Global Optimization of Integrated Transformers for High Frequency Microwave Circuits Using a Gaussian Process Based Surrogate Model

Bo Liu, Ying He, Patrick Reynaert, Georges Gielen

ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium. E-mail: {Georges.Gielen, Bo.Liu, Patrick.Reynaert, Ying.He}@esat.kuleuven.be

ABSTRACT

Design and optimization of microwave passive components is one of the most critical problems for RF IC designers. However, the state-of-the-art methods 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 fully depend on electromagnetic (EM) simulations, whose solution quality is high but are too expensive. To address the problem, a new method, called Gaussian Process-Based Differential Evolution for Constrained Optimization (GPDECO) is proposed. In particular, GPDECO performs global optimization of the microwave structure using EM simulations, and a Gaussian process (GP) based surrogate model is constructed ON-LINE at the same time to predict the results of expensive EM simulations. GPDECO is tested by two 60GHz transformers and comparisons with the state-of-the-art methods are performed. The results show that GPDECO can generate high performance RF passive components that cannot be generated by the available efficient methods. Compared with available methods with the best solution quality, GPDECO can achieve comparable results but only costs 20%-25% of the computational effort. Using parallel computation in an 8-core CPU, the synthesis can be finished in less than 0.5 hour.

Keywords

Transformer synthesis, Microwave components, Microwave design, Gaussian process, Surrogate model, Differential evolution

1. INTRODUCTION

Computer-aided design optimization methodologies of microwave components can be classified into three categories: (1) equivalent circuit model and global optimization algorithm-based (ECGO) methods [1,2], (2) electromagnetic (EM)-simulation and global optimization algorithm-based (EMGO) methods [3,4], and (3) surrogate model and local optimization algorithm-based (SMLO) methods [5-9].

The ECGO methods fully depend on the equivalent circuit model to obtain the performances of the microwave structure. Their advantage is high efficiency. However, equivalent circuit models available in the microwave area are often not accurate enough [7], especially when the frequency is high (e.g. higher than 60GHz). Hence, even with global optimization algorithms, the synthesis may also fail as the used equivalent circuit model may not reflect the performances of the microwave structure well.

The EMGO methods can provide accurate performances analysis of the microwave structure because of using EM simulations. Combined with global optimization algorithms, the quality of the solution is the best among all the available methods. However, its major bottleneck of high computational cost of the EM simulations limits its use in practice [5]. Reference [4] improves the efficiency of EMGO, which provides an off-line surrogate model-based EMGO method. It first trains an artificial neural network (ANN) to approximate the performance of the microwave structure, and then uses it in the optimization. The training data are generated uniformly in the design space and the corresponding performances are obtained by EM simulations. In this method, the training data generation process is expensive. Moreover, only a small part of the design space is useful in optimization, and many of these expensive EM simulations to cover the whole design space are wasted.

The SMLO methods [5-9] combine the efficiency of ECGO and the accuracy of EM simulations. 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*, 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), and the data received by the EM simulations will update the surrogate model to make it more accurate. Some works focusing on selecting the coarse model [5,7] and the surrogate model [5,8] have been proposed.

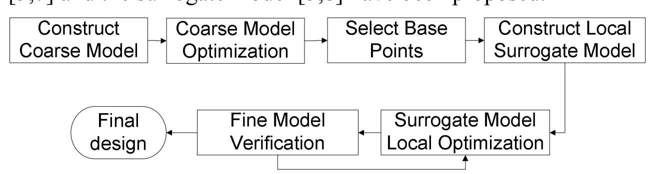

Fig.1. Flow of the SMLO methods

However, SMLO still highly depends on the accuracy of the coarse model, which leads to two significant challenges. First, the optimal solution of the coarse model defines the search space and the constructed surrogate model is only accurate in that space (as the base points are selected around it) [5]. 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 [5-9]. However, this assumption only holds when the coarse model is accurate enough. Although it has been shown that SMLO works well for microwave components synthesis problems in comparatively low frequency (e.g. 5GHz) [5-9], for passive components in high-frequency RF ICs (e.g. 60GHz), this assumption is often not true, as will be shown by the example 1 in section 3 (a 60GHz transformer). Therefore, the challenge of high

