

Adaptive Synthesis Using Hybrid Genetic Algorithm and Particle Swarm Optimization for Reflectionless Filter With Lumped Elements

Yifan Li¹, Graduate Student Member, IEEE, and Xun Luo¹, Senior Member, IEEE

Abstract—In this article, an adaptive synthesis based on the hybrid genetic algorithm and particle swarm optimization (HGAPSO) is proposed for reflectionless filter design with lumped capacitors, resistors, and inductors. The synthesis starts with a preset topology, where each branch of the topology represents a small passive network of lumped elements. The proposed HGAPSO is used to trim branches and obtain proper values of elements for a required filtering response. Focus on this synthesis model, the HGAPSO is embedded with local searching policies based on random coordinate and neighborhood search to improve its searching ability. Besides, a classifier-based strategy and a probabilistic method are introduced to accelerate convergence and boost iteration. Suitable topologies and component values are determined automatically by the HGAPSO to meet the specific filtering response. To predict the response accurately, the EM-simulated result of the corresponding layout and parasitic parameter models of lumped elements are considered during the fine-tuning. Based on the mechanisms mentioned above, four reflectionless bandpass filters (BPFs) are synthesized to validate the effectiveness of the proposed synthesis procedure. The fabricated filters exhibit good selectivity and low reflection coefficient in the measurement.

Index Terms—Adaptive synthesis, gradient descent (GD), hybrid genetic algorithm and particle swarm optimization (HGAPSO), nonconvex optimization, reflectionless filter.

I. INTRODUCTION

FILTER plays a critical role in microwave systems. It is widely used to select the desired signals and suppress the interferences at unwanted frequencies. Conventional filters have relatively consummate design procedures based on lumped elements, transmission lines, or coupled resonators [1], [2], [3]. However, such filters are usually reflective, where the unwanted out-of-band interferences are reflected back to the input terminals. Those reflected signals generate additional

intermodulation products at the nonlinear front-end circuit (e.g., mixer). Meanwhile, the reactive input impedance of a reflective filter leads to system instability once the filter is connected with an amplifier [4], [5], [6], [7]. Reflectionless filters using lumped components [4], [5] are developed to absorb the interferences at unwanted frequencies. Then, the performance of microwave systems can be improved when active modules are connected with reflectionless filters. Besides, the reflectionless filters replace the conventional ones with isolators or circulators in integrated systems, which improve system integration and reduce design cost [7], [8], [9]. To improve the performance of reflectionless filters, two main methods are proposed. By using two networks with complementary frequency responses, the reflectionless operation can be obtained at one port [10], [11], [12], [13], [14]. Such filters usually exhibit the merits of low reflection, high selectivity, and wide stopband. The other method is using symmetrical topologies to implement dual-port reflectionless filters [14], [15], [16], [17], [18], [19], [20], [21], [22]. Such filters are featuring two-port reflectionless operation in a wide frequency range. Based on lumped elements, filters with ideal reflectionless response have been developed with specific topologies [23], [24]. However, most of them require a number of inductors or transformers. The non-negligible parasitic parameters have a significant influence on the matching and degrade the reflectionless performance of manufactured filters. Meanwhile, transformers have larger sizes compared with other lumped components, which limits the placement of components and circuit connections on the PCB. To overcome the design challenges caused by parasitics and reduce the time consumption of finding suitable circuit topologies, an adaptive synthesis procedure is necessary.

Recently, adaptive optimization methods [25], [26], [27], [28] are introduced for filter design. In optimizing a filter with several design parameters and constraints, there are various solutions that can approach the desired requirements. Therefore, the target function has multiple local optimums, which makes it nonconvex. Up to now, global nonconvex optimization is still a great challenge and no algorithm can guarantee to find the best solution without traversing all possible solutions. Even so, heuristic methods, e.g., particle swarm optimization (PSO)-based algorithms [29], show remarkable performance in circuit synthesis and optimization [25], [28]. In [25], PSO has a significant role to optimize the coupling

Manuscript received 29 November 2022; revised 15 February 2023 and 17 April 2023; accepted 4 May 2023. Date of publication 23 May 2023; date of current version 5 December 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFE0205600, in part by the National Natural Science Foundation of China under Grant 61934001 and Grant 62161160310, and in part by the Shenzhen Science and Technology Program under Grant JCYJ20210324120004013. (Corresponding author: Xun Luo.)

The authors are with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC), Shenzhen 518110, China, and also with the Center for Advanced Semiconductor and Integrated Micro-System, UESTC, Chengdu 611731, China (e-mail: xun-luo@ieee.org).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMTT.2023.3276212>.

Digital Object Identifier 10.1109/TMTT.2023.3276212

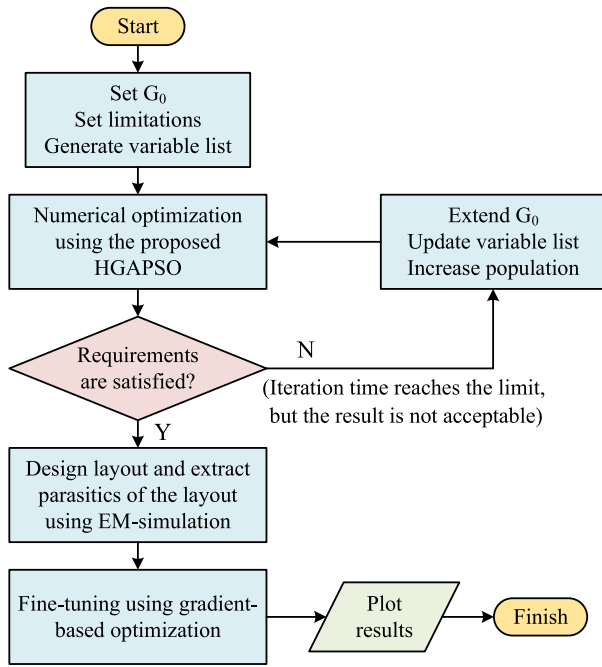


Fig. 1. Flowchart of the whole synthesis process.

matrix for designing coupled-resonator filters. Nevertheless, in high-dimensional solution space, particles are always sparse, which makes particles miss the global optimum and be congregated to a local optimum. To overcome this drawback, PSO and genetic algorithm (GA) are combined to form the hybrid genetic algorithm and particle optimization (HGAPSO) [30], [31], [32]. The HGAPSO features the advantages of searching heuristically and reduces the probability of being trapped into local optimums. Nonetheless, once a time-consuming target function is used, PSO-based algorithms are slow in iteration. Introducing gradient-based local search is an intuitive method to accelerate PSO [33]. However, in high-dimensional optimization, calculating gradient vectors still significantly decreases the speed of iteration. To avoid calculating gradients, algorithms using classifiers have been reported, which show high efficiency in high-dimensional optimization [34], [35].

The purpose of this work is to achieve the adaptive synthesis of proper topologies for reflectionless filters. In the implemented procedure, the inputs are constraints on the filtering response and range of component values, while the outputs are the synthesized topology and proper values of all components. The whole synthesis procedure is shown in Fig. 1. This procedure starts with a preset graph G_0 , and circuits are represented by trimming branches of G_0 . Constraints on G_0 can be applied to prevent complicated circuit structures and avoid unnecessary parasitics from the layout. After introducing the general branch model, the trimming procedure is realized by changing the variables' values. Such operations convert the synthesis into a high-dimensional nonconvex optimization. Focusing on this optimization, an enhanced HGAPSO is proposed, which embeds local search strategies and improves the heuristics ability of PSO-based algorithms. Meanwhile, a sorting-based classifier is introduced to accelerate convergence, and probabilistic methods are used for reducing

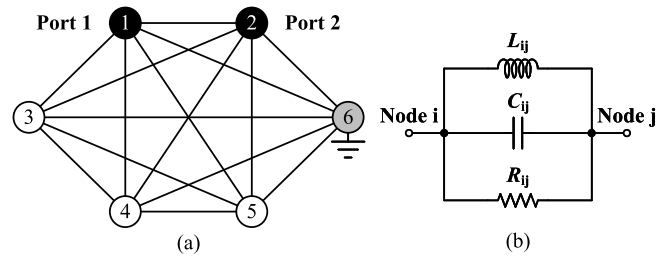


Fig. 2. (a) Complete graph K_6 that is used as an example of G_0 . (b) Circuits model for each branch in a graph.

unnecessary calculations. Once the preset graph G_0 is able to cover feasible topologies, the proposed HGAPSO can obtain the desired circuit. If the optimization could not get a satisfying performance after a long iteration, the optimization should be paused. Then, the graph is extended to improve its capability of expressing circuits and the optimization procedure is restarted. When the optimization using the proposed HGAPSO is finished, a suitable topology and corresponding values of all components are generated. After the layout is designed, a gradient-based algorithm is utilized for fine-tuning components' values. The influence of the parasitic parameters from the layout and components is considered to predict the response of practical filters accurately, which ensures the fabricated filters with expected characteristics.

