Synthesis of Integrated Passive Components for High-Frequency RF ICs Based on Evolutionary Computation and Machine Learning Techniques

Bo Liu, Dixian Zhao, Patrick Reynaert, and Georges G. E. Gielen, *Fellow, IEEE*

***Abstract*—State-of-the-art synthesis methods for microwave passive components suffer from the following drawbacks. They either have good efficiency but highly depend on the accuracy of the equivalent circuit models, which may fail the synthesis when the frequency is high, or they fully depend on electromagnetic (EM) simulations, with a high solution quality but are too time consuming. To address the problem of combining high solution quality and good efficiency, a new method, called memetic machine learning-based differential evolution (MMLDE), is presented. The key idea of MMLDE is the proposed online surrogate model-based memetic evolutionary optimization mechanism, whose training data are generated adaptively in the optimization process. In particular, by using the differential evolution algorithm as the optimization kernel and EM simulation as the performance evaluation method, high-quality solutions can be obtained. By using Gaussian process and artificial neural network in the proposed search mechanism, surrogate models are constructed online to predict the performances, saving a lot of expensive EM simulations. Compared with available methods with the best solution quality, MMLDE can obtain comparable results, and has approximately a tenfold improvement in computational efficiency, which makes the computational time for optimized component synthesis acceptable. Moreover, unlike many available methods, MMLDE does not need any equivalent circuit models or any coarse-mesh EM models. Experiments of 60 GHz syntheses and comparisons with the state-of-art methods provide evidence of the important advantages of MMLDE.**

***Index Terms*—Artificial neural network, differential evolution, gaussian process, inductor synthesis, microwave components, surrogate model, transformer synthesis.**

I. Introduction

IN RECENT years, design methodologies for high-frequency microwave circuits have attracted a lot of attention. In particular, research on RF building blocks for 40 GHz to 120 GHz and beyond is increasing drastically. On-chip passive components, e.g., inductors and transformers, are one of the major components of the RF IC that strongly influence the circuit performances [1]. For example, the loss of a transformer has a large impact on the power-added efficiency and the output power of a power amplifier. Therefore, the synthesis of passive components, including both the sizing and the layout optimization, is a critical problem in high-frequency RF IC design automation. High-frequency RF component synthesis faces two challenges. First, accurate equivalent circuit models are often not available in literatures at these frequencies. Some designers rely on experience and simulation verification in their design work. Another challenge is that the performance requirements of RF ICs keep on increasing, and therefore powerful optimization methods are needed. Hence, the "experience and trial" method or local optimization is often not good enough for high-frequency RF component design. This paper focuses on these problems.

Most RF passive components synthesis can be naturally expressed as a constrained optimization problem [2]: the optimization of an objective (e.g., quality factor), usually subject to some constraints (e.g., self-resonance frequency). The special point is that in order to obtain an accurate result, electromagnetic (EM) simulation of the component structure is typically necessary, especially at high frequencies. However, EM simulations are often very CPU time expensive [3]. This fact highly increases the need of high efficiency of the synthesis framework. Hence, most of the state-of-the-art methodologies [1]–[9] focus on the tradeoff between the solution quality and the efficiency.

In this paper, we propose a new framework, the memetic machine learning-based differential evolution (MMLDE) method, focusing on optimized RF passive component synthesis at high frequencies. Compared to available methods with the best solution quality, MMLDE can obtain comparable results, but has approximately a tenfold improvement in computational efficiency. A high-performance passive component for RF ICs can be synthesized in a very reasonable time, which is in the order of a few hours clock time on a single CPU node.

The remainder of this paper is organized as follows. Section II reviews the related works and motivates the strategy of MMLDE. Section III introduces the components and the general framework of MMLDE. Section IV tests MMLDE on practical examples at 60 GHz. Comparisons with the state-of-the-art methods are also performed. Concluding remarks are presented in Section V.

Manuscript received March 4, 2011; revised May 6, 2011; accepted June 17, 2011. Date of current version September 21, 2011. This work was supported by a bilateral agreement scholarship of Katholieke Universiteit Leuven, Leuven, Belgium, and Tsinghua University, Beijing, China. This paper was recommended by Associate Editor H. E. Graeb.

The authors are with Katholieke Universiteit Leuven, Leuven 3000, Belgium (e-mail: bo.liu@esat.kuleuven.be; dixian.zhao@esat.kuleuven.be; patrick.reynaert@esat.kuleuven.be; georges.gielen@esat.kuleuven.be; liu_bo 765@yahoo.com.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2011.2162067

II. Related Works and Motivations

The available computer-aided design optimization methodologies for microwave components can be classified into four categories: 1) equivalent circuit model and global optimization algorithm based (ECGO) methods [4], [5]; 2) EM-simulation and global optimization algorithm based (EMGO) methods [1]; 3) off-line surrogate model, EM-simulation and global optimization algorithm based (SEMGO) methods [2]; and 4) surrogate model and local optimization algorithm based (SMLO) methods [3], [6]–[9]. These will now be described in more detail.

1) The ECGO methods [4], [5] depend on the equivalent circuit model to obtain the performances of the microwave structure. Their advantage is high efficiency. The synthesis of a 5 GHz inductor considering process variations, which requires many performance evaluations, has been achieved successfully and efficiently by ECGO [4]. On the other hand, when the frequency is high, equivalent circuit models available in the microwave area are typically not accurate enough or difficult to find. Hence, even with global optimization algorithms, the synthesis of high-frequency components may also fail as the used equivalent circuit models may not reflect well the performances of the microwave structures.
2) The EMGO methods [1] can provide an accurate performance analysis of the microwave structure because they use EM simulations. Combined with global optimization algorithms, the quality of the solution is the best among all the available methods, especially in high-frequency RF component synthesis. However, its major bottleneck is the high computational cost of the EM simulations limiting their use in practice [3].
3) Reference [2] represented a surrogate-model EMGO (SEMGO), which is an important progress of EMGO. SEMGO uses an off-line artificial neural network (ANN) model to enhance the speed of the standard EMGO. In [2], the surrogate model is first trained to approximate the performance of the microwave structure before optimization. Then, the optimization algorithm uses this surrogate model as the performance evaluator to find the optimal design. The training data are generated uniformly in the design space and the corresponding performances are obtained by EM simulations. When combined with global optimization algorithms, this method has the ability of global search. However, the training data generation process in this method is expensive and we found that the constructed ANN model is not always reliable in our 60 GHz inductor synthesis example (see Section IV).
4) The SMLO methods [3], [6]–[9] combine the efficiency of ECGO with the accuracy of the EM simulations from EMGO. Fig. 1 shows the general flow. First, a *coarse model*, either an equivalent circuit model or a model evaluated by EM simulation but with coarse meshes, is constructed and optimized. Then, some base vectors in the vicinity of the optimal point of the coarse model are selected as the base points to train a surrogate model,

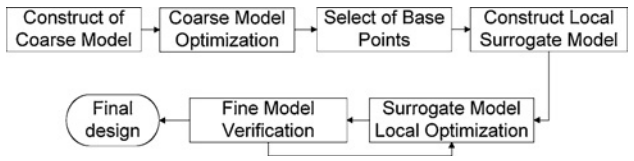
Construct of Coarse Model
Coarse Model Optimization
Select of Base Points
Construct Local Surrogate Model
Surrogate Model Local Optimization
Fine Model Verification
Final design