978-3-9810801-7-9/DATE11/©2011 EDAA

dependence on the accuracy of the coarse model limits the use of SMLO for the synthesis of high frequency RF components, because an accurate model in high-frequency is often difficult to find [7]. The second challenge is that SMLO can only do local search. 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 [5-9]; using global optimization algorithms then 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 for high-frequency microwave structures. EMGO can provide high quality results but is too expensive.

To address these problems, we propose a new framework, the Gaussian Process-Based Differential Evolution for Constrained Optimization (GPDECO) algorithm. The method aims to:

- achieve comparable results with EMGO (the best algorithm on the solution quality aspect) in high-frequency synthesis;
- highly improve the efficiency of EMGO and make the computational time practical.

We choose RF transformers to illustrate the approach, but the algorithm is also applicable to other microwave structures.

The remainder of the paper is organized as follows. Section 2 introduces the basic components, key techniques and the general framework of GPDECO. Section 3 tests GPDECO by practical examples. The concluding remarks are presented in Section 4.

2. THE GPDECO ALGORITHM

2.1 Key Ideas of GPDECO

Three conclusions can be drawn from the available methods: (1) Global optimization and EM simulations are the keys to obtain high quality solutions. (2) Using a surrogate model to predict the performances is the critical point to enhance the efficiency. (3) The reason why ECGO and SMLO highly depend on the accuracy of the coarse model is that the optimal solution and the performances of the coarse model play extremely vital roles. Motivated by these observations, GPDECO proposes a new framework. It performs global optimization of the microwave structure using EM simulations, and a Gaussian process (GP)-based surrogate model is constructed and updated ON-LINE at the same time to predict the results of future expensive EM simulations. Note that this surrogate model is not the surrogate model used in SMLO. This surrogate model is constructed and updated during the optimization of the fine model (or on-line) using the data from fine model evaluations, while the surrogate model in SMLO is constructed off-line using the data from coarse model evaluations. To the best of our knowledge, such framework has not been reported in the RF component synthesis field yet.

Then, we ask the following question: How to integrate the on-line surrogate model to global optimization algorithms? The answer, however, is *not* trivial. The challenge is that the quality of the surrogate model is improving gradually, as more training data are provided in the optimization process. Hence, to maintain the solution quality, some predicted performances cannot be used. But how to recognize if a prediction can be used remains an open question. Towards this goal, we propose a method of determining a threshold of uncertainty first to decide whether the prediction from the GP model can be used for a candidate or the expensive fine model evaluation is necessary. The threshold value is obtained by analyzing the data of the coarse model (section 2.4.1).

For the surrogate model type, we choose GP machine learning [10-13]. The reasons are that: (1) GP can provide an uncertainty measurement for a prediction, which is very useful when combined with optimization. (2) Unlike ANN, GP does not require a predefined structure. However, GP has the assumption of Gaussian distribution of the input and output data. Hence, normalization should be done on the training data. But for some problems, directly normalizing the design points X and performances Y may not yield good result. In many cases, using transformation functions, e.g. $\log(Y)$, will achieve better results. Whether to use transformation functions for each specification needs to be determined before constructing the GP-based surrogate model. We also decide that by analyzing the data of the coarse model (section 2.4.2).

It can be seen that GPDECO also uses a coarse model, but the purpose is to extract the necessary thresholds and transformation functions. Unlike SMLO, GPDECO does not use the optimal solution of the coarse model to define the search space and also does not use the coarse model performances-based surrogate model to guide the search, so the requirement of accuracy of the coarse model is much lower. This is also shown by the examples in section 3. The key ideas of GPDECO are shown in Fig. 2.