The rest of this article is organized as follows. In Section II, the proposed general branch model and related network theory are introduced. Then, the procedure for constructing the target function which maps circuits into vector space is described. The details of the proposed HGAPSO algorithm and gradient-based fine-tuning method are introduced and compared with traditional algorithms in Section III. In Section IV, the schematic, layout, and response of synthesized reflectionless filters are presented and discussed. Those filters are fabricated and measured for verification. In Section V, the conclusions of this article are provided.

II. PROPOSED MODEL AND RELATED THEORY

A. Expressing Circuits in Graph With Proposed General Branch Model

As mentioned in [3] and [36], a passive lumped circuit can be expressed using an undirected graph described by an incidence matrix. To avoid a variable number of components changing the dimensionality of the solution space, a predefined initial graph G_0 is used to fix the total number of components. G_0 can be a complete graph, e.g., the graph K_6 as shown in Fig. 2(a), to guarantee the ability to synthesize any graph with an equal or fewer number of nodes. In this article, for a graph of N nodes, port 1 is defined as node 1, port 2 is defined as node 2, and ground is defined as node N (i.e., the ground-node). For each pair of node i and node j , the corresponding branch b_{ij} is composed of three parallel components, i.e., a resistor R_{ij} , an inductor L_{ij} , and a capacitor C_{ij} , as shown in Fig. 2(b). Thus, the admittance of branch b_{ij} is expressed as

$$y_{ij}(\omega) = \frac{1}{R_{ij}} + \frac{1}{j\omega L_{ij}} + j\omega C_{ij}. \quad (1)$$

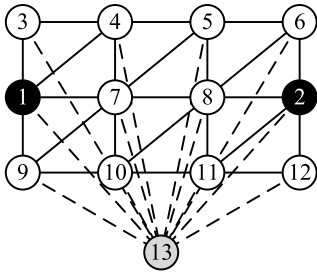


Fig. 3. Example of a quasi-planar graph that has 12 nodes aligned in a grid and an extra node utilized as the ground node. (The subgraphs connected by branches drawn in solid lines are planar graphs. Branches drawn in dashed lines are ground-connecting branches.)

The value of each component is limited to ensure the synthesized result is manufacturable. R_{\min} , L_{\min} , and C_{\min} are the lower bound of possible prototype value for resistors, inductors, and capacitors, while R_{\max} , L_{\max} , and C_{\max} are the upper bound. The components' values smaller than the lower bound are set to positive infinitesimal. Meanwhile, the components' values are set to infinite once the upper bound is exceeded. Therefore, by changing the value of components, each branch can approach all situations summarized as follows.

- 1) *Trimmed Branch*: $R_{ij} \rightarrow \infty$ and $L_{ij} \rightarrow \infty$ and $C_{ij} \rightarrow 0^+$.
- 2) *Branch Shorts the Connected Nodes*: $R_{ij} \rightarrow 0^+$ or $L_{ij} \rightarrow 0^+$ or $C_{ij} \rightarrow \infty$.
- 3) *Normal Branch*: Other than 1) and 2), $R_{ij} \in [R_{\min}, R_{\max}]$ or $L_{ij} \in [L_{\min}, L_{\max}]$ or $C_{ij} \in [C_{\min}, C_{\max}]$.

The number of required components grows with the number of branches. For a complete graph K_n , the number of branches is calculated using the following:

$$N(K_n) = \frac{n(n-1)}{2}. \quad (2)$$

Thus, a massive number of branches are introduced once a large n is used. Meanwhile, variables of high dimensionality lead to a large amount of computation, which slows down the synthesis. Therefore, except for the complete graph K_n , other graphs are used as G_0 . Besides, if the synthesized topology is planar (i.e., all components can be placed on the same side of the PCB, and no crossover is needed in the signal path), parasitics introduced by the interconnections are reduced. G_0 is initially set as a planar graph. However, for the PCB with metal layers at both sides, the whole bottom layer is used as the ground. To take advantage of the bottom layer, a type of quasi-planar graph is introduced. In the quasi-planar graph, the subgraphs without the ground node are planar graphs, and each node has an individual branch connecting the ground node. Such quasi-planar graphs are used as G_0 since it is acceptable to connect the ground plane using vias. Thus, parasitics from vias and long wires across multiple metal layers in the signal paths are avoided. An example of such a quasi-planar graph with 13 nodes (including the ports and ground node) is presented in Fig. 3. In this graph, all nodes are placed and connected in a grid. Then, diagonal branches are added to extend the flexibility in the synthesis. Compared to

a complete graph of the same node number, such graphs have fewer branches, and the number of total variables is reduced.

Graphs with more nodes are needed to synthesize circuits with higher order and larger scale. Meanwhile, if the target filter is a bandpass filter (BPF) or a bandstop filter (BSF), and the desired response is symmetric about the center frequency, the response can be converted to a low-pass type. Thus, a low-pass filter (LPF) is firstly synthesized. Then, the desired BPF/BSF is converted from the synthesized LPF prototype [1]. During the conversion, each inductor in the LPF is replaced with a series-connected LC pair, while each capacitor is replaced with a parallel-connected LC pair. Therefore, synthesizing LPF requires G_0 with a smaller scale compared to BPF/BSF, which reduces the total computation and improves the synthesis efficiency.

B. Mapping Circuits Into a Vector Space: Element Denormalization, Circuit Solver, and Assessment

In this section, a vector \vec{X} is utilized to represent the components' values, where the dimensionality of the vector is D . And D is the same as the number of total components in G_0 . Note that the value of element x_i in vector \vec{X} is limited from 0 to 1. Therefore, a circuit can be represented by G_0 and \vec{X} . Then, the target function $f(\vec{X}, G_0)$ is introduced. $f(\vec{X}, G_0)$ includes the procedures of solving the circuit, assessing the response, and indicating the discrepancy between the response and constraints. For simplification, $f(\vec{X}, G_0)$ is abbreviated as $f(\vec{X})$ since G_0 is consistent in a single optimization.

To assess the response of the circuits, the scattering matrix is obtained. Before calculating scattering parameters, \vec{X} is denormalized to get the actual values of components. In consideration of the finite practical component's value, a piecewise function is introduced. The actual value $p(x_i)$ of component x_i is expressed as

$$p(x_i) = \begin{cases} m_1 \cdot x_i, & x_i \in [0, a) \\ e^{(k_i(x_i - s_i))}, & x_i \in [a, b] \\ m_2 \cdot x_i, & x_i \in (b, 1] \end{cases} \quad (3)$$

where a and b represent the lower and upper bounds of x_i , respectively. The exponential function is utilized due to the discrete and exponential distribution of available lumped components. Besides, to ensure $p(x_i)$ is between the minimum and maximum component value, parameters in exponential function (i.e., k_i and s_i) are determined as

$$\begin{cases} k_i = \frac{\ln(V_{i,\max}) - \ln(V_{i,\min})}{b - a} \\ s_i = \frac{a \cdot \ln(V_{i,\max}) - b \cdot \ln(V_{i,\min})}{\ln(V_{i,\max}) - \ln(V_{i,\min})} \end{cases} \quad (4)$$

where $V_{i,\min}$ is the available minimal component value and $V_{i,\max}$ is the maximal one. When x_i is less than a or more than b , the corresponding component is shorted or opened. Thus, m_1 should be small enough and m_2 should be large enough to ensure the calculated admittance is ideal. The reason for not setting m_1 and m_2 to zero and infinity is to avoid a singular matrix when calculating scattering parameters.

Meanwhile, when x_i is in the range of $[0, a)$ or $(b, 1]$, the mapping function is still linear. Therefore, in the step of extending G_0 , if x_i generated by the optimizer is close to 0^+ or 1^- , then x_i is used to determine whether the corresponding component should be removed.