Fig. 1. Flow of the SMLO methods.

whose purpose is to predict the performances of the microwave structure. At last, the surrogate model is used to optimize the microwave component, whose result is verified by the fine model using expensive high-fidelity EM simulations. The data received by the fine simulations will update the surrogate model to make it more accurate. In the development of the SMLO methods, some works have been presented focusing on selecting the coarse model [3], [7] and the surrogate model [3], [8]. SMLO, however, highly depends on the accuracy of the coarse model, which leads to two significant challenges for high-frequency RF passive component synthesis. First, the optimal solution of the coarse model defines the search space and the constructed surrogate model is only accurate in that space, because the base points are selected around it [3]. The success of SMLO comes from the basic assumption that the optimal point of the coarse and fine models are not far away in the design space, as shown in [3] and [6]–[9]. However, this assumption only holds when the coarse model is accurate enough. Although it has been shown that SMLO can solve RF component synthesis well at comparatively low frequencies [3], [6]–[9] (e.g., 10 GHz), for passive components in high-frequency RF ICs (e.g., 60 GHz), this assumption is often not true. In many cases, a workable equivalent circuit model is even difficult to find, and the mesh density of the coarse-mesh EM model is difficult to decide. The second challenge is that SMLO can only do local search, which is not suited for synthesis with strong requirements. This is not only because of the fact that the current SMLO methods use local optimization algorithms, but also because of the fact that the search space is decided first by the coarse model [3], [6]–[9]. Therefore, using global optimization algorithms makes little sense.

In summary, ECGO and SMLO work well in comparatively low-frequency RF component synthesis, but their high dependence on the accuracy of the equivalent circuit or coarse model limits their use for the synthesis of high-frequency microwave structures. EMGO can provide high-quality results even when the frequency is high, but is too CPU time intensive. Although SEMGO [2] makes a great progress on EMGO, to the best of our knowledge, the development of sufficiently effective and efficient synthesis methods for high-frequency microwave components is still in great need.

To address these problems, we propose a new framework, the MMLDE method. The key idea of MMLDE is the proposed online surrogate-model-based memetic evolutionary optimization mechanism, whose training data are generated adaptively in the optimization process. The efficiency versus quality targets aimed at with MMLDE are shown in Fig. 2.

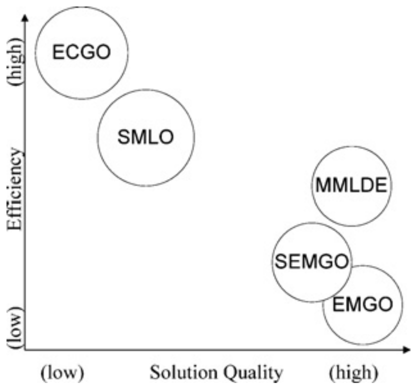

Fig. 2. Review of the available methods in HIGH-FREQUENCY component synthesis and the targets of MMLDE.

In addition, MMLDE does not need any coarse model nor the complex tuning of the parameters.

III. The MMLDE Algorithm

A. Key Ideas of MMLDE

Two conclusions can be drawn from the available methods: 1) global optimization and EM simulations are the keys to obtain high-quality solutions, and 2) machine learning techniques, or the surrogate model in this application, are the keys to enhance the efficiency.

Hence, the question becomes: how to integrate the machine learning techniques with the global optimization and the EM simulation-based algorithm? The answer, however, is *not* trivial. In the literature, there are mainly two kinds of methods that use surrogate models in the synthesis of RF components. The first one is the method used in SMLO. The second one is the off-line surrogate model in SEMGO [2]. More information of these two methods has been presented in Section II.

In MMLDE we propose a new framework. After a small Latin-hypercube sampling (LHS) of the design space and using EM simulations to evaluate these samples, we train the initial surrogate model as a rough estimation of the performances of the microwave structure. Then, we use our online surrogate-model-based evolutionary algorithm. In each iteration, the candidate solutions are generated by the memetic evolutionary computation algorithm, whose performances are evaluated by the surrogate model. We perform EM simulation to the candidate solution with the possible best *potential* to improve the objective function. Note that the candidate solution with the best potential is not simply the one with the best predicted value, like SMLO. In MMLDE, the potential is calculated based on the used machine learning technique and the corresponding potential measurement method. We then *update* the surrogate model by including the new candidate with EM simulation result. There is only one EM simulation in each iteration.

The MMLDE mechanism is different from the mechanism of SEMGO [2]. SEMGO first constructs a good surrogate model which covers the whole design space and then uses it. In the optimization process, there are no EM simulations and updating. To obtain a reliable surrogate model, the training data need to cover the whole design space with a reasonably high density. Hence, a lot of EM simulations are necessary. On the other hand, only a small part of the design space is useful in the optimization. The reason is that the optimization algorithm in [2] is not based on enumeration, but based on iteration, so many of these expensive EM simulations are wasted. In contrast, MMLDE holds to the idea of "in the deep darkness of the design space, there is no need to lighten the whole world but rather the close vicinity of the path to the destination." MMLDE, therefore, first constructs a very rough surrogate model, and then improves it online but only in the necessary area of the design space, which is determined by the optimization algorithm and the updating technique. Consequently, MMLDE is more efficient in terms of the number of EM simulations than SEMGO. Moreover, because all the performances that have potential to be used as the final result are evaluated by EM simulations, rather than by the surrogate model, MMLDE is also more accurate.

Although there also exists an updating process in SMLO, this updating in MMLDE is largely different from the updating in SMLO. The main purpose of the updating in SMLO is to improve the local accuracy and to help local search. One reason is that the search space is defined by the optimal point of the coarse model and the surrogate model is only accurate in the vicinity of that point. Moreover, even when the coarse model is accurate and the global optimal point is included in the newly defined search space, the updating which only considering the predicted value also causes a low probability to achieve global optimization [10], [11]. The reason is that the updating mechanism only using the predicted value puts too much emphasis on exploiting the predictor and no emphasis on exploring points where we are uncertain. In contrast, the updating in MMLDE can both guide the global and local search, which is achieved by the memetic evolutionary algorithm and the method to decide the candidate with the possible best potential. For the Gaussian process-based surrogate model, we use the expected improvement (EI) [10] to measure the potential of the candidate. For the ANN-based surrogate model, we directly use the predicted value to measure the potential. The EI measurement has the ability to judge the potential for global search for a candidate because the uncertainty of the Gaussian process prediction is considered. Hence, the quality of a candidate point is considered in a global picture. When combined with evolutionary algorithms, global optimization can therefore be achieved. On the other hand, the potential measurement used for the ANN surrogate model is more powerful in local refinement compared with the EI measurement. Hence, we combine the two machine learning and potential measurement techniques to construct a memetic evolutionary algorithm with enhanced search ability and efficiency.

In the following, the basic components of MMLDE will be introduced first. The key techniques and the general framework will be presented afterward.