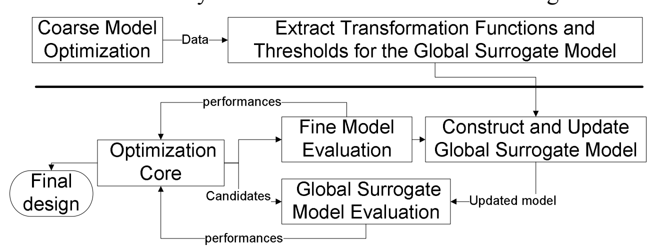

Fig. 2. Key ideas of GPDECO

For the optimization core, we choose evolutionary computation (EC) algorithms. It seems that it may cost more function evaluations compared with non-population-based algorithms. However, choosing EC is motivated by the following 3 considerations: (1) EC algorithms can achieve global optimization, which is the aim of this paper. (2) The GP-based surrogate model is constructed on-line, and a reliable GP model needs a certain amount of training data, which does not depend on which kind of optimization algorithm to use. When the GP model is well trained, most of the function evaluations are performed by GP. (3) The evaluations of individuals in the EC algorithms are independent from each other in a population, so it is very suited for parallel computation (although this paper only uses an 8-core CPU, more powerful parallel computation techniques are available), which can considerably enhance the efficiency.

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

2.2 Basics of Gaussian Process

We use a Gaussian process (kriging) model as the approximation method in GPDECO. Here we provide an intuitive introduction and the main formulas. More details are provided in [10,13].

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:

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

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 are in [14]. The values of μ, σ and θ are determined by maximizing the likelihood function of the observed data. Suppose there are n observed data, $x = (x_1, x_2 \cdots, x_n)$, and their corresponding function values are $y = (y_1, y_2 \cdots, y_m)$. The optimal values of μ and σ can be found by setting the derivatives of the likelihood function to 0 and solve the equations, which are as follows:

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

$$\hat{\sigma}^2 = (y - I\hat{\mu})^T R^{-1} (y - I\hat{\mu}) n^{-1} \tag{3}$$

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

$$R_{i,j} = Corr(x_i, x_j),\ i, j = 1, 2, \cdots n\,. \tag{4}$$

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):

$$\hat{y}(x^*) = \hat{\mu} + r^T R^{-1} (y - I\hat{\mu}) \tag{5}$$

where $r = [Corr(x^*, x_1), Corr(x^*, x_2), \cdots, Corr(x^*, x_n)]^T$ (6)

The measurement of the uncertainty of the prediction (mean square error (MSE)), which is used to access the model accuracy, can be described as:

$$MSE(x^*) = \hat{\sigma}^2 [I - r^T R^{-1} r + (I - r^T R^{-1} r)^2 (I^T R^{-1} I)^{-1}] \tag{7}$$

In this work, we use the DACE toolbox [14] to implement the Gaussian process-based surrogate model.

2.3 The Optimization Kernel

Synthesis of microwave structures is a constrained optimization problem. For example, maximize the efficiency under the constraint of input impedance for a transformer. The optimization engine and the constraint handling method are the two key factors in a constrained optimization algorithm. In this paper, the DE algorithm [15] is selected as the global search engine. The DE algorithm outperforms many EC algorithms in terms of solution quality and convergence speed. More details are in [15].

GPDECO uses a selection-based method to handle constraints [16]. The selection rules are: (1) Given two feasible solutions, select the one with the better objective function value; (2) Given two infeasible solutions, select the solution with the smaller constraint violation; (3) If one solution is feasible and the other is not, select the feasible solution. The most important advantages are that this method needs no penalty coefficient and is efficient. The advantages of DE combined with the selection-based constraint handling method have been shown in [17].