After the denormalization of \vec{X} , the scattering matrix of the corresponding circuit is solved using G_0 and denormalized component values. The first step of the solver is to calculate the primitive indefinite admittance matrix $[Y_p]$ from the incidence matrix $[A_c]$ of G_0 and the branch admittance matrix $[Y_b]$. The matrix $[A_c]$ is obtained from the topology of G_0 , which has the following form:

$$[A_c] = \begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}_{N \times M} \quad (5)$$

where N and M are the number of nodes and the number of branches of G_0 , respectively. Each column of $[A_c]$ represents a branch of G_0 . In the column representing branch b_{ij} , the element of the i th row is 1 and the element of the j th row is -1 (e.g., the first column in (5) represents the branch b_{13} with the direction from nodes 1 to 3). Note that in a passive reciprocal circuit, the predefined current direction in each column of $[A_c]$ has no effect on the result. In matrix $[Y_b]$, all elements are zero except diagonal ones. $[Y_b]$ has the following form:

$$[Y_b] = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_M \end{bmatrix}_{M \times M} \quad (6)$$

The i th row and column element of $[Y_b]$ is the admittance of the branch that the i th column of $[A_c]$ indicates, and it is calculated using (1). Then, $[Y_p]$ is calculated using following formula [3]:

$$[Y_p] = [A_c] \cdot [Y_b] \cdot [A_c]^T \quad (7)$$

Since node N is the ground, the N th row and N th column of $[Y_p]$ are directly removed to form the definite admittance matrix $[Y_d]$. Then, the admittance matrix of this circuit is obtained by eliminating $[Y_d]$ from node 3 to node $(N - 1)$. This step is performed by applying the following [36]:

$$\{[Y_d]^{(i,j)}\}^* = [Y_d]^{(i,j)} - \frac{[Y_d]^{(i,k)} \cdot [Y_d]^{(k,j)}}{[Y_d]^{(k,k)}} \quad (8)$$

Here, k denotes the node is eliminated. k begins from the last node (i.e., $k = N - 1$) to the third node (i.e., $k = 3$). In eliminating the k th node, (8) is applied to each element in $[Y_d]$ other than the k th node by traversing i and j from 1 to $(k - 1)$. Then, the k th row and column are removed, and k is decreased by 1. After k decreases to 3 (i.e., the third row and column are removed), the remaining $[Y_d]$ with the first two rows and columns is the admittance matrix $[Y]$ of this circuit. The impedance matrix $[Z]$ is obtained by

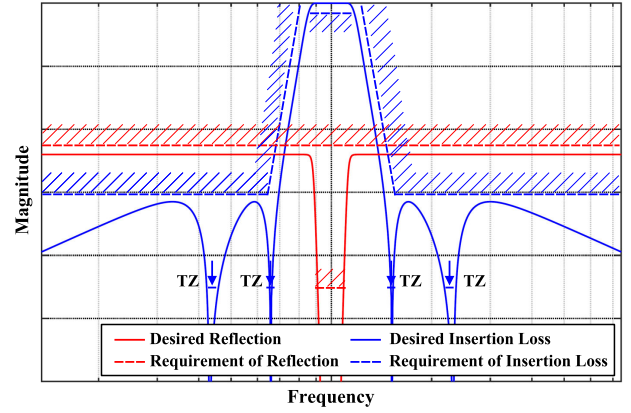


Fig. 4. Example of bandpass response and its constraints described by the target function.

inverting the admittance matrix $[Y]$. Then, scattering matrix $[S]$ is calculated by the following [1]:

$$[S] = ([Z] + Z_0[I])^{-1}([Z] - Z_0[I]) \quad (9)$$

where $[I]$ is identity matrix, and Z_0 is characteristic impedance.

The scattering matrix $[S]$ of each frequency ω is calculated from graph G_0 and vector \vec{X} , where the vector \vec{X} contains values of R , L , and C . The value of the target function f is calculated from $[S]$ of each frequency. In constructing f , both the average and maximum distances of each frequency point are utilized. f is expressed as

$$f = \sum_{(i,j) \in \{(1,1), (2,1), (2,2)\}} (f_{a,(i,j)} + f_{m,(i,j)}) \quad (10)$$

where $f_{a,(i,j)}$ denotes the average cost value calculated from scattering parameter S_{ij} of all frequencies, and $f_{m,(i,j)}$ denotes the maximum cost value. $f_{a,(i,j)}$ is described as

$$f_{a,(i,j)} = \sum_{\omega} w_{a1,ij}(\omega) r^2(|S_{ij}(\omega)| - S_{ij,UB}(\omega)) + \sum_{\omega} w_{a2,ij}(\omega) r^2(S_{ij,LB}(\omega) - |S_{ij}(\omega)|) \quad (11)$$

where $r(x)$ is the ramp function defined as $r(x) = \max(0, x)$. Meanwhile, $f_{m,(i,j)}$ is described as

$$f_{m,(i,j)} = w_{m,ij}[\max(\{r(|S_{ij}(\omega)| - S_{ij,UB}(\omega))|\forall\omega\}) + \max(\{r(S_{ij,LB}(\omega) - |S_{ij}(\omega)|)|\forall\omega\})]. \quad (12)$$

In (11) and (12), the desired response is described by setting the upper bound $S_{ij,UB}(\omega)$ and lower bound $S_{ij,LB}(\omega)$ for each frequency point, as an example of constraints on reflectionless BPF shown in Fig. 4. When TZs are required, $S_{21,UB}(\omega)$ of the corresponding frequency points is set lower than nearby frequency points. The distance between each scattering parameter and the corresponding required value is passed into the ramp function. At a single frequency point ω_0 , the ramp function produces 0 only if the scattering parameter $S_{ij}(\omega_0)$ satisfies the requirements defined by $S_{ij,LB}(\omega_0)$ and $S_{ij,UB}(\omega_0)$. In (11), all ramp function outputs are summated, and the weighted averages are calculated. Quadratic operation is applied on

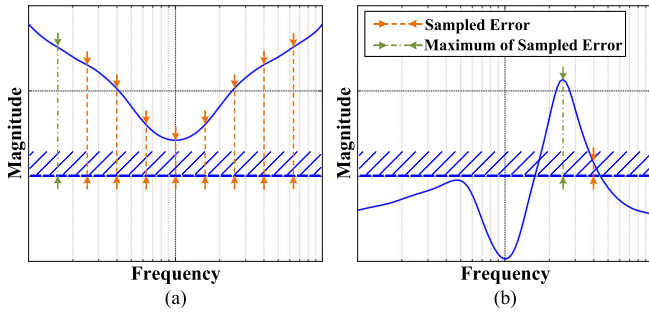


Fig. 5. Examples of two different sampling methods in different situations. (a) Average error of all sampling points dominates the early stage of optimization. (b) Maximum error of all sampling points dominates the late stage of optimization.

the ramp function to make f tend to be convex, which can afford convergence in the optimization. Parameters of w_{a1} and w_{a2} should be configured to ensure $f_{a,(i,j)}$ contributes a larger proportion of f at the beginning of optimization. Thus, $f_{a,(i,j)}$ allows the target response to take shape, as illustrated in Fig. 5(a), where the optimizer needs to suppress all violations to decrease the value of f . However, (11) only calculates average distances. A small number of large violations would be ignored which allows unwanted ripples and peaks of the response to occur, as illustrated in Fig. 5(b). Equation (12) is introduced to suppress those unwanted ripples and peaks in the response. In (12), the highest ramp function output is selected which indicates the maximal violation. The weights of w_m need to be configured to ensure that $f_{m,(i,j)}$ introduces less effect on $f_{a,(i,j)}$ when most of the requirements are unsatisfied. Meanwhile, $f_{m,(i,j)}$ can dominate f when only a few violations occur. Those parameters in (10) and (11) are set according to the specific filter performance.

Finally, circuits are represented by vectors defined in the solution space. The dimensionality of solution space is the same as the component number, and the definition domain is $[0, 1]$ in each dimension. The target function $f(\vec{X})$ maps the vector \vec{X} to a single value. The goal is to search the solution space and find a suitable \vec{X} that makes $f(\vec{X})$ small enough or be zero.

III. ALGORITHMS IN SYNTHESIS AND OPTIMIZATION

A. Details of the Proposed HGAPSO

To obtain a suitable \vec{X} for the target function described in Section II-B, a specifically enhanced HGAPSO is proposed. This algorithm is based on the PSO algorithm. The strategy of PSO is that each particle moves toward its own best location and the global best location. This strategy gives the heuristic ability to PSO. To overcome the drawbacks of PSO in high-dimensional solution space, the mutation and selection parts of GA are introduced to construct the proposed HGAPSO. The mutation part is realized through random perturbation and classifier-assisted particle reinitialization. The selection part reduces unnecessary calculations by removing worse particles among congregated particles. The selection mechanism of the proposed HGAPSO is achieved through the lifespan and fitness of particles. Each particle has a parameter

of lifespan with an initial value. In the iteration, the fitness of each particle is determined from the corresponding target function value. Then, lifespan is decreased according to fitness. Once the lifespan of a particle becomes negative, the particle is removed. Most of the removed particles are reinitialized in the optimal zone. The optimal zone is determined by the best particles using a classifier. Therefore, reinitialized particles can be regarded as variants of best particles, which form the mutation part and boosts the convergence of the proposed HGAPSO. Any type of classifier can be employed while a simple hyperrectangle classifier is used in this implementation. To avoid the optimal zone being very narrow and reduce the probability of converging to a local optimum, two policies are introduced. The first policy expands the upper and lower bounds of the hyperrectangle. The optimal zone is extended to prevent only the local optimum is covered. Another policy is allowing a portion of particles to be reinitialized outside the optimal zone. Thus, those particles explore solutions far from the particle swarm. In the experiment, such a strategy shows good performance in optimization while not taking up too much calculation.