B. Using Gaussian Process in MMLDE

Gaussian process (GP) machine learning [12]–[14] is one of the chief techniques to construct the surrogate model in MMLDE. GP machine learning not only has very good prediction ability, but also can provide a meaningful uncertainty measurement for a prediction. This is very important when combined with optimization. For online surrogate-model-based optimization, the accuracy and reliability of the GP model is improved gradually in the process of optimization, as more additional training data are provided throughout the optimization process. This leads to a problem that some data predicted by the GP model may have large differences compared with the real EM simulation results, especially when the training data are not sufficient. Hence, if we only use the predicted values, it is very easy to be trapped in a local optimum. To prevent this, we use the EI measurement [10] to call for a balance between exploration and exploitation, which are computed by the predicted value and the standard error (uncertainty measurement).

Here, we provide an intuitive introduction and the main formulas for the technique of GP machine learning [15].

GP predicts a function value $y(x)$ at some design point x by modeling $y(x)$ as a stochastic variable with mean μ and variance σ. If the function is continuous, the function values of two points x_i and x_j should be close if they are highly correlated. In this paper, we use the Gaussian correlation function to describe the correlation between two variables

$$Corr(x_i, x_j) = \exp(-\sum_{l=1}^{d} \theta_l |x_{il} - x_{jl}|^2) \tag{1}$$

where d is the dimension of x and θ_l is the correlation parameter which determines how fast the correlation decreases when x_{il} moves in the l direction. The formulas to decide θ_l can be found in [16]. The values of μ, σ and θ are determined by maximizing the likelihood function of the observed data. Suppose that there are n observed data $x = (x_1, x_2 \cdots, x_n)$, and their corresponding function values are $y = (y_1, y_2 \cdots, y_n)$, then the optimal values of μ and σ can be found by setting the derivatives of the likelihood function (2) to 0

$$\begin{aligned} h = &\frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}|R|^{1/2}} \exp(-\frac{1}{2\sigma^2} \\ &(y - I\mu)^T R^{-1}(y - I\mu)) \end{aligned} \tag{2}$$

where I is a $n \times 1$ vector of ones, R is the correlation matrix and

$$R_{i,j} = Corr(x_i, x_j),\ i, j = 1, 2, \cdots n. \tag{3}$$

By solving the equations, the $\hat{\mu}$ and $\hat{\sigma}^2$ are as follows:

$$\hat{\mu} = (I^T R^{-1} I)^{-1} I^T R^{-1} y \tag{4}$$

$$\hat{\sigma}^2 = (y - I\hat{\mu})^T R^{-1}(y - I\hat{\mu})n^{-1}. \tag{5}$$

Using the GP model, the function value $y(x^*)$ at a new point x^* can be predicted as (x^* should be added in R, r)

$$\hat{y}(x^*) = \hat{\mu} + r^T R^{-1}(y - I\hat{\mu}) \tag{6}$$

where

$$r = [Corr(x^*, x_1), Corr(x^*, x_2), \cdots, Corr(x^*, x_n)]^T. \tag{7}$$

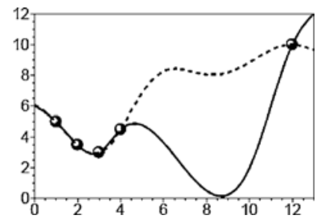

Fig. 3. Solid line represents an objective function that has been sampled at the five points shown as dots. The dotted line is a DACE predictor fit to these points (from [10]).

The measurement of the uncertainty of the prediction, i.e., the mean square error (MSE), which is used to assess the model accuracy, can be described as

$$MSE(x^*) = \hat{\sigma}^2[I - r^T R^{-1} r + (I - r^T R^{-1} r)^2 (I^T R^{-1} I)^{-1}]. \tag{8}$$

In this paper, we use the DACE toolbox [16] to implement the Gaussian process machine learning.

Besides the above basic principles from GP machine learning, we introduce another important concept, the expected improvement EI [10], which is calculated as

$$\begin{aligned} E[I(x)] = &(f_{\min} - y(x))\Phi(\frac{f_{\min} - y(x)}{\sqrt{MSE(x)}}) \\ &+ \sqrt{MSE(x)}\phi(\frac{f_{\min} - y(x)}{\sqrt{MSE(x)}}) \end{aligned} \tag{9}$$

where $f_{\min}$ is the current best function value in the population (the population with EM simulation results, not the generated population after evolutionary operators). $\phi(\cdot)$ is the standard normal density function, and $\Phi(\cdot)$ is the standard normal distribution function. $I(x)$ is the improvement of f.

EI measures the potential of a candidate solution in MMLDE, which considers both global search and local search. EI is the part of the curve of the standard error in the model that lies below the best function value sampled so far. Figs. 3 and 4 provide an example. As shown in Fig. 3, the function value of $x = 8$ is better than that of $x = 3$, but $x = 8$ cannot be selected when directly using the GP prediction values. However, the point $x = 8$ is possible to be selected when using the EI measurement. In Fig. 4, the probability density of the prediction uncertainty at the point $x = 8$ in the curve of the DACE predictor is shown by curve B. We can find that at the tail of the density function (area A), the EI value of $x = 8$ is better than the EI of the current $f_{\min}$ (near $x = 3$), so it is possible that the true value at $x = 8$ is better than the current $f_{\min}$. Mathematically, the potential is calculated by (9). More details are in [10].

C. *Using Artificial Neural Network in MMLDE*

An ANN is a computational mechanism, the structure of which essentially mimics the process of knowledge acquisition, information processing and organizational skills of a human brain. An ANN has the capability of learning complex nonlinear relationships and associations from a large volume of data, and enables the analysis of a wide range of pattern recognition [17]. An ANN is composed of a number of highly interconnected neurons, usually arranged in several layers. These layers generally include an input layer, a number of

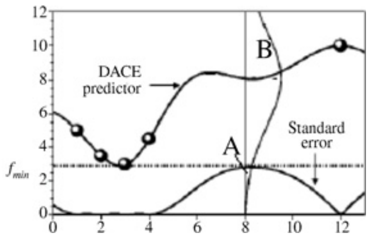

Fig. 4. Uncertainty about the function's value at a point (such as $x = 8$ above) can be treated as if there were a realization of a normal random variable with mean and standard deviation given by the DACE predictor and its standard error (from [10]).

hidden layers, and an output layer. Signals generated from the input layer propagate through the network on a layer-by-layer basis in the forward direction. Neurons in the hidden layers are used to find associations between the input data and to extract patterns that can provide meaningful outputs. The output of each neuron that responds to a particular combination of inputs has an impact on the overall output. The weight is controlled by the level of the activation of each neuron, and the strength of the connections between the individual neurons. Patterns of activation and interconnections are adjusted through a training process to achieve the desired output for the training data. If the averaged error is within a predefined tolerance, the training is stopped and the weights are locked in; the network is then ready to be used [18]. In MMLDE we use a feed-forward ANN with one hidden layer and the predicted value of the ANN model is used to measure the potential of the microwave structure.

D. Optimization Kernel: The DE Algorithm

For the optimization core we choose an evolutionary computation (EC) algorithm. It may seem that they may cost more function evaluations compared with non-population-based algorithms. However, choosing EC is motivated by the following three considerations: 1) EC algorithms can achieve global optimization, which is the aim of this paper; 2) although a group of candidates is generated in each iteration, we only perform one EM simulation for the candidate with the possible best potential; and 3) the evaluations of individuals in the EC algorithms are independent of each other in a population, so it is very suited for parallel computation to enhance the efficiency. On the other hand, non-population-based optimization algorithms can do this not so easily, hence many of them cannot be combined with parallel computation. Although our current implementation in this paper does not yet use parallel computation techniques, powerful parallel computation techniques are available.

