The relative error signifies the closeness of the ANN outputs to the EM simulated values. The correlation coefficient is a measure of how closely the neural output fits with the target values. If this number is equal to 1.0, then there is a perfect fit between the targets and the outputs.

TABLE III
SYNTHESIS RESULTS FOR 4-nH SPIRAL INDUCTORS AT 1 GHz WITH DESIGN SPECIFICATIONS: SRF $>$ 6 GHz, $d = 100{-}300$ μm, $W = 8{-}24$ μm, $N = 2{-}6$, AND $s = 1{-}4$ μm

L (nH)	d (μm)	W (μm)	N	s (μm)	Q	SRF (GHz)
3.9999	275	15.1	4.0	2.4	3.6120	10.256
4.0091	288	15.9	3.8	2.1	3.7244	9.7435
3.9969	285	13.0	3.0	1.3	3.3639	11.588
3.9957	238	11.8	4.2	2.0	3.1627	11.477
4.0051	252	11.8	3.9	2.5	3.1153	11.256
4.0008	300	19.5	3.6	1.1	4.2319	9.6278
3.9905	300	18.0	3.4	1.3	4.0410	9.9174
4.0077	300	19.0	3.5	1.0	4.1970	9.7372
4.0000	300	16.4	3.3	1.7	3.8356	10.31
4.0044	283	14.4	3.4	1.9	3.5571	10.839
3.9905	263	11.2	3.3	2.1	3.0323	11.954
4.0044	283	14.4	3.4	1.9	3.5571	10.839

Table II shows the percentage average error and the correlation coefficient of each neural-model output with respect to the EM simulated value. The average relative errors of L, Q, and SRF were found to be less than 5%. This indicates good accuracy of the trained neural network. In our examples, correlation coefficients are very close to 1.0, which indicates a good fit. The maximum relative error over the different frequencies for L, Q, and SRF are 15%, 8%, and 8.5%, respectively.

B. Inductor Synthesis

During the synthesis process, the objective of optimization is to find inductor structures for a target-inductance value within the desired accuracy level. From a design point of view, the synthesized set of inductor-layout parameters that meets the design specifications of SRF and L is to be selected. Table III shows the layout geometries of the inductors as synthesized by the proposed approach for a desired inductance value of 4 nH at 1-GHz operating frequency. It is seen that the PSO-search process generates multiple sets of layout parameters with different Q and SRF values. In this example, 12 sets of layout parameters are shown for a target inductance of 4 nH within $\pm$0.01-nH accuracy. This helps the designer to make a tradeoff between Q, area (outer diameter), and SRF. It should be noted that it may not be feasible to fabricate all the inductor geometries synthesized by this approach due to the design rules of a particular process. In this case, the design values may be rounded off to the nearest grid point when doing the layout.

TABLE IV
SYNTHESIS RESULTS FOR 3-nH SPIRAL INDUCTORS AT 2.5 GHz WITH DESIGN SPECIFICATIONS: SRF > 8 GHz, $d = 100-300$ μm, $W = 8-24$ μm, $N = 2-6$, AND $s = 1-4$ μm

L (nH)	d (μm)	W (μm)	N	s (μm)	Q	SRF (GHz)
3.0028	300	15.3	2.6	2.9	6.4661	14.010
2.9908	277	18.0	3.3	3.3	6.3834	13.275
3.0042	300	17.1	2.8	4.0	6.5031	13.728
2.9909	266	13.5	2.8	1.3	6.3198	14.359
3.0020	254	15.6	3.3	1.5	6.4344	13.668
3.0041	300	18.4	2.8	1.0	6.7314	12.600
2.9941	260	16.6	3.3	1.2	6.5086	13.193
3.0076	237	13.0	3.2	1.0	6.2323	14.658
2.9947	300	24.0	3.5	1.1	6.5875	11.897
2.9928	229	11.4	3.1	1.1	5.9456	15.519
3.0000	208	11.6	3.8	2.5	5.6899	15.931

TABLE V
OPTIMIZED-INDUCTOR GEOMETRIES AND EM-SIMULATION RESULTS FOR 4-nH AND 3-nH SPIRAL INDUCTORS AT 1-GHz AND 2.5-GHz OPERATING FREQUENCIES, RESPECTIVELY

Operating Frequency		L (nH)	Q	SRF (GHz)	d (μm)	W (μm)	N	s (μm)
1GHz	ANN	4.0008	4.2319	9.6278	300	19.5	3.6	1.1
	Rounded geometries				300	19.5	3.5	1.1
	ANN	3.8943	4.209	9.7958	300	19.5	3.5	1.1
	EM	3.8315	4.1295	10.1	300	19.5	3.5	1.1
	Error (%)	1.61	1.89	3.08				
2.5GHz	ANN	3.0041	6.7314	12.6	300	18.4	2.8	1.0
	Rounded geometries				300	18.4	2.75	1.0
	ANN	2.9295	6.7507	12.81	300	18.4	2.75	1.0
	EM	3.0519	6.4536	12.51	300	18.4	2.75	1.0
	Error (%)	4.18	4.40	2.34				