The flowchart of the implemented procedure is shown in Fig. 6. Note that all variables and solution space are normalized and limited to $[0, 1]$, while the target function is $f(\vec{X})$. The initialization part sets each element in vectors \vec{C}_{\max} to maximum and each element in \vec{C}_{\min} to minimum. Those two vectors describe the optimal zone of the classifier. Thus, the classifier initially covers the whole solution space of D dimensions. Here, D is the same as the component number. After that, the location of all particles, i.e., \vec{X}_i , are initialized randomly in the solution space. Here, a function denoted as $\text{rand}(\vec{V})$ is employed to generate a vector that has the same dimensionality as its input vector \vec{V} . The generated vector is located randomly inside the hyperrectangle defined by \vec{V} and coordinate origin. Then, the local best location $\vec{X}_{\text{LB},i}$ of each particle is set to its initial location \vec{X}_i . The lifespan L_i of each particle is set to initial lifespan L_{init} . The particle initialization process is executed in a loop which is carried for P times. Here, P is the number of particles. When all particles are initialized, the target function value of each particle is calculated. The location of the particle with minimum function value is selected as the global best point \vec{X}_{GB} . Finally, the iteration counter is reset. This counter is employed to stop iteration once the number of total iterations reaches its limit of T .

After initialization, the iteration starts. In the normal iteration section, the subprocess shown in Fig. 7 controls the movement of each particle \vec{X}_i . During the subprocess, the results using four movement strategies are calculated based on each particle's current location \vec{X}_i . The first two movement policies, denoted as $\vec{X}_i^{(1)}$ and $\vec{X}_i^{(2)}$, are based on the original PSO. Here, the particle moves toward the global best location \vec{X}_{GB} with the weight of w_g to location $\vec{X}_i^{(1)}$. Then, it further moves toward local best location $\vec{X}_{\text{LB},i}$ with the weight of w_l from $\vec{X}_i^{(1)}$ to location $\vec{X}_i^{(2)}$. The third strategy $\vec{X}_i^{(3)}$ introduces perturbation on current position \vec{X}_i . In each dimension, the coordinate value of the particle is added with a random value. The scale of this random value is limited by the weight w_r ,

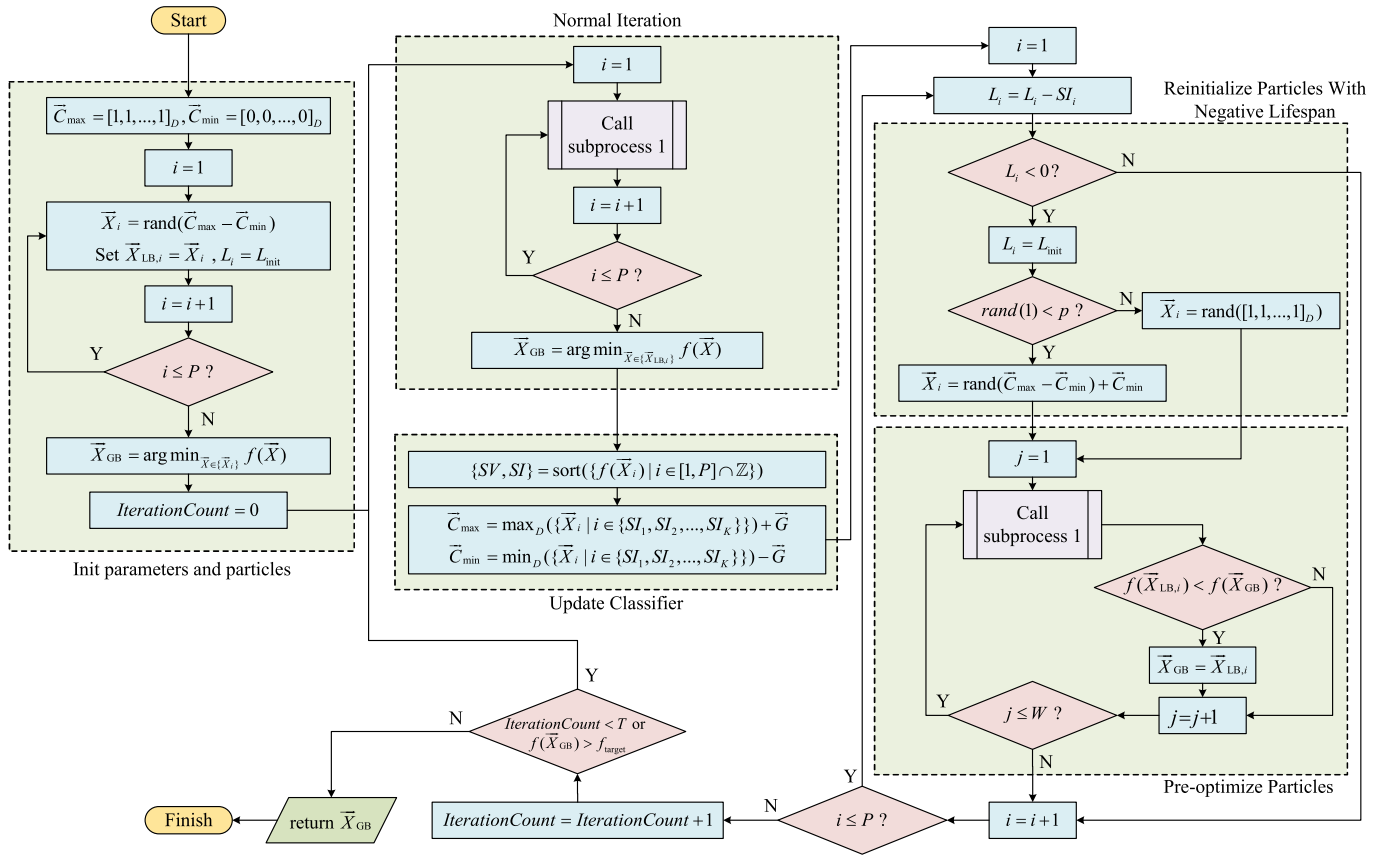


Fig. 6. Flowchart of the proposed HGAPSO algorithm.

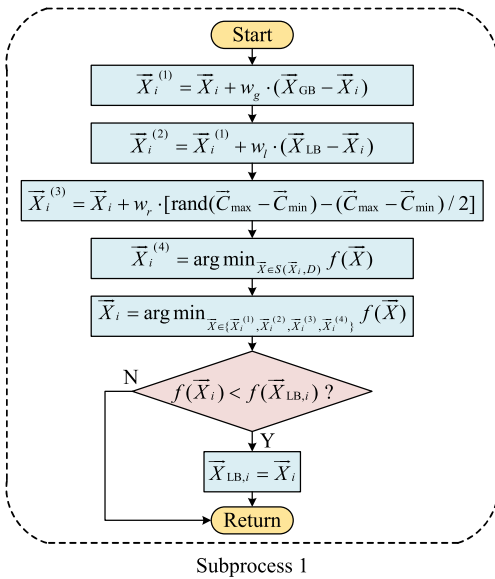


Fig. 7. Flowchart of the subprocess 1 in Fig. 6.

multiplied by the length of the classifier's coverage of the corresponding dimension. Note that the gap in each dimension between \bar{C}_{\max} and \bar{C}_{\min} shrinks along with iteration, which makes the perturbation of $\bar{X}_i^{(3)}$ gradually become smaller. Thus, the resolution of random neighborhood search actualized by $\bar{X}_i^{(3)}$ increases with iteration. To calculate $\bar{X}_i^{(4)}$, a function $S(\bar{X}, D)$ is defined as follows: $S(\bar{X}, D)$ randomly selects a

dimension from all D dimensions. Then, $S(\bar{X}, D)$ takes values in this dimension to generate a set of vectors, where values from other dimensions of \bar{X} are held. This step is noted as random coordinate search (RCS), which is a simplification of random direction search (RDS) [37], [38]. The principle of RDS is that it randomly selects a direction vector, then take samples along the line defined by this vector. To balance the speed and search capability, in this embedded RCS, the direction vector is always parallel to a coordinate axis. In a single iteration, the best sample point from $S(\bar{X}, D)$ is set as $\bar{X}_i^{(4)}$. Note that the selected coordinate axis should be different for the next iteration. As shown in Fig. 8, it can help particles cross long-distance gaps between local optimums. After new positions of $\bar{X}_i^{(1)} - \bar{X}_i^{(4)}$ are calculated, the subprocess chooses the best one to replace the previous position of the particle \bar{X}_i . If this new position is better than $\bar{X}_{LB,i}$, $\bar{X}_{LB,i}$ is updated using \bar{X}_i . When all particles are updated, the best $\bar{X}_{LB,i}$ is selected instead of the global best position \bar{X}_{GB} .