2.4 Constrained Optimization with Gaussian Process-based Surrogate Model

2.4.1 Determination of the MSE thresholds

First, as described in section 2.1, the surrogate model needs to be constructed based on the data by the fine model (high-fidelity EM simulations). Hence, unlike previous works, the accuracy and reliability of the GP model is improved gradually in the process of optimization, as more new training data are provided in the optimization process. This leads to a problem that some data predicted by the GP model may have large differences compared with fine simulation (especially in the beginning of the optimization when there are not enough training data), and cannot be used. Fortunately, by using GP, the uncertainty measurement MSE (eqn. 7) of each prediction can be a judgment. If the MSE value for a prediction is higher than a certain threshold, we discard the predicted value and fine simulation will be performed; otherwise, it means that the error is in an acceptable region, so the prediction will be used in the evolution process. But how to decide the value of the threshold is a problem. A large threshold will cause predictions with low accuracy to be the fitness value of some individuals, which may mislead the direction of evolution or even fail it. A small threshold can maintain the accuracy, but it will also discard predictions that can be used, so the efficiency cannot be enhanced much. In GPDECO, we extract the MSE thresholds by observing the data generated by the coarse model. Although the coarse model is not accurate, it at least represents the same problem as the fine model. Hence, we can reasonably assume that the range of the input/output, the severity of the constraints and the convergence process are similar. Therefore, using the data from the coarse model to extract the thresholds is reliable. This is also shown through experiments with different problems and constraints. Section 3 provides two examples.

We will now explain how it works. First, we use the coarse model with the same objective function and constraints to do the optimization. After each iteration, the GP model is trained based on the available data and predicts the performances of the candidate solutions in the next iteration (but the predicted performances are not used in the optimization). After this step, we have all the individuals that appeared in the coarse model optimization and their performances of both coarse model evaluation and GP model prediction, which are the source to extract the thresholds. Note that the fine model is not considered in this step, and this surrogate model trained by coarse model performances are not used in the synthesis step of GPDECO. In addition, this step is very cheap, because an equivalent circuit is used as the coarse model.

Then, we split the selection of the MSE thresholds into two trade-off problems and provide a general rule. First, the distance thresholds between the values predicted by GP and the values by the coarse model simulation are decided. If the distance is within the threshold, the predicted values will be used. The larger the distance thresholds values, the highly the inaccuracy of the prediction. The smaller the distance thresholds values, the less predictions that can be used. Hence, there is a trade-off. For constraints, we use the misjudging rate to reflect the accuracy. The number of misjudgments is the sum of individuals whose predicted values satisfy a constraint but for which the true simulation value does not or vice versa. The misjudging rate is the ratio of the number of misjudgments to the number of all individuals. Fig. 3(a) shows the curve of a constraint in example 2 section 3, the quality factor of the primary inductor $Q1 > 10$. According to experiments with different transformers and different constraints, the distance threshold with a misjudging rate less than 2% often gives a good accuracy. In this range, we can maximize the predictions that can be used. It can be seen from Fig. 3(a) that with a misjudging rate of 1.2%, the individuals that can use GP prediction to replace fine simulation is 94% of all the individuals. For the objective function, the distance itself can reflect the accuracy. According to experiments, setting the distance threshold within 1% to 3% of the maximum value that the objective function can reach often performs well.

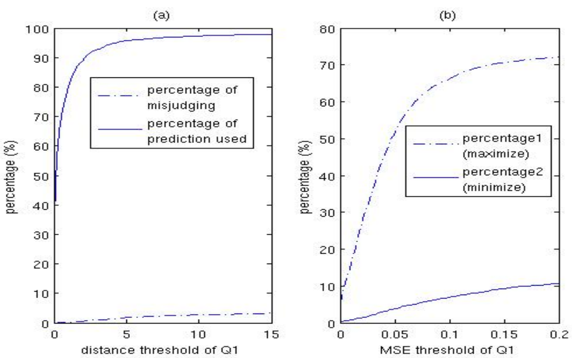

Fig. 3. Two trade-offs to decide the MSE thresholds (Q1, example 2)