The DE algorithm [19] is selected as the search engine in MMLDE. The DE algorithm outperforms many other EC algorithms in terms of solution quality and convergence speed [19]. DE uses a simple differential operator to create new candidate solutions and a one-to-one competition scheme to greedily select new candidates.

The ith candidate solution in the d-dimensional search space at generation t can be represented as

$$x_i(t) = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}]. \tag{10}$$

At each generation t, the *mutation* and *crossover* operators are applied to the candidate solutions, and a new population arises. Then, *selection* takes place, and the corresponding candidate solutions from both populations compete to comprise the next generation. The operators are now explained in detail.

For each target candidate solution, according to the *mutation* operator, a mutant vector is built

$$V_i(t+1) = [v_{i,1}(t+1), \ldots, v_{i,d}(t+1)]. \tag{11}$$

It is generated by adding the weighted difference between a given number of candidate solutions randomly selected from the previous population to another candidate solution. The mutation operation is therefore described by the following equation (DE/best/1/bin [19]):

$$V_i(t+1) = x_{\text{best}}(t) + F(x_{r1}(t) - x_{r2}(t)) \tag{12}$$

where indices r_1 and r_2 ($r_1, r_2 \in \{1, 2, \ldots, NP\}$) are randomly chosen and mutually different, and also different from the current index i. Parameter $F \in (0, 2]$ is a constant called the scaling factor, which controls the amplification of the differential variation $x_{r1}(t) - x_{r2}(t)$. The base vector to be perturbed $x_{\text{best}}(t)$ is the best member of the current population, so that the best information can be shared among the population. To avoid stagnation and to improve the balance between exploration and exploitation, we use the random-scale search DE mutation operator. In this mutation, for the scaling factor we use a vector $\hat{F}$ composed of Gaussian-distributed random variables with mean value μ and variance σ: $\hat{F}_{i,j} = norm(\mu, \sigma)$, $i = 1, 2, \ldots NP$, $j = 1, 2, \ldots d$. Equation (12) is therefore changed to (13). For more details please refer to [19]

$$V_i(t+1) = x_{\text{best}}(t) + \hat{F}_i(x_{r1}(t) - x_{r2}(t)). \tag{13}$$

After the *mutation* phase, the *crossover* operator is applied to increase the diversity of the population. Thus, for each target candidate solution, a trial vector is generated as follows:

$$U_i(t+1) = [u_{i,1}(t+1), \ldots, u_{i,d}(t+1)] \tag{14}$$

$$u_{i,j}(t+1) = \begin{cases} v_{i,j}(t+1), \text{ if } (rand(i,j) \le CR) \text{ or } j = randn(i) \\ x_{i,j}(t), \text{ otherwise} \end{cases} \tag{15}$$

where $rand(i, j)$ is an independent random number uniformly distributed in the range [0, 1]. Parameter $randn(i)$ is a randomly chosen index from the set $\{1, 2, \ldots, d\}$. Parameter $CR \in [0, 1]$ is a constant called the crossover parameter, which controls the diversity of the population.

Following the *crossover* operation, the *selection* operation decides on the population of the next generation ($t + 1$). In standard DE, $U_i(t + 1)$ is compared to the initial target candidate solution $x_i(t)$ by a one-to-one-based greedy selection criterion. However, in MMLDE, we do not use this selection operator, because we need to minimize the number of EM simulations. Instead, we select the best solution (or solution with the possible best potential) among all the trial solutions $U(t+1)$ and then perform EM simulation to it. Then, we add this new solution to the population. Note that in MMLDE the "population" only refers to the samples with EM simulation results, and a new point is added into the population at each iteration. The candidates generated by the DE operators are called trial individuals.

E. Integrating Surrogate Models into the EC Algorithm

In MMLDE, there is an initial surrogate model and the surrogate model is continuously updated throughout the optimization process. Let us introduce the initial surrogate model first. On one hand, the number of samples in the initial surrogate model needs to be as small as possible; otherwise the efficiency will decrease. On the other hand, we need to cover the design space as much as possible, because too sparse samplings will cause very little information in some areas and the reliability of the surrogate model will be poor. To make a good tradeoff, we use LHS sampling. The LHS sampling method samples the design space more uniformly, and hence, can use fewer samples to achieve more effective sampling. For example, LHS often requires 20%–25% of the number of samples compared to primitive Monte Carlo sampling when estimating yield [21]. In MMLDE, the number of samples is correlated to the dimension of the design variables, d. If d is larger than 3, we use $11 \times d - 1$ as the number of initial samples; otherwise, we use $8 \times d - 1$.

Although the initial surrogate model can roughly reflect the performances of the RF passive components, its quality is not good enough. Therefore, if we would directly use the predicted value to guide further search, it would be very easy to be trapped in a local optimum. The interpretation is quite intuitive. For the GP model the predicted value of a point is decided by the function values of the points near it. It may be possible that there are more points around a local optimal point than around the global optimal point. In this case, the function value of the local optimal point can be predicted quite well and with less uncertainty, while the function value of the global optimal point may be predicted poorly (as the information is little) and have a large uncertainty. The result may be that the predicted value of the global optimal point is worse than that of the local optimal point, which will cause a wrong selection. To address this problem, we use the EI [10] (9) to measure the potential, which considers both the predicted value and the uncertainty of the prediction (see Section III-B).

EI considers both global search and local search. Especially when the sampling is sparse (large standard error), it has a high ability to consider the potential for global search. It can also consider the potential for local search, especially when the sampling is dense. However, in the MMLDE mechanism the sampling can seldom be dense, because we want to use a limited number of EM simulations to finish the synthesis. Hence, the local search ability of using the EI measurement is comparably weak. We therefore use a memetic algorithm [22] to compensate the local search ability. In addition to the global optimization engine, memetic algorithms use a population-based strategy coupled with individual search heuristics capable of performing local refinements. In the revised DE algorithm used in MMLDE, we use the same evolution operators and select the candidate with the possible best potential in each iteration. Hence, the global or local search engine is defined by the method to determine the potential of a candidate and the corresponding surrogate model type. For GP, we use EI to measure the potential when focusing on global exploration; for ANN, we use the predicted value to measure the potential when focusing on local exploitation. An illustrative example is shown in Fig. 5. The function to be predicted is $y = 2.5 \times \sin(x)$. From the ANN prediction values and the potential measured by EI, we can see that EI predicts that x_B (the corresponding x value of point B) has the best potential, but ANN prediction correctly selects the best point x_A. It can be noticed that there are two training data on each side of x_A and x_B, which influence the GP prediction and EI measurement mostly. x_B has a smaller distance to the training point with f_{min} compared with x_A, and the opposite on the other side. Hence, the EI value of x_B is larger. On the other hand, in this local area without dense sampling, ANN catches the shape of the curve and predicts better.