The subprocess is called independently for each particle which makes it suitable for parallel executing. Besides, to balance the speed of iteration and heuristic ability, a method inspired by simulated annealing (SA) is introduced in the subprocess of calculating $\bar{X}_i^{(3)}$ and $\bar{X}_i^{(4)}$ to accelerate iteration. In SA, the probability is utilized to determine whether to accept a new solution that is worse than the current solution. This probability decreases as the algorithm converges. In the proposed strategy, the probability depends on $f(\bar{X}_{GB})$ and

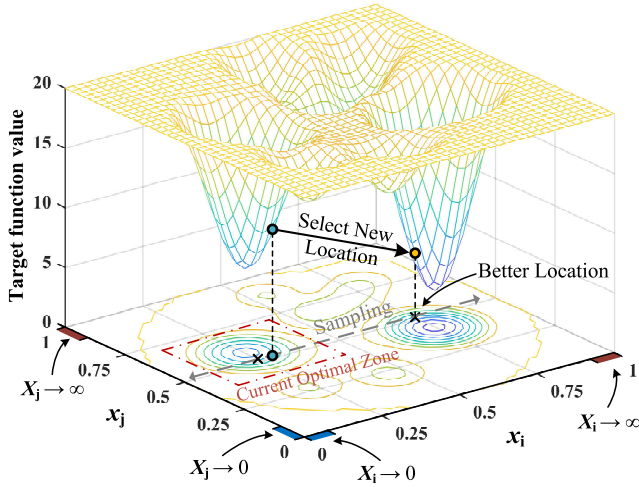


Fig. 8. Example for illustrating the effectiveness of the embedded RCS. (The range of $X_i \rightarrow 0$ and $X_j \rightarrow 0$ denotes a shorted resistor, a shorted inductor, or an opened capacitor, while the range of $X_i \rightarrow \infty$ and $X_j \rightarrow \infty$ denotes an opened resistor, an opened inductor, or a shorted capacitor.)

the maximum value of the target cost function f_{\max} . For the embedded RCS of calculating $\bar{X}_i^{(4)}$, whether a single component is in the short zone or the open zone can be confirmed easily. Thus, it takes effect at the beginning when the topology is undetermined. In late iteration, better positions are hard to be found by RCS since the topology is stable. Meanwhile, since variables change significantly in the early iteration due to $\bar{X}_i^{(4)}$, the random disturbance of $\bar{X}_i^{(3)}$ could not lead to contributions. If particles are trapped in a local optimum, random disturbance could afford particles to jump out to better locations nearby. Besides, because of the shrunk optimal zone, $\bar{X}_i^{(3)}$ searches the solution space finely when $f(\bar{X}_{GB})$ is small. Therefore, the probability distribution for calculating $\bar{X}_i^{(4)}$ has a higher value when $f(\bar{X}_{GB})/f_{\max}$ is near 1 other than 0. The probability for calculating $\bar{X}_i^{(3)}$ has the opposite distribution to $\bar{X}_i^{(4)}$. To satisfy characteristics of such requirements, probability distribution functions using tanh are constructed as

$$\begin{cases} P_{\bar{X}_i^{(3)}} = \frac{1}{2} \tanh\left(k_1 \left(\frac{1}{2} - \frac{f(\bar{X}_{GB})}{f_{\max}}\right)\right) + \frac{1}{2} \\ P_{\bar{X}_i^{(4)}} = \frac{1}{2} \tanh\left(k_2 \left(\frac{f(\bar{X}_{GB})}{f_{\max}} - \frac{1}{2}\right)\right) + \frac{1}{2} \end{cases} \quad (13)$$

where parameters k_1 and k_2 are used to adjust the slope in the middle. The distributions of those functions are shown in Fig. 9. By introducing this probability-based strategy for deciding whether to calculate $\bar{X}_i^{(3)}$ and $\bar{X}_i^{(4)}$, the amount of calculation for each iteration is nearly reduced by half. Meanwhile, the converging performance is almost unchanged.

After the normal iteration, the classifier needs to be updated, which covers optimal particles by defining the optimal zone. The optimal zone is used in subprocess 1 and the particle-reinitialization, as an example shown in Fig. 10. Here, a simple classifier update strategy is introduced, which can be based on any sorting algorithm. The sorting function sorts target function values corresponding to each particle in order from lowest to highest. After all particles are sorted, the sorting

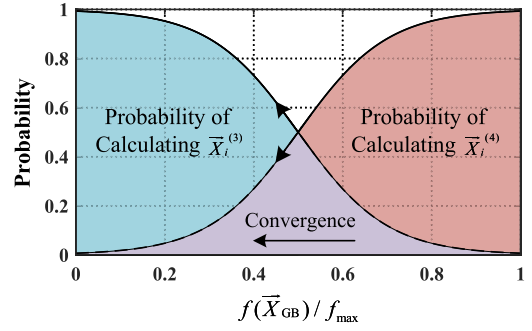


Fig. 9. Distributions of the constructed probability functions (k_1 and k_2 are set to 5).

function returns the function value list SV and index list SI of particles. Though sort is used in the classifier updating part, it can be carried out before updating \bar{X}_{GB} in the normal iteration part. Then, \bar{X}_{GB} is directly set as \bar{X}_{SI_1} if SV_1 is smaller than $f(\bar{X}_{GB})$. After the sorting process, the best K particles are selected. From such K particles, the function \max_D returns the maximum value of each dimension to form the upper bound \bar{C}_{\max} , while the function \min_D returns the minimum value of each dimension to define the lower bound \bar{C}_{\min} . To avoid particles being aggregated to a local minimum, a vector \vec{G} is employed to extend the gap between \bar{C}_{\max} and \bar{C}_{\min} in each dimension. Here, the vector \vec{G} is generated by setting each element to $(G/2)$, where the parameter G denotes the extending length in each dimension. After \bar{C}_{\max} and \bar{C}_{\min} are calculated, \bar{C}_{\max} is increased by \vec{G} , and \bar{C}_{\min} is decreased by \vec{G} . Note that each element of \bar{C}_{\max} and \bar{C}_{\min} is defined between 0 and 1. Thus, the value out of range is fixed to the maximum of 1 or the minimum of 0.

When \bar{C}_{\max} and \bar{C}_{\min} are updated, all particles are already sorted according to their value. The index list SI generated from the sorting function is utilized as fitness to calculate the lifespan of each particle. Here, a simple strategy is imported that each particle's lifespan L_i is decreased by their sort index SI_i (i.e., the lower the target function value of the particle, the less the lifespan is reduced.). Once L_i becomes negative, the i th particle is reset. Such a procedure consists of two steps. The first step is resetting parameters, while the second step is executing the preoptimization. In the first step, the negative L_i is reset to the initial value L_{init} . Then, a random value between 0 and 1 is used to determine whether to reset \bar{X}_i in the optimal zone. If the random value is larger than the probability p , and \bar{X}_i is reset in the whole solution space. Otherwise, it is reset in the optimal zone defined by \bar{C}_{\max} and \bar{C}_{\min} . To help the reinitialized particles catch up with existing particles in iteration, preoptimization is applied to those particles in the second step. The preoptimization uses the same procedure by calling subprocess 1 with repeated W times, where W increases with I_C . Here, I_C is the abbreviation of IterationCount in Fig. 6. In this implementation, W is calculated as

$$W(I_C) = \lg(I_C + 1). \quad (14)$$

Since W is constructed using a logarithmic function, the preoptimization procedure does not slow down the iteration

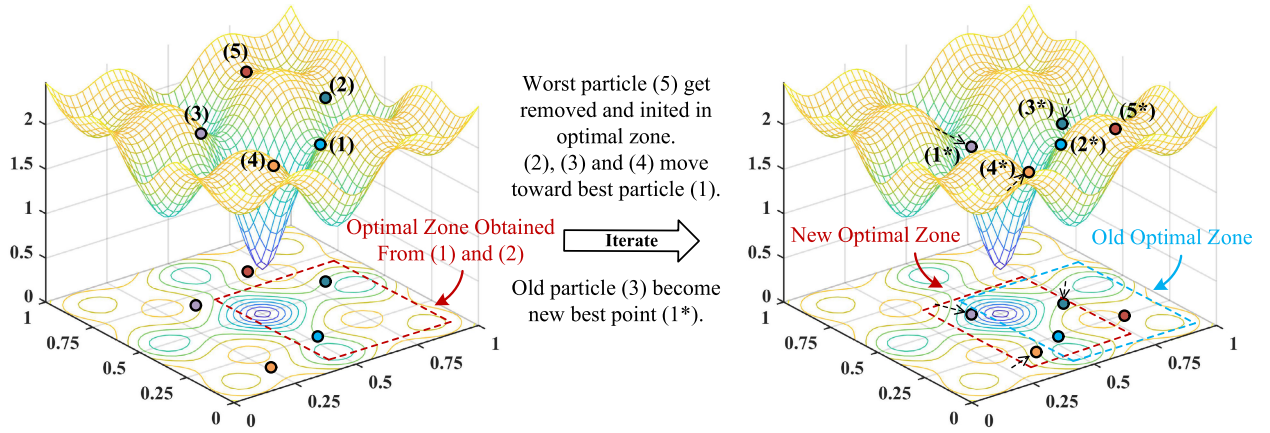


Fig. 10. Example showing one step iteration of five particles in Ackley function of two dimensions. In this example, the classifier is used to cover the best two particles.