TABLE VI
RUNTIME FOR SYNTHESIS PROCEDURE AND EM SIMULATION

L (nH)	4	2	3	2	2	4
Operating Frequency (GHz)	1	1	2.5	2.5	3	3
Average Runtime (sec)	26.82	26.89	22.48	18.09	20.57	27.94
Each EM simulation run (sec)	3750	3900	2550	2700	3150	3360

Table IV shows the synthesized data for a 3-nH inductor for an operating frequency of 2.5 GHz. The optimized synthesized layout geometries for the maximum Q for the two designs of Tables III and IV are consolidated in Table V. To validate the accuracy of our synthesis approach, we simulated the inductors with geometries given in Table V using the IE3D EM simulator. The L, Q, and SRF of these inductors were extracted from simulated S-parameters using the steps mentioned in Section II. The synthesized inductors show good matching with the EM simulated results. Table VI shows the average runtime required for the synthesis approach for 20 runs to achieve a target-inductance value and also the runtime for the IE3D EM simulation. The synthesis procedure and EM simulation were run on a 1.73-GHz Pentium-IV machine with 256-MB RAM. EM simulation of spiral inductors were done up to a frequency of 15 GHz with 0.5 GHz as step size. On average, our proposed synthesis procedure evaluates more than 6000 spiral structures using the trained neural network in each run for which the EM solver would have taken several thousand hours. Since the initial EM simulation done for generating training data in our approach is an off-line process, it does not adversely affect the efficiency of the on-line synthesis steps.

IV. CONCLUSION

We have proposed an efficient layout synthesis procedure for RF on-chip spiral inductors. A four-layer MLP neural model has been developed. All the output parameters of the neural model show good matching (within 5% accuracy) when compared with the data generated by an EM simulator.

The synthesis procedure is based on a PSO technique that evaluates the electrical parameters from the geometric parameters using the neural model. No EM simulation is required during the synthesis procedure thus making the process efficient. The synthesis procedure provides multiple solutions for a given design specification that helps the designer in making a tradeoff between the competing objectives. Several design examples have been presented using the proposed approach. The synthesized inductors were resimulated using the IE3D EM solver. The results obtained by our synthesis approach show good agreement with the EM simulation results. Future directions would be to consider other inductor geometries as well.

REFERENCES

[1] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design—From theory to practice," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003.

[2] P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 9, no. 3, pp. 158–174, May 1999.

[3] A. M. Niknejad and R. G. Meyer, "Analysis, design, and optimization of spiral inductors and transformers for Si RF IC's," *IEEE J. Solid-State Circuits*, vol. 33, no. 10, pp. 1470–1481, Oct. 1998.

[4] S. S. Mohan, M. M. Hershenson, S. P. Boyd, and T. H. Lee, "Simple accurate expressions for planar spiral inductances," *IEEE J. Solid-State Circuits*, vol. 34, no. 10, pp. 1419–1424, Oct. 1999.

[5] Y. Zhan and S. Sapatneker, "Optimization of integrated spiral inductors using sequential quadratic programming," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, Feb. 2004, vol. 1, pp. 622–627.

[6] C. P. Yue and S. S. Wong, "Physical modeling of spiral inductors on silicon," *IEEE Trans. Electron Devices*, vol. 47, no. 3, pp. 560–568, Mar. 2000.

[7] J. E. Post, "Optimizing the design of spiral inductors on silicon," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 1, pp. 15–17, Jan. 2000.

[8] N. A. Talwalkar, C. P. Yue, and S. S. Wong, "Analysis and synthesis of on-chip spiral inductors," *IEEE Trans. Electron Devices*, vol. 52, no. 2, pp. 176–182, Feb. 2005.

[9] T. Wang, Y. Wang, and K. Chen, "A global genetic algorithm based optimization technique for spiral inductor on silicon design," in *Proc. 5th World Congr. Intell. Control Autom.*, Hangzhou, China, Jun. 15–19, 2004, vol. 3, pp. 2095–2098.

[10] J. Ababneh, M. Khodier, and N. Dib, "Synthesis of interdigital capacitors based on particle swarm optimization and artificial neural network," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 16, no. 4, pp. 322–330, Jul. 2006.

[11] Y. Rahmat-Samii, "Genetic algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics," in *Proc. ICECom*, Dubrovnik, Croatia, Oct. 1–3, 2003, pp. 1–5.

[12] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Congr. Evol. Comput.*, 2001, vol. 1, pp. 81–86.

[13] S. K. Mandal, S. Sural, and A. Patra, "Broadband scalable model for Si-RF on-chip spiral inductors with substrate eddy current effect," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 17, no. 6, pp. 560–573, Nov. 2007.

[14] *IE3D*, Fremont, CA: Zeland Software, Inc. [Online]. Available: http://www.zeland.com

[15] H. Demuth and M. Beale, *Neural Network Toolbox for use with MATLAB*. Natick, MA: MathWorks Inc.

[16] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proc. 7th EP*, pp. 591–600.

[17] J. Park, K. Choi, and D. J. Allstot, "Parasitic-aware RF circuit design and optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 1953–1966, Oct. 2004.