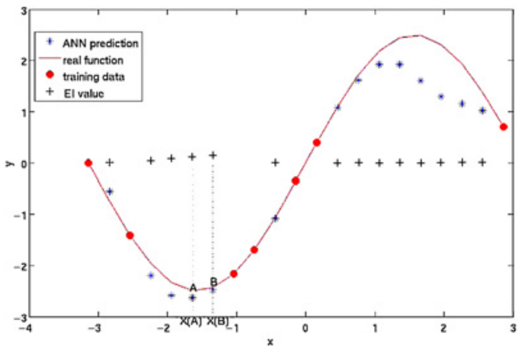

Fig. 5. Illustrative example for EI and ANN results in local search.

From the experiments on passive components synthesis, we also found that the best candidate chosen by the ANN and the corresponding potential measurement method has a higher probability of being the local refinement compared with GP with EI measurement. Note that it does not mean that GP is for global exploration and ANN is for local exploitation. They cooperate with each other. The GP-based surrogate model and the EI measurement also consider local refinement, and the ANN may also help GP on global search by its samplings.

The mechanism of MMLDE is as follows: after the LHS sampling to construct the initial surrogate model, we first use GP with the EI measurement to determine the potential of the candidates for a certain number of generations. The reason is that in this period the number of samples is not sufficient, global search needs to be emphasized. Then, we continue to use GP, but when no improvement is shown for a certain number of generations using GP, we use the ANN prediction values to define the potential. If there is no improvement for a certain number of generations by the ANN, we come back to the GP and EI. We iteratively do this process until the termination condition is met.

F. Other Components

To handle constraints, we use the static penalty function method [23] in MMLDE.

Algorithm 1 The MMLDE algorithm

Step 0: Initialize the parameters, e.g., the generation threshold of continuously using GP/ANN when no improvement is shown, the DE algorithm parameters (e.g., CR), the GP parameters (e.g., the correlation function), the ANN parameters (e.g., the number of neurons, the training algorithm).
Step 1: Initialize the population by LHS sampling of the design space. The EM simulations are performed for the sampled design points.
Step 2: Check if the stopping criterion (e.g., a convergence criterion or a maximum number of iterations) is met. If yes, output the result; otherwise go to step 3.
Step 3: Judge to use the GP or ANN machine learning technique as described in Section III-E.
Step 4: Train the selected surrogate model according to the available samples (population).
Step 5: Use the available samples as the current population, and perform the mutation operation according to (13) to obtain each candidate solution's mutant counterpart.
Step 6: Perform the crossover operation between each candidate solution and its corresponding mutant counterpart according to (14) and (15) to obtain each individual's trial individual.
Step 7: According to the selected model in Step 3, use the EI or the predicted value to select the individual with the possible best potential and perform the EM simulation to it.
Step 8: Update the population by adding the point from Step 7 and its performance. Update the best solution obtained so far. Update other parameters. Go back to Step 2.

GP has the assumption of Gaussian distribution of the input and output data. Hence, normalization should be done to the training data. But when the range of the data is large, directly normalizing the design points X and the performances Y may not yield good results. In this case, we use transformation functions (e.g., $\log_m^x$) to decrease the range of the input/output variables.

G. General Framework of MMLDE

Based on the above components, the overall MMLDE algorithm for the synthesis of high-frequency RF passive components can now be constructed. The detailed flow diagram is shown in Fig. 6 and the description is in Algorithm 1.

It can be seen from Fig. 6 that at every iteration there is a step to decide to use the GP model or the ANN model. The rules have been described in the last paragraph of Section III-E and are not shown in the figure.

H. Parameter Settings of MMLDE

In MMLDE, there are several algorithm parameters which need to be set by the user. They can be classified into five groups: the DE parameters, the GP parameters, the ANN parameters, the number of initial samples and the generation threshold for alternating GP and ANN. Here, we provide some recommended settings for each of them.

1) The DE parameters: two parameters need to be set in the DE optimization algorithm, which are the scaling factor F and the crossover rate CR. For F, we use a Gaussian distributed random number with $\mu = 0.75$ and $\sigma = 0.25$. The reason is shown in [20]. For CR, it is often set from 0.5 to 1. The smaller, the more diversity of the population, but the convergence rate is also lower. We set CR to 0.8, which is a very commonly used setting for single objective optimization [19].
2) The GP correlation function: for the problem of high-frequency passive components synthesis, we suggest using the Gaussian correlation function. The reason is that through experiments, we found this correlation model having the best results, while the exponential correlation model [16] can also be considered.
3) The ANN parameters: for the training algorithm, the LM algorithm [24] is used, which is a very commonly used method in ANN training. For the number of hidden layers, in most real world applications, one hidden layer is chosen if a feed-forward ANN is used for fitting. In the problem of passive component synthesis, the number of design variables is not too many, and 8–15 neurons in the hidden layer is a common choice. We use ten neurons in the hidden layer, as other settings in this range do not have much effect both on run time and performances.
4) The number of initial samples: we use $11 \times d - 1$ when the dimension is larger than 4, and $8 \times d - 1$ when the dimension is less than 4. This setting is based on the "10k" rule for space filling [10]. The "10k" rule suggests using $10 \times d$ (d is the number of design variables) LHS samples to uniformly cover the design space for initialization in meta-model assisted optimization. Because LHS sampling is used, the design space can often be uniformly covered in not very high-dimensional problems. The density of the filling is decided by m when using $m \times d$ LHS samplings. $8 \times d$ to $12 \times d$ are all used in practice. In order to avoid that a relatively large part of the design space is not sampled because

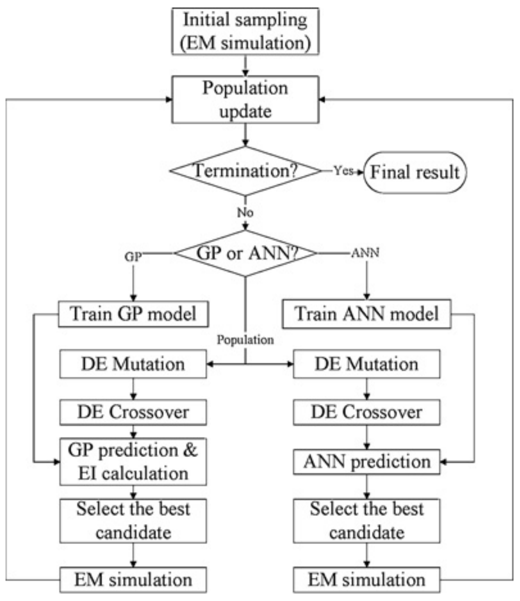

Fig. 6. Flow diagram of MMLDE.

of the sparseness of the initial sampling, $11 \times d - 1$ is used when the number of design variables is larger than 4, which is a very safe setting. For the problems with less than four dimensions, on the other hand, even when we have sparse initial samples, the updating mechanism has a high probability to remedy the sparseness of the initial sampling because the dimension is low. Therefore, we select $8 \times d - 1$. Example 1 also shows that a good result is obtained using the $8 \times d - 1$ setting.

5) The generation threshold: the generation threshold determines after how many generations without improvement the adaptive learning method, i.e., GP or ANN is switched. The recommended setting is 20 if the number of design variables is larger than 4; otherwise, this parameter is set to 10. This is a setting based on experience and tests. The two experiments show that this setting is effective.