TABLE I
LIST OF HYPER-PARAMETERS USED IN THE TEST

Parameter	Value
P	1000
L_{init}	$120 \cdot P$
w_g	0.024
w_l	0.016
w_r	0.6
G	0.3
K	$0.02 \cdot P$
p	0.7

significantly. Besides, during the preoptimization, if $f(\vec{X}_i)$ is smaller than $f(\vec{X}_{GB})$, \vec{X}_{GB} should be updated using \vec{X}_i .

Finally, when the procedure described above is finished, an iteration is completed, and I_C is increased by 1. Unless I_C reaches its limitation T or $f(\vec{X}_{GB})$ satisfies the requirement of f_{target} , the iteration loop is continued. Once the iteration loop is stopped, the global best location \vec{X}_{GB} is the final result.

B. Comparison of Heuristic Algorithms

To demonstrate the advantages of the proposed HGAPSO, the function of (10) with the same parameters for synthesizing reflectionless filters is applied to different algorithms for comparison. The results of convergence are shown in Fig. 11. Each algorithm is executed 50 times while the convergence curve of the best result is selected. The proposed HGAPSO is denoted as “HGAPSOCRS” in the legend of Fig. 11. The “C” suffix stands for “classification.” The “R” suffix stands for “random neighborhood search” which is the step of calculating $\vec{X}^{(3)}$. The “S” suffix stands for RCS which is the step of calculating $\vec{X}^{(4)}$. In these tests, the same hyperparameters are used for all algorithms, except for the gradient descent (GD) and RDS, which have no concept of particles. The hyperparameters used are listed in Table I. The details of selecting hyper-parameters are presented in the Appendix. Besides, since GD tends to fall into local optimums, a preliminary procedure of randomly generating initial positions is carried out. Then, the best position of 1000 initial positions is selected to start GD and RDS.

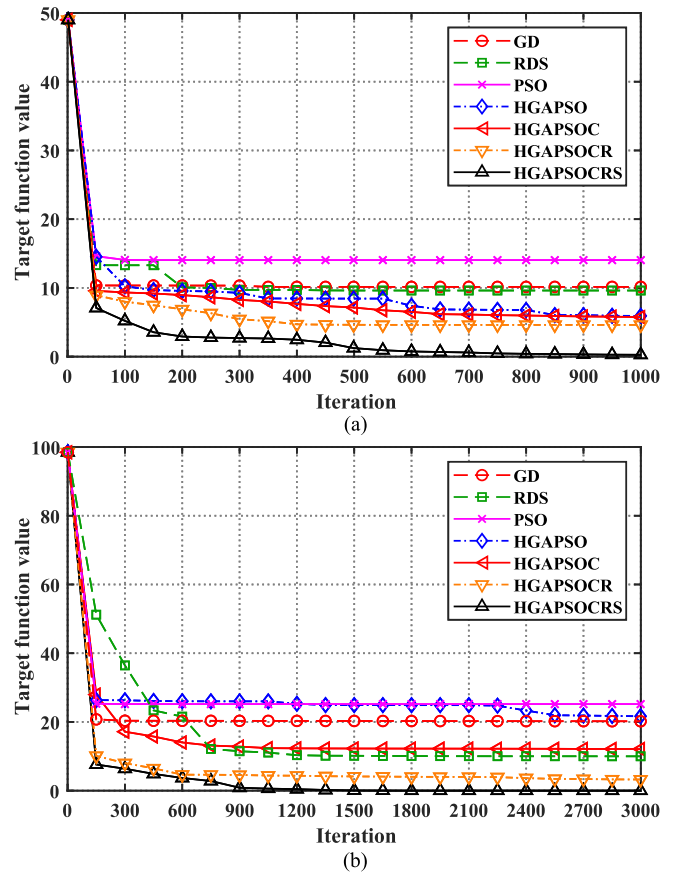


Fig. 11. Convergence comparison of different algorithms using target functions in synthesizing reflectionless filter. (a) Target of a small-scale reflectionless BPF. (b) Target of a reflectionless wideband BPF.

As illustrated in Fig. 11, particles of PSO can be trapped into local optimums in high-dimensional optimization. Compared to PSO, HGAPSO has the ability to prevent particle aggregation, but particles still lack searching ability in such a high-dimensional solution space. When classification is introduced, convergence is boosted. However, classification cannot improve searching ability. Meanwhile, the embedded optimizations of random neighborhood search and RCS ameliorate the heuristic ability significantly. In Fig. 11(b), the target function

of synthesizing wideband reflectionless BPF is used, which has more sample points and tighter passband constraints. In both comparisons, the proposed HGAPSO shows the ability of fast convergence and better final results.

C. Design Layout and Fine-Tune

The topology of the synthesized filter is determined by referring to \bar{X}_{GB} generated from the proposed HGAPSO and its corresponding graph G_0 . Then, the layout is designed along with substrate characteristics, components' package, and circuit terminations. In this step, all normalized variables are denormalized to actual component values. When the circuit is synthesized directly (i.e., the cost function describes a bandpass response instead of a low-pass response), the LPF prototype converting formulas from [1] is utilized which are listed below

$$\begin{cases} L = \frac{Z_0 g_{L_i}}{\omega_c} \\ C = \frac{g_{C_i}}{Z_0 \omega_c} \\ R = Z_0 g_{R_i} \end{cases} \quad (15)$$

where g_{L_i} , g_{C_i} , and g_{R_i} are prototype values of the inductor, capacitor, and resistor, respectively. Z_0 is the characteristic impedance of the transmission line connected to the terminals, while ω_c is the desired center frequency.

If the target is a BPF but the response is described in a low-pass response, the LPF is synthesized. The low-pass-to-bandpass transformation is performed to convert the synthesized LPF into BPF. Here, each inductor is connected in series with a new capacitor, while each capacitor is paralleled with a new inductor. The prototype value of the added component in each pair is the reciprocal value of the original component value [1]. After all components in the prototype are converted, (15) is applied to calculate the actual values. Besides, if the target is a BSF or high-pass filter, similar procedures described in [1] can be applied for converting.

During the circuit implementation, the influence of the layout is considered. The scattering matrix of the layout without lumped components is obtained through EM simulation. Then, the indefinite impedance matrix is calculated by using (9) in reverse as follows:

$$[Z] = Z_0([I] + [S])([I] - [S])^{-1}. \quad (16)$$

The admittance matrix of the layout (i.e., $[Y_l]$) is the inverse matrix of the impedance matrix $[Z]$. Then, the admittance matrix of each component is added to the corresponding rows and columns of $[Y_l]$ to form the indefinite admittance matrix $[Y_p]$. After $[Y_p]$ is formed, the method described in (8) and (9) is utilized to calculate scattering matrix $[S]$ of the circuit. The assessment procedure in (10) is applied to fine-tune components' values considering the effect of the layout. Besides, the frequency in the target function is changed to the denormalized frequency.