After the distance thresholds are decided, we can decide the MSE thresholds (used in GPDECO). The MSE value provided by GP prediction for each candidate is an estimation of uncertainty, which can generally reflect the distance described above, but it is not true that if the MSE value of a performance is within a threshold, its distance must be within the corresponding distance threshold, or vice versa. Under a determined distance threshold, for different MSE thresholds, there are some individuals whose MSE values are within a certain threshold, and their distances are also within the given distance threshold, and there also exist some individuals whose MSE values are within a certain threshold but their distances are out of the given distance threshold. Both percentages can be calculated. We want to maximize the first percentage and minimize the second percentage. A trade-off can be made by using the two corresponding percentages of different MSE thresholds. Fig. 3(b) shows the curve of the Q1 in example 2 section 3 under a determined distance threshold. According to experiments, the second percentage is suggested to be within 10% to maintain the reliability of the mapping, and then we can choose an MSE threshold to maximize the first percentage. It can be seen that there are parameters that need to be determined empirically, e.g. the misjudging rate upper limit. These parameters are decided by testing different transformers and different kinds of constraints. Once set, they do not change.

2.4.2 On-line surrogate model-based optimization

With the on-line constructed GP model and the decided MSE thresholds, the surrogate model, whose predictions can be judged to decide if they can be used or EM simulations are needed, is able to be integrated in the optimization flow. However, two additional considerations are necessary. First, even when the GP model predicts the performances well (within the MSE thresholds), there still exists a small error, so the final result must be verified by the EM simulation. Hence, in GPDECO, we verify the best candidate in the population in each generation with EM simulations. Second, we use transformation functions in the training of the GP model. Because the GP model predicts the performance values by the correlation of the design variables and the performance functions values, input / output data in a large range will make the training more difficult. A possible solution is to use transformation functions (e.g. $\log_m^x$) to decrease the range of the input / output variables.

2.5 The General Framework of GPDECO

Based on the above ideas, the GPDECO algorithm can be constructed. The detailed flow diagram is shown in Fig. 5.

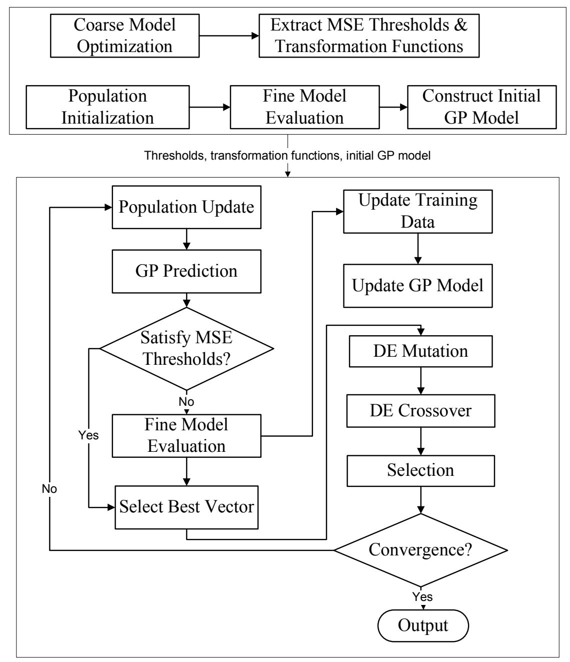

Fig.5. Flow diagram of GPDECO

The algorithm consists of the following steps.

Step 0: Optimize the coarse model and extract transformation functions and the MSE thresholds according to the method described in section 2.4.

Step 1: Initialize the population randomly, perform fine model evaluations to the individuals and construct the GP model.

Step 2: For a new population of candidates, use the GP model to predict the performances (eqn. 5) and the corresponding MSE values (eqn. 7).

Step 3.1: For the individuals whose MSE values satisfy the thresholds, use the predicted values and go to step 4.

Step 3.2: For the individuals whose MSE values do not satisfy the thresholds, use fine model evaluations. Add these data to the training data set and update the GP model. Go to step 4.

Step 4: Select the current best candidate using the criterion described in section 2.3.

Step 5: Perform the DE mutation and crossover operation [15] to obtain each individual's trial individual.