IV. Experimental Results and Comparisons

In this section, the MMLDE algorithm is demonstrated for the synthesis of a 60 GHz inductor and a 60 GHz transformer in a 90 nm CMOS technology. The top two metal layers are used. ADS-momentum is used as the EM simulator. The bounds of the design variables are set both by the design rules of the technology and the experience of the designer. The two examples are all constrained optimization problems. For the optimization core, DE is used. Because the advantages of the DE algorithm in circuit sizing have been demonstrated in [25], such comparisons will not be repeated here. MMLDE stops when the performance cannot be improved for 40 consecutive generations. The performance of evolutionary algorithms may be affected by the random numbers used in the evolution operators. Therefore, ten runs with independent random numbers are performed for all the experiments and the results are analyzed and compared statistically. The examples are run on a PC with Intel 2.4 GHz Xeon CPU and 12 GB RAM under Linux operating system. No parallel computation is applied yet in these experiments. All the time consumptions in the experiments are clock time.

The reference methods we selected for the comparisons are as follows. The common reference method for the two examples is standard EMGO with the same DE optimization kernel but fully using EM simulations. The purpose is to provide the best result to test the other methods. Obviously, it is the most CPU expensive method. Because, to the best of our knowledge, a good enough equivalent-circuit model for 60 GHz integrated inductors is difficult to find, we do not compare MMLDE with SMLO for Example 1. Instead, we select SEMGO [2] and the MMLDE framework but only with the GP or ANN model as the reference methods. The latter two reference methods all use the same optimization kernel and parameter settings as MMLDE, and are abbreviated as EMGOG and EMGOA, respectively. For the second example, we choose a widely used equivalent-circuit transformer model and the reference method is a revised SMLO (RSMLO). SMLO clearly has the best efficiency, but the goal of MMLDE is to combine good synthesis ability with low, practical computation time.

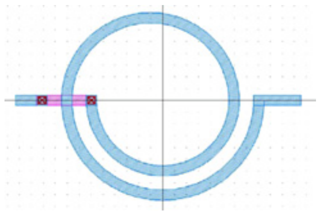
Fig. 7. Typical inductor result for Example 1.

So comparing the speed with SMLO is not our purpose unless SMLO also receives a good result for high-frequency RF component synthesis. Hence, we revise SMLO to enhance the synthesis ability. The original SMLO uses a surrogate model which can have errors, and the performances are related to the type of surrogate model and its corresponding parameters. If the synthesis fails by SMLO, the reason may be either the framework itself or a bad surrogate model, or even a bad search algorithm. In RSMLO, after the optimal solution of the coarse model (initial optimal point) is decided, we directly use EM simulations, which is analogous to using an absolutely accurate surrogate model, and the same optimization kernel as MMLDE. The search range is within a small deviation (e.g., 3%, 5%) from the initial optimal point [3], which is a common setting of SMLO.

A. Test Example 1

The first example is a 60 GHz inductor with circular shape in a 90 nm CMOS process for VCO design. The design variables are the inner diameter (din), the metal width (mw), and the metal spacing (ms). The number of turns (nr) is 1.5, because for most of the inductors in 60 GHz RF ICs, nr is 1 or 1.5. The ranges of the design variables are $din \in [30, 100]$, $mw \in [3, 10]$, $ms \in [3, 8]$ (all in μm). The design specifications (constraints) are the inductance $L \in [0.45, 0.5]$ nH and the self-resonance frequency $SRF > 100$ GHz. The goal is to maximize the quality factor Q. The results are shown in Table I. RCS is the number of designs satisfying the constraints over ten different runs. N is the number of evaluations (average for the ten runs). T is the average time of ten runs. The average performance for the optimization goal Q and the typical performances of the constraints (L, SRF) are provided for each method. For SRF we calculate the corresponding Q at 100 GHz. If Q at 100 GHz is larger than 0, the SRF is larger than 100 GHz.

It can be seen from Table I that the result of MMLDE is comparable to that of EMGO (the benchmark) and achieves an efficiency improvement of ten times. Both of the design solutions of MMLDE and EMGO (in the sequence of din, mw, ms) are near [5], [6], [28]. The standard deviation of MMLDE on the optimization goal, Q, is 0.14. Because the shape of the inductor is circular, the mesh density is set to 30 cells/wavelength and the Arc. resolution is set to 5° in this example. With these settings, the CPU time cost of each EM simulation is longer, but the accuracy is high. A typical result of MMLDE for the 60 GHz inductor is shown in Fig. 7.

Next, we look at the performances of the MMLDE framework only with the GP model and the EI measurement

TABLE I
RESULTS OF DIFFERENT METHODS FOR EXAMPLE 1

	EMGO	EMGOG	EMGOA	SEMGO	MMLDE
RCS	10/10	9/10	2/10	5/10	10/10
Q	14.7	14.5	15.1	15.4	14.5
L	0.46	0.45	0.42	0.40	0.46
SRF	>100G	>100G	<100G	>100G	>100G
N	356	92	Fail	96	40
T	19.1 h	5.0 h	Fail	4.9 h	2.1 h

As EMGOA failed to satisfy the constraints in most cases, we do not consider its clock time and write Fail.

(EMGOG) and only with the ANN model (EMGOA). For the EMGOA method, we can see that the synthesis does not succeed in many cases. This verifies that if only the predicted value is used in online surrogate-model-based optimization, it is very easy to be trapped in local optimum points and fail the synthesis. For the EMGOG method we see that in most cases the synthesis is successful, because the EI measurement considers global exploration. However, it is about 2.5 times slower than MMLDE. We found that the number of EM evaluations can show a large variation between different runs using EMGOG: sometimes the necessary number of EM simulations is less than 30, but sometimes it can increase to more than 180 for a similar final result. Hence, the memetic algorithm in MMLDE enhances the speed and the optimization ability considerably. We also compare with the method of SEMGO [2]. For the ANN we use eight steps for din, four steps for mw, and three steps for ms (in total 96 points for EM simulation). We normalize the input and output data to $[-1, 1]$ by means of linear scaling, and use the LM training algorithm. Different numbers of neurons and layers have been tested. We randomly select five ANNs from the group of trained ANNs whose training error is within 0.01. The results show that sometimes SEMGO can obtain a good result and sometimes it cannot. In the ten runs the results violate the specification on L for five times (according to the EM simulation results, not the ANN predictions). In those runs which provide infeasible solutions, the Q value is much higher. Hence, the average value of Q is higher. The reason of violating the L constraint is not only the training errors of the ANNs; another important reason is that the training data are not smooth enough, making some of the trained ANN overfitted and loosing generality [18]. These are inherent problems for ANNs. We can also find that SEMGO is nearly 2.5 times slower than MMLDE. The difference comes from the core idea of MMLDE, i.e., the adaptive generation of the training data, which leads to two advantages compared with SEMGO: 1) the time to perform EM simulations in non-promising areas is minimized, and 2) the ANN prediction is often used for local refinements, so it is more reliable and accurate. Even if the ANN in MMLDE is over-fitted, there is another learning mechanism, the Gaussian process, which can adjust the search direction.

B. Test Example 2

The second example is a 60 GHz overlay transformer [26] with octagonal shape in a 90 nm CMOS process. The design variables are the inner diameter of the primary inductor

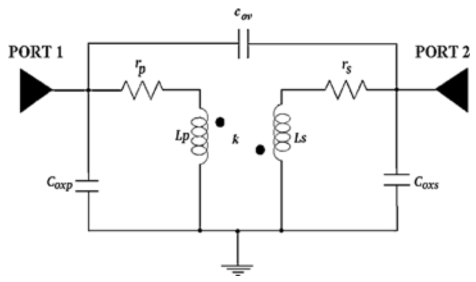

Fig. 8. Equivalent circuit model of a transformer used as coarse model.