The parasitic parameters of the packaged lumped components are considered once self-resonator frequencies are near the passband frequency. The models of parasitic parameters

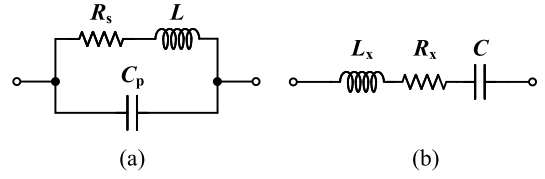


Fig. 12. Model of parasitic parameters of SMD components. (a) Inductor. (b) Capacitor.

employed in this work are shown in Fig. 12. For inductors, C_p is the parasitic capacitance of the package, which is fixed for a specific inductor series. R_s is the metal resistor that is approximately linear to inductance L . It is expressed as a function $R_s(L)$. Thus, the admittance of an inductor with parasitic parameters is calculated as

$$Y_L = \frac{1}{j\omega L + R_s(L)} + j\omega C_p. \quad (17)$$

For capacitors, L_x and R_x are the parasitic inductance and resistance of the metal, respectively, which are approximately linear to $1/C$. Therefore, functions $L_x(C)$ and $R_x(C)$ are introduced to model parasitics of capacitors. The admittance of a capacitor with parasitic parameters is calculated as

$$Y_C = \frac{j\omega C}{j\omega C R_x(C) - \omega^2 C L_x(C) + 1}. \quad (18)$$

Parasitic parameters are obtained by referring to the self-resonator frequency and Q from datasheets. To assist the optimization, linear spline interpolation is used, which is expressed as follows:

$$\hat{S}_i(x) = \frac{s_{i+1} - s_i}{x_{i+1} - x_i} \cdot x + \frac{x_{i+1}s_i - x_i s_{i+1}}{x_{i+1} - x_i} \quad (19)$$

where x_i and x_{i+1} are two adjacent component values, s_i and s_{i+1} are the corresponding parasitic parameters, and $\hat{S}_i(x)$ is the predicted parasitic value, respectively. The domain of interpolated function $\hat{S}_i(x)$ is $[x_i, x_{i+1}]$. During the fine-tuning, discrete parasitic parameters are modeled as piecewise continuous functions using (19).

After considering the parasitic parameters of both layout and lumped components, reflectionless characteristics deteriorate. Since the components' values generated from the HGAPSO are near the optimal solution, it is effective to use a gradient-based method for further optimization. The flowchart is shown in Fig. 13. Parameter D_0 limits the minimum length of the gradient vector. Once the length is small enough, it means that the shape of the function f is flat enough near \vec{X} . Thus, \vec{X} is returned as the final value. Parameters step_0 and step_{\min} control the sweep step in the direction of the gradient. During the sweeping, if $f(\vec{X}^*)$ is smaller than $f(\vec{X}_{\text{cur}})$, \vec{X}_{cur} is updated using \vec{X}^* , then \vec{X} will be replaced by \vec{X}_{cur} after the sweeping loop. When step is too large, i.e., the first attempt strides across all regions where the value of f is lower than $f(\vec{X}_{\text{cur}})$, the branch of $t = t + \text{step}$ is not triggered. Therefore, t is equal to step which triggers the branch of reducing parameter step by half. Once step reaches the minimum limitation of step_{\min} (i.e., current location \vec{X} is close enough to the optimal solution), the iteration is stopped. Besides, the variable count that functions as an iteration

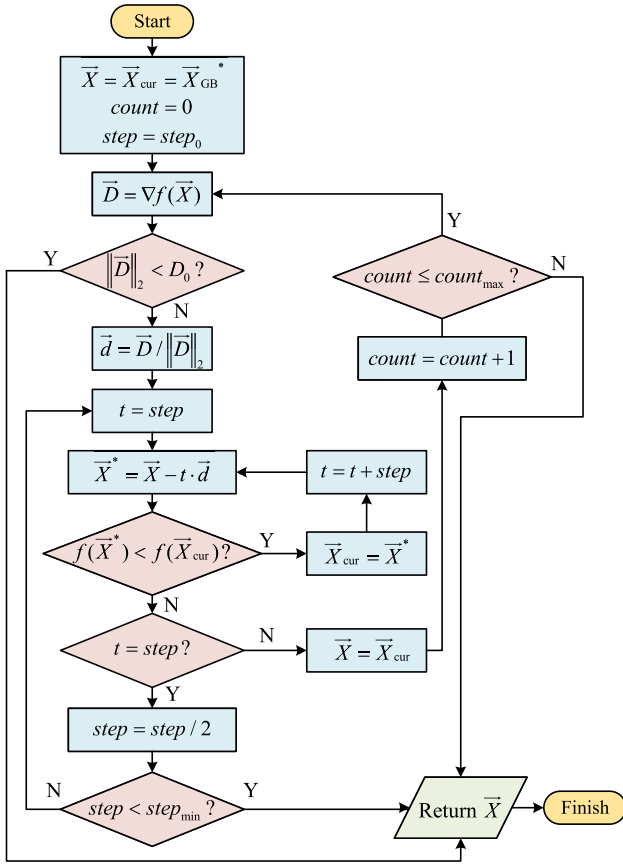


Fig. 13. Flowchart of the gradient-based optimization for fine-tuning.

counter are employed. If count reaches its limit $count_{max}$, the iteration is terminated, and \bar{X} is the final result.

IV. DESIGN EXAMPLE

In this section, four synthesized reflectionless filters are presented to demonstrate the effectiveness of the proposed procedure. The grid-type quasi-planar graphs, which have an example of 13 nodes illustrated in Fig. 3, are used as the final G_0 for all results. The layouts are designed according to the topologies synthesized by the proposed HGAPSO. Then, the component values of each filter are denormalized and fine-tuned with the corresponding layout using gradient-based optimization. Note that parasitics of both layouts and components are considered in the fine-tuning. The closest practical value of each component is selected to fabricate the filter, where the inductors are from the muRata LQW15AN series and the capacitors are from the muRata GJM1555 series.

A. Reflectionless Filter With Deep Stopband Attenuation

The first synthesized result is a reflectionless BPF with deep stopband attenuation. This design example is also used to validate that topologies of fewer components can be found by the proposed procedure, while passband characteristics are maintained. By inputting a synthesized topology as G_0 to the synthesis procedure, a simpler topology is obtained if G_0 can be simplified. This procedure is repeated multiple times until G_0 can no longer be further simplified. To demonstrate this

functionality, the three-stage reflectionless filter from [4] is used as the reference of response. The passband and TZs are constrained as the reference, while the stopband rejection is set to 55 dB, which is 10 dB deeper than the reference. Besides, the topology is also constrained symmetrically due to the reference also having a symmetrical topology. The reflection coefficient is limited to -30 dB. All hyper-parameters used in synthesizing this filter are listed in Table I. The schematic of the final synthesized filter is shown in Fig. 14. This filter only needs 15 inductors, 15 capacitors, and 4 resistors. Compared to the referenced filter, three inductors, three capacitors, and two resistors are saved. The calculated responses from schematics are presented in Fig. 15. The synthesized filter has a passband that is nearly identical to the reference. The TZs are added at the same frequency, while the insertion peak in the stopband is 58.3 dB.

In fabricating this filter, the center frequency is set to 433 MHz. The designed layout is shown in Fig. 16. The PCB is implemented using the dielectric substrate of RO4350B (i.e., typical ϵ_r of 3.66 and a thickness of 0.762 mm). Values of practical components are presented in Fig. 14. Photographs of the fabricated filter are shown in Fig. 17. The measured, simulated, and calculated responses are shown in Fig. 18. In Fig. 18(a), the normalized frequencies are predicted using the following:

$$\Omega = \frac{1}{\text{FBW}} \left(\frac{f}{f_0} - \frac{f_0}{f} \right) \quad (20)$$

where Ω is the normalized frequency, FBW is the fractional bandwidth (FBW) and is set to 1, f is the original frequency, and f_0 is the center frequency for normalization, respectively [2]. Note that in calculating the result with parasitics, both layout and components are considered. The post-layout simulation is performed using the component models provided by muRata. As shown in Fig. 18, the measured insertion loss at the center frequency is 1.70 dB. The 0.5-dB passband is 383–517 MHz (FBW of 29.8%). For the upper stopband starting from 674 MHz, the rejection is better than 52 dB. The measured peak reflection coefficient is -17.6 dB near the passband, and this performance is sustained up to 1.65 GHz. The measured return loss is larger than 10 dB in the range from dc to 3.04 GHz, which is roughly seven times the center frequency. The comparison between the measured responses of this filter and the referenced filter is presented in Table II.