Step 6: Perform selection between each individual and its corresponding trial counterpart according to the selection rule described in section 2.3 to update the population.

Step 7: If the stopping criterion is met (e.g. a convergence criterion or a maximum number of generations), then output X_{best} and its performance values; otherwise go back to Step 2.

3 Experimental Results and Comparisons

In this section, the GPDECO algorithm is demonstrated by two 60GHz transformers in a 90nm CMOS technology. Example 1 is a constraint satisfaction problem and example 2 is a constrained optimization problem. The population size is 20, the crossover rate is 0.8 and the DE step size is 0.8, which is a common setting [15]. GPDECO stops when the performance cannot be improved for 30 consecutive generations. The examples are run on a PC with 8 cores 12GB RAM and Linux operating system. SONNET is used as the EM simulator. For the mesh settings, we use the conformal fill type and the maximum length is set to 750. The performance of evolutionary algorithms (EA) may be affected by the random numbers used in the evolution operators. Therefore, 10 runs with independent random numbers are performed for all the experiments and the results are analyzed and compared statistically.

We select two reference methods for the comparisons. The first one is EMGO with the same optimization kernel but fully depends on the EM simulations. The purpose is to provide the best result to test the other methods. Obviously, it is the most CPU expensive method. The second reference method is a revised SMLO (RSMLO). SMLO clearly has the best efficiency. But our goal of GPDECO is high synthesis ability for high-frequency RF components and reasonable computational time. So comparing the speed with SMLO is not our purpose unless SMLO also receives an acceptable result in high-frequency synthesis. Hence, we revise SMLO to enhance the synthesis ability. The original SMLO uses a surrogate model which has errors, and the performances are related to the kind of surrogate model and the 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 (starting point) is decided, we directly use EM simulations (it is analogous to using an absolutely accurate surrogate model) and the same optimization kernel as GPDECO. The search range is within a 3% deviation from the starting point [5], which is a common setting of SMLO.

3.3 Example 1

The first example is a 60GHz stacked transformer with circular shape in a 90nm CMOS process. The design variables are the inner diameter of the primary inductor (*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 $dinp, dins \in [20,150]$, $wp, ws \in [5,10]$ (in μm). The design specifications are the coupling coefficient k > 0.4, the quality factor of the primary inductor $Q1$ > 10, the quality factor of the secondary inductor $Q2$ > 10 and the efficiency *Gmax* > 70%. The output impedance is $25\ \Omega$. The input impedance is described in a complex number form. The specifications of the input impedance are $\mathrm{Re}(Z_{in}) \in [5,20]$ and $\mathrm{Im}(Z_{in}) \in [10,60]$. The coarse model of both RSMLO and GPDECO we selected is a widely used equivalent circuit of a transformer [18], which is shown in Fig. 6. The results are shown in Table 1. *SR* is the successful rate in 10 runs. *N* is the number of evaluations. Typical performances for 10 runs are provided for each method.

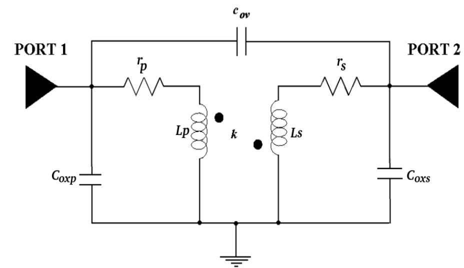

Fig. 6. Equivalent circuit model of a transformer used as coarse model

Table 1. Results of different methods to example 1

	EMGO	RSMLO	GPDECO
Gmax	71.46%	70.75%	71.40%
k	0.42	0.56	0.42
Q1	12.79	7.73	12.91
Q2	14.67	8.87	14.60
Z_{in}	9.24+56.88j	19.22+88.28j	9.37+54.06j
SR	**10/10**	**0/10**	**10/10**
Average N	636	failed	151
Average Time	**1.02 hours**	**failed**	**0.24 hours**

(As RSMLO fails to satisfy the constraints in all the 10 runs, we do not consider the time of it)