TABLE II
RESULTS OF DIFFERENT METHODS FOR EXAMPLE 2

	EMGO	RSMLO	MMLDE
RCS	10/10	0/10	10/10
PTE	89.0%	86.1%	88.8%
N	965	Fail	87
T	24.7 h	Fail	2.3 h

As RSMLO using equivalent circuit model failed to satisfy the constraints in all the ten runs, we do not list its clock time.

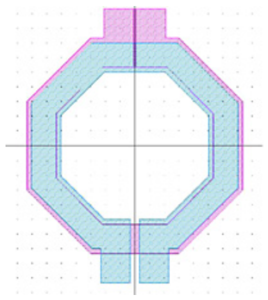

Fig. 9. Typical transformer result for Example 2.

($dinp$), the inner diameter of the secondary inductor ($dins$), the width of the primary inductor (wp), and the width of the secondary inductor (ws). The ranges of the design variables are $dinp, dins \in [20, 150]$, $wp, ws \in [5, 10]$ (all in μm). The design specifications are the coupling coefficient $k > 0.85$, the quality factor of the primary inductor $Q1 > 10$, the quality factor of the secondary inductor $Q2 > 10$. The output load impedance is 25 Ω, which is the input resistance of the following stage. The specifications of the input impedance (in 60 GHz) are $Re(Z_{in}) \in [10, 20]$ and $Im(Z_{in}) \in [10, 25]$ (Ω), which is the required optimal load impedance of the driver stage. The optimization goal is to maximize the power transfer efficiency (PTE). The coarse model selected for RSMLO is a widely used equivalent circuit of a transformer [27], which is shown in Fig. 8. The transformer synthesis results are shown in Table II. The average PTE for ten runs are provided for each method.

From Table II, the results can be analyzed. MMLDE costs 2.3 h on a single CPU node, which is very reasonable for practical use. Moreover, the result of MMLDE is comparable with the benchmark (the EMGO method), but MMLDE is more than ten times faster. The standard deviation of MMLDE on the optimization goal, PTE, is 0.25%. The mesh density is set to 30 cells/wavelength and the Arc. resolution is set to 45°. A typical transformer result of MMLDE is shown in Fig. 9.

We also see that the RSMLO method using a coarse equivalent-circuit model cannot satisfy the constraints. The design solutions of MMLDE and EMGO (in the sequence of $dinp$, $dins$, wp, ws) are close to [50, 53, 10, 10], but that of RSMLO is close to [123, 108, 10, 10]. The result is good when using equivalent circuit model, but when checking with fine EM simulations, the performances are far from the specifications. It can be seen that the search around the optimal point of the coarse equivalent-circuit model is not appropriate in this example. We then look at the coarse-mesh SMLO. The Nelder–Mead simplex direct search method [28] is used to perform local search. The starting point is the optimal solution of the above coarse equivalent-circuit model, which is better than a random start in most cases. For the mesh density, 10 cells/wavelength and 20 cells/wavelength are tried. Because the EM solver uses four threading, the time spent by using 10 cells/wavelength is about 86% of the time spent by using 30 cells/wavelength (the fine EM simulation). We use 150 coarse mesh model evaluations to find the initial optimal point, while MMLDE uses 87 fine EM simulations in the whole process. With the optimal point of the coarse model, a 5% deviation is used to set the search range and global optimization is performed using fine EM simulations in the selected range. As said above, this is like using an absolutely accurate surrogate model, whose solution quality must be better than the original SMLO which is affected by learning errors. Five runs are performed for each setting. The best results are: using ten cells/wavelength, the optimal point of the coarse mesh model is [111.956, 86.778, 11.653, 12.610], and after global optimization in the selected range, the performances [in the sequence of PTE, k, $Q1$, $Q2$, Z_{in} (in complex form)] are [86.91%, 0.75, 15.664, 15.361, 18.7081 + 60.929i]. Using 20 cells/wavelength, the initial optimal point is [114.633, 91.192, 11.736, 11.761], and the optimized performances are [87.06%, 0.762, 14.882, 15.029, 19.990 + 64.414i]. These results are much better than the result by the equivalent circuit model. However, the k and $Im(Z_{in})$ constraints are still not satisfied, though this will be an acceptable result if the constraints are loose. At last, we use the fine EM simulation and the Nelder–Mead simplex direct search method in the whole process, and the k and $Im(Z_{in})$ constraints also cannot be satisfied.

From this example we can see the two challenges of SMLO: the accuracy of the coarse model and the selection of the starting point. Note that "the starting point" is not the initial point to generate the base points, or the optimal result of the local optimization, but the starting point to run the local optimization. For the coarse equivalent-circuit model that is cheap in evaluation, we can do global optimization to it and no starting point is needed. However, the coarse model often has serious accuracy problems at high frequencies. For coarse-mesh EM models that are more expensive, when using SMLO we can only do local optimization to find the initial best point. Although the model is more accurate, the selection of the starting point for the local optimization becomes the new problem. This problem may seriously affect the performance of SMLO, as the starting point is very critical for local optimization algorithms. In contrast, MMLDE does not need any coarse model nor a good starting point, so overall it has a more reliable performance at a good computational cost for high-frequency component synthesis. The upper frequency limit of MMLDE is the frequency that fine EM simulations cannot be trusted, or the same as EMGO. The reason is that MMLDE only relies on accurate EM simulations, rather than on coarse models. For any frequency where the designer relies on the result of fine EM simulations, MMLDE is applicable. Note that SMLO is a good choice when the frequency is relatively low or the specifications are moderate. From the two difficult examples presented in this paper, it is clear that MMLDE and SMLO face different kinds of problems and therefore are suited for different applications.

V. Conclusion

In this paper, the MMLDE algorithm has been proposed for the optimized synthesis of passive components in high-frequency RF ICs. MMLDE can provide results that are comparable with the standard EMGO framework, which is the best framework in terms of solution quality, but at far lower computational cost (an order of magnitude smaller). Compared with the state-of-the-art methods, i.e., SMLO and SEMGO, MMLDE also showed clear advantages in optimization ability, accuracy, efficiency and robustness, as demonstrated by the presented 60 GHz examples. These results are achieved by the core idea of generating the training data adaptively to construct a dynamically online surrogate-model-based memetic evolutionary algorithm in combination with the techniques from evolutionary computation and machine learning. Future work will focus on developing MMLDE-embedded tools and introducing parallel computation to the MMLDE framework.

Acknowledgment

The authors sincerely thank B. Yang, B. Machiels, W.-T. Lo, and Y. He for valuable discussions.

References

[1] A. M. Niknejad and R. G. Meyer, *Design, Simulation and Applications of Inductors and Transformers for Si RF ICs*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000.

[2] S. K. Mandal, S. Sural, and A. Patra, "ANN and PSO-based synthesis of on-chip spiral inductors for RF ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 1, pp. 188–192, Jan. 2008.

[3] S. Koziel, "Surrogate-based optimization of microwave structures using space mapping and kriging," in *Proc. 39th Eur. Microw. Conf.*, Sep.–Oct. 2009, pp. 1062–1065.

[4] A. Nieuwoudt and Y. Massoud, "Variability-aware multilevel integrated spiral inductor synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2613–2625, Dec. 2006.

[5] S. Mandal, A. Goyel, and A. Gupta, "Swarm optimization based on-chip inductor optimization," in *Proc. Int. Conf. Comput. Devices Commun.*, Dec. 2009, pp. 1–4.

[6] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, "Space mapping: The state of the art," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 337–361, Jan. 2004.

[7] S. Koziel and J. W. Bandler, "Space mapping with multiple coarse models for optimization of microwave components," *IEEE Microw. Wirel. Compon. Lett.*, vol. 18, no. 1, pp. 1–3, Jan. 2008.

[8] J. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 420–435, Jan. 2004.

[9] S. Koziel and J. W. Bandler, "Space-mapping optimization with adaptive surrogate model," *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 3, pp. 541–547, Mar. 2007.

[10] D. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[11] D. Jones, "A taxonomy of global optimization methods based on response surfaces," *J. Global Optimization*, vol. 21, no. 4, pp. 345–383, Dec. 2001.

[12] T. J. Santner, B. J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. Berlin, Germany: Springer, 2003.

[13] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.

[14] W. Liu, Q. Zhang, E. Tsang, and B. Virginas, "Fuzzy clustering based Gaussian process model for large training set and its application in expensive evolutionary optimization," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 2411–2415.

[15] B. Liu, Y. He, P. Reynaert, and G. Gielen, "Global optimization of integrated transformers for high frequency microwave circuits using a Gaussian process based surrogate model," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[16] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, *DACE: A MATLAB Kriging Toolbox*. Lyngby, Denmark: Technical University of Denmark, 2002.

[17] A. Javadi, "Estimation of air losses from tunnels driven under compressed air using neural networks," in *Proc. 10th Int. Conf. Comput. Methods Adv. Geomechanics*, vol. 1. 2001, pp. 207–211.

[18] P. D. Wasserman, *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold, 1988.

[19] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[20] B. Liu, F. V. Fernandez, Q. Zhang, M. Pak, S. Sipahi, and G. Gielen, "An enhanced MOEA/D-DE and its application to multiobjective analog cell sizing," in *Proc. IEEE World Congr. Evol. Comput.*, Jul. 2010, pp. 1–7.

[21] J. Swidzinski and K. Chang, "Nonlinear statistical modeling and yield estimation technique for use in Monte Carlo simulations," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2316–2324, Dec. 2000.

[22] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms," Caltech Concurrent Comput. Program, California Instit. Technol., Pasadena, Tech. Rep. 158-79, 1989.

[23] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proc. 4th Annu. Conf. Evol. Programming*, 1995, pp. 135–155.

[24] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[25] B. Liu, F. V. Fernández, G. Gielen, R. Castro-López, and E. Roca, "A memetic approach to the automatic design of high-performance analog integrated circuits," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 14, no. 3, p. 42, May 2009.

[26] J. Long, "Monolithic transformers for silicon RF IC design," *IEEE J. Solid-State Circuits*, vol. 35, no. 9, pp. 1368–1382, Sep. 2000.

[27] T. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd ed. New York: Cambridge University Press, 2004.

[28] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

Bo Liu was born in Beijing, China, on September 23, 1984. He received the B.S. degree from Tsinghua University, Beijing, in 2008. Since 2008, he has been pursuing the Ph.D. degree and is a Research Assistant with MICAS Laboratories, Katholieke Universiteit Leuven, Leuven, Belgium, under the supervision of Prof. G. Gielen.

He has authored or co-authored more than 20 papers in international journals and conference proceedings. He is a Technical Consultant with Accelicon Technologies, Inc., Cupertino, CA. His current research interests include evolutionary computation, machine learning, fuzzy logic, and design automation methodologies of analog and RF integrated circuits.

Mr. Liu is a reviewer in artificial intelligence and analog design automation fields, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, *Information Sciences* (Elsevier), and *Integration, the VLSI Journal* (Elsevier). He is also a book reviewer of Elsevier and Bentham Science publishers.

Dixian Zhao received the B.Sc. degree in microelectronics from Fudan University, Shanghai, China, in 2006, and the M.Sc. degree in microelectronics from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 2009. Currently, he is working toward the Ph.D. degree with Katholieke Universiteit Leuven, Leuven, Belgium.

From late 2005 to 2007, he was with the Auto-ID Laboratory, Fudan University, Shanghai, designing the non-volatile memory for passive radio-frequency identification tags. From 2008 to 2009, he was an Intern with Philips Research, Eindhoven, The Netherlands, where he designed a 60 GHz beamforming transmitter for presence detection radar. From 2009 to 2010, he was with TU Delft, where he developed the 94 GHz wideband receiver for imaging radar. His current research interests include RF and millimeter-wave integrated circuit design for wireless communications.

Patrick Reynaert was born in Wilrijk, Belgium, in 1976. He received the Master of Industrial Sciences degree in electronics (ing.) from the Karel de Grote Hogeschool, Antwerpen, Belgium, in 1998, and both the Master of Electrical Engineering (ir.) and the Ph.D. degrees in engineering science (dr.) from Katholieke Universiteit Leuven (K. U. Leuven), Leuven, Belgium, in 2001 and 2006, respectively.

From 2001 to 2006, he was a Teaching and Research Assistant with the MICAS Research Group, Department of Electrical Engineering (ESAT), K. U. Leuven. From 2006 to 2007, he was a Post-Doctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. In 2007, he was a Visiting Researcher with Infineon, Villach, Austria. Since October 2007, he has been an Associate Professor with ESAT, K. U. Leuven, and a Staff Member with the ESAT-MICAS Research Group. His current research interests include CMOS power amplifiers and mm-wave CMOS integrated circuits.

Georges G. E. Gielen (S'87–M'92–SM'99–F'02) received the M.Sc. and Ph.D. degrees in electrical engineering from Katholieke Universiteit Leuven, Leuven, Belgium, in 1986 and 1990, respectively.

In 1990, he was a Post-Doctoral Research Assistant and Visiting Lecturer with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. In 1993, he was an Assistant Professor with Katholieke Universiteit Leuven, where he was promoted to Full Professor in 2000. He has authored or co-authored two books and more than 400 papers in edited books, international journals, and conference proceedings. His current research interests include the design of analog and mixed-signal integrated circuits, and especially analog and mixed-signal computer-aided design tools and design automation (modeling, simulation and symbolic analysis, analog synthesis, analog layout generation, analog, and mixed-signal testing).

Dr. Gielen received the 1995 Best Paper Award in the John Wiley *International Journal on Circuit Theory and Applications*, and was the 1997 Laureate of the Belgian Royal Academy on Sciences, Literature and Arts in the Discipline of Engineering. He received the 2000 Alcatel Award from the Belgian National Fund of Scientific Research for his innovative research in telecommunications, and won the DATE 2004 Best Paper Award. He served as an elected member of the Board of Governors of the IEEE Circuits and Systems (CAS) Society and as the Chairman of the IEEE Benelux CAS Chapter. He served as the President of the IEEE CAS Society in 2005.