From Table 1, it can be seen that EMGO and GPDECO satisfy all the constraints (this example is a constraint satisfaction problem, so there is no design goal) in all the 10 runs. GPDECO finishes the synthesis in 0.24 hours and speeds up EMGO for 4.3 times. It can be seen that the computational time of GPDECO is very reasonable for practical use. On the other hand, RSMLO fails in all the 10 runs. We can find the reason if we compare the optimal point of the coarse model and the fine model. The optimal point of the coarse model is near [*dins, dinp, ws, wp*] = [48, 54, 8, 10]. The corresponding performances using EM simulation are near [*Gmax, k, Q1, Q2,* Z_{in}] = [75%, 0.72, 6.16, 6.33, 22.84+72.66j]. It is shown from Table 1 that the results are improved by RSMLO, but are still far from the specifications. The design solutions of GPDECO are near [*dins, dinp, ws, wp*] = [30, 40, 9, 8]. Most of the solutions of GPDECO and EMGO converge to similar points. It can be seen that this design point is far from the above optimal point of the coarse model. On the contrary, in references [5-9] using the SMLO algorithms, the optimal point of the coarse model and the fine model are very near. This difference makes the RSMLO method fail in this high-frequency RF component synthesis example, because an accurate enough coarse model is difficult to find for high frequency RF components. In contrast, GPDECO also uses the same coarse model, but receives good results because its requirement of accuracy is much lower. The synthesized transformer is shown in Fig. 7.

3.4 Example 2

The second example is a 60GHz stacked transformer with octagonal shape in a 90nm CMOS process. The design variables are *dinp*, *dins*, *wp* and *ws*. The ranges of the design variables are the same as example 1. The design specifications are $k > 0.85$, $Q1 > 10$, $Q2 > 10$, $\mathrm{Re}(Z_{in}) \in [10, 20]$ and $\mathrm{Im}(Z_{in}) \in [10, 25]$. The output impedance is 25 Ω. Besides the differences of the constraints compared with example 1 (k is very high, the range of Z_{in} is drastically narrowed down), this example is a constrained optimization problem, which is more difficult than constraint satisfaction problem, and is a good example to test the optimization ability of GPDECO. The optimization goal is to maximize the efficiency, *Gmax*. The results are shown in Table 2 (*Gmax* is the average value of the 10 runs). *RCS* is the number of designs satisfying the constraints over 10 runs.

Table 2. Results of different methods to example 2

	EMGO	GPDECO
RCS	**10/10**	**10/10**
Gmax	86.1%	86.1%
Average N	**979**	**202**
Average Time	**1.4hours**	**0.3hours**

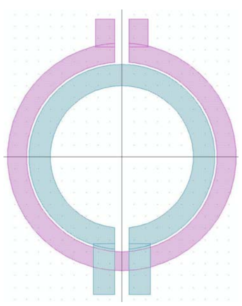

Fig. 7. A typical result (example 1)

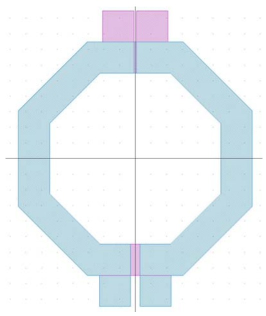

Fig. 8. A typical result (example 2)

From Table 2, it can be seen that the result of GPDECO almost has no loss compared with the benchmark (EMGO) method, but it is nearly 5 times faster. GPDECO costs 0.3 hours. Note that the reason why the time consumption of EMGO seems not very long is because parallel computation using an 8-core CPU is applied. If without parallel computation, EMGO will cost very long time. A typical result of GPDECO is shown in Fig. 8. The design solutions of GPDECO and EMGO are close to [54.7, 54.8, 10, 10].

4 CONCLUSIONS

In this paper, the GPDECO algorithm has been proposed for the surrogate-model-based synthesis of RF transformers working in high frequency. GPDECO solves the limitation of high dependence on the accuracy of the coarse model. The examples show that GPDECO can synthesize high-performance transformers that cannot be generated by the current SMLO framework. In addition, GPDECO has similar quality of results compared with the EMGO framework, which is the best framework in terms of solution quality. Yet GPDECO only costs 20%-25% of the computational effort, and therefore makes the synthesis has a good efficiency. The gain comes from the newly proposed framework of a global optimization algorithm using EM simulations and an ON-LINE GP-based surrogate model, the proposed method of extracting MSE thresholds, as well as the parallel computation using a multi-core CPU.

5 ACKNOWLEDGMENTS

This research was supported by a special bilateral agreement scholarship of Katholieke Universiteit Leuven, Belgium and Tsinghua University, P. R. China. We sincerely thank Mr. Chao Li, Mr. Bohan Yang and Miss. Wan-Ting Lo, ESAT-MICAS, K. U. Leuven, for valuable discussions.

6 REFERENCES

[1] A. Nieuwoudt et al., 2006. "Variability-Aware Multilevel Integrated Spiral Inductor Synthesis", *IEEE TCAD,* pp. 2613-2625.

[2] S. Mandal et al., 2009. "Swarm Optimization Based On-Chip Inductor Optimization", Proc. International Conference on Computers and Devices for Communication, pp. 1-4.

[3] A. Niknejad et al., 2000. Design, Simulation and Applications of Inductors and Transformers for Si RF ICs. Kluwer Academic Publishers.

[4] S. K. Mandal et al., 2008. "ANN- and PSO-Based Synthesis of On-Chip Spiral Inductors for RF ICs", *IEEE TCAD*, pp. 188-192.

[5] S. Koziel et al., 2009. "Surrogate-Based Optimization of Microwave Structures Using Space Mapping and Kriging", *Proc.of 39th European Microwave Conference*, pp. 1062-1065.

[6] J. Bandler et al., 2004. "Space Mapping: The State-of-the-Art", *IEEE MTT*, pp. 337-361.

[7] S. Koziel et al., 2008. "Space Mapping With Multiple Coarse Models for Optimization of Microwave Components", *IEEE Microwave and Wireless Components Letters*, pp. 1-3.

[8] J. Rayas-Sanchez et al., 2004. "EM-Based Optimization of Microwave Circuits Using Artificial Neural Networks: The State-of-the-art", *IEEE MTT*, pp. 420-435.

[9] S. Koziel et al., 2007. "Space-Mapping Optimization with Adaptive Surrogate Model", *IEEE MTT*, pp. 541-547.

[10] T. Santner et al., 2003. "The Design and Analysis of Computer Experiments", Springer.

[11] M. Emmerich et al., 2006. "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels", *IEEE TEVC*, pp. 421-439.

[12] W. Liu et al., 2009. "Fuzzy Clustering Based Gaussian Process Model for Large Training Set and Its Application in Expensive Evolutionary Optimization", *Proc. of IEEE Congress on Evolutionary Computation*, pp. 2411-2415.

[13] D. Jones, 2001. "A Taxonomy of Global Optimization Methods Based on Response Surfaces", *Journal of Global Optimization*, pp. 345-383.

[14] S. N. Lophaven et al., 2002. "DACE: A MATLAB Kriging Toolbox", Technical University of Denmark.

[15] K. Price, et al., 2005. Differential Evolution. A Practical Approach to Global Optimization. Springer, Berlin, Heidelberg, New York.

[16] K. Deb, 2000. "An Efficient Constraint Handling Method for Genetic Algorithm", *Computer Methods in Applied Mechanics and Engineering*, pp. 311-338.

[17] B. Liu, et al., 2009. "A Memetic Approach to the Automatic Design of High Performance Analog Integrated Circuits", *ACM Trans. on Design Automation of Electronics Systems*, 14(3), Article 42.

[18] T. Lee, 2004. "The Design of CMOS Radio-Frequency Circuits (second edition)", Cambridge University Press.