B. Wideband Reflectionless BPF

The second synthesized result is a reflectionless BPF with a flat and wider passband. Meanwhile, multiple TZs are also required in stopbands. The constraints similar to Fig. 4 are used, while 0.5-dB FBW is set to 30%, the rejection in stopbands is set to 30 dB, and the minimum return loss of both ports is assigned to 20 dB. Besides, values of inductors and capacitors are limited to avoid the use of tiny capacitance and huge inductance in synthesis. In this case, the upper bound of inductors is set to 47 nH, and the lower bound of capacitors is set to 2 pF. After 3000 iterations, the topology is obtained. The synthesized schematic and corresponding layout are shown in Figs. 19 and 20, respectively. The PCB is implemented using

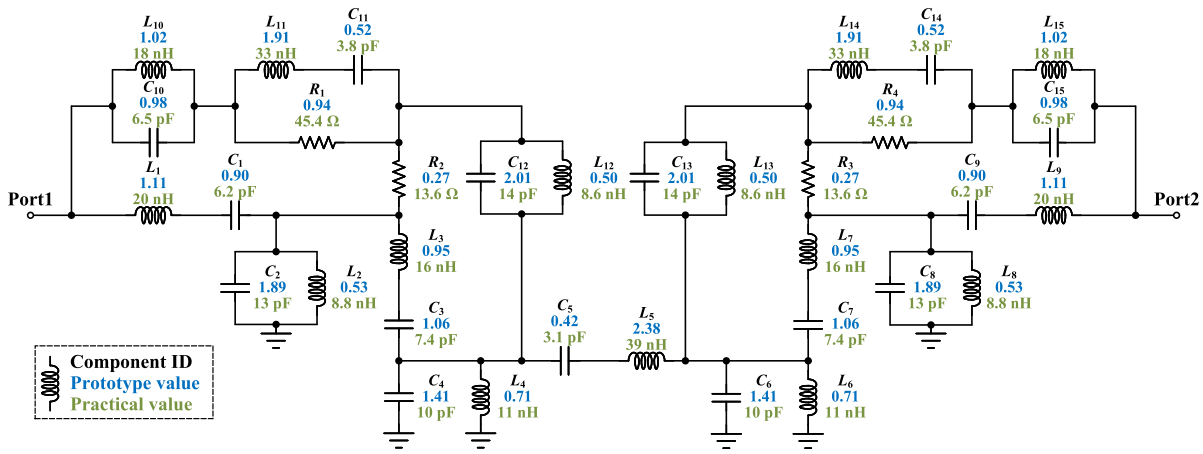


Fig. 14. Schematic of the synthesized reflectionless BPF with symmetrical topology.

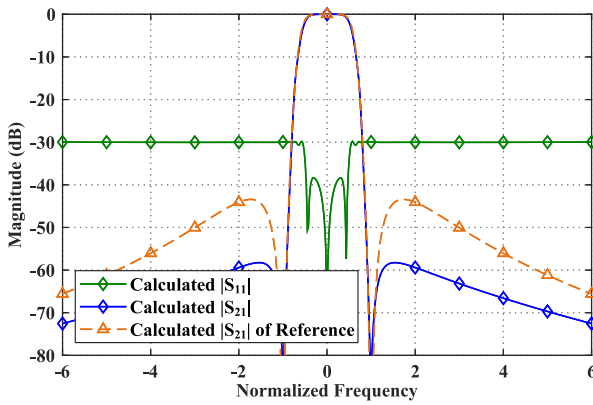


Fig. 15. Calculated response of the synthesized filter and the three-stage BPF from [4].

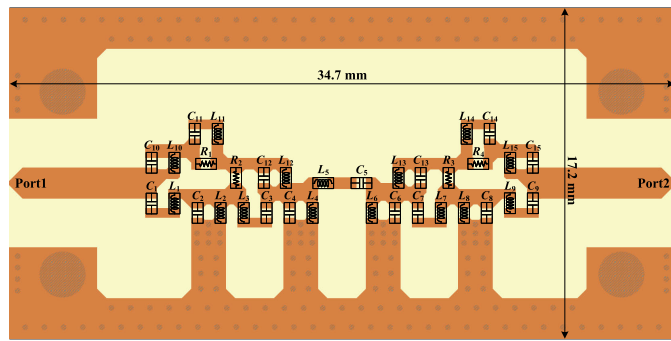
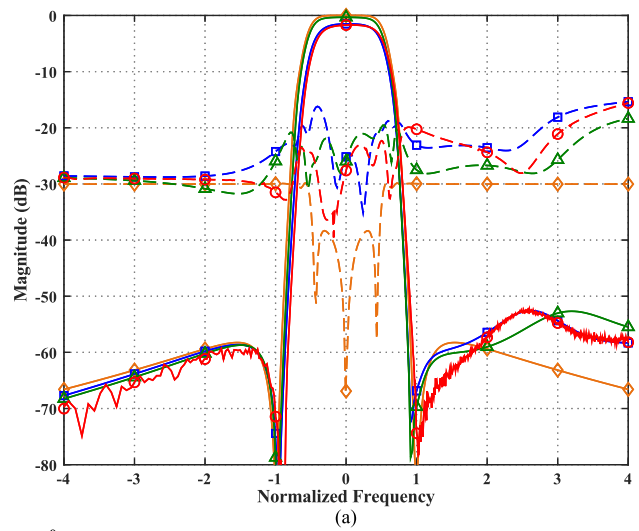


Fig. 16. Layout of the synthesized reflectionless BPF.

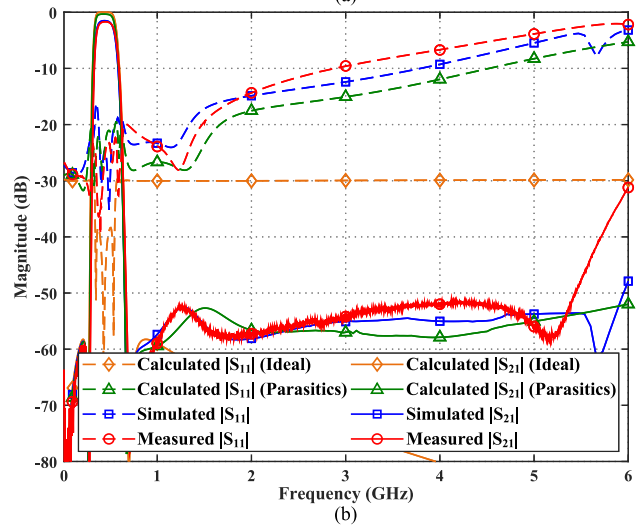


Fig. 17. Photographs of the fabricated reflectionless BPF. (a) Top view. (b) Bottom view.

Fig. 18. Calculated, simulated, and measured results of the synthesized reflectionless BPF with symmetrical topology. (a) Response with frequency normalization using the center frequency of 433 MHz. (b) Response without frequency normalization.

the dielectric substrate of RT5880 (i.e., typical ϵ_r of 2.2 and a thickness of 0.508 mm). Pads of SMD 0402 (i.e., component length of 1 mm and width of 0.5 mm) are used for soldering lumped components. The practical values are calculated from

the prototype by setting the center frequency of 433 MHz, and they are fine-tuned with the layout using gradient-based optimization.

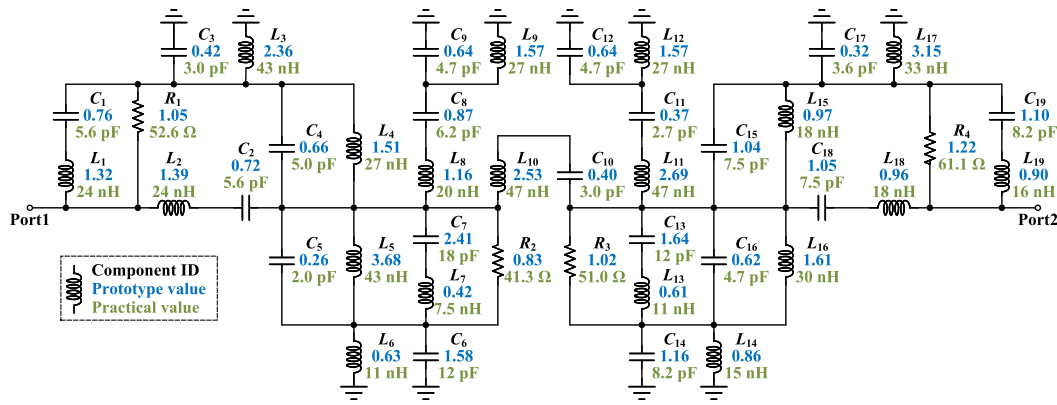


Fig. 19. Schematic of the synthesized reflectionless wideband BPF with flat-passband. (The middle line of each label, i.e., the blue line, is the corresponding prototype value. The bottom line of each label, i.e., the green line, is the value of the component used in fabricating the filter.)

TABLE II

COMPARISON BETWEEN MEASURED RESPONSES OF LUMPED BPFs

Ref.	[4]	This work
Passband Response	Flat	Flat
Center Frequency f_0	192 MHz [▲]	435 MHz
Insertion Loss (@ f_0)	3.4 dB [▲]	1.7 dB
FBW _{0.5dB} *	20% [▲]	29.7%
FBW _{3dB} *	43% [▲]	47.4%
Stopband Rejection	>43 dB (>1.5 f_0) [▲]	>52 dB (1.55-12.3f_0)
Maximum Frequency of $S_{11} < -15$ dB	3.9 f_0 [▲]	4.5 f_0
Components**	18L, 18C, 6R	15L, 15C, 4R

*: FBW is calculated by $FBW = (f_H - f_L) / [(f_H + f_L) / 2]$.

** : L: Inductor, C: Capacitor, R: Resistor.

▲: Estimated from the figure.



Fig. 21. Photographs of the fabricated reflectionless wideband BPF. (a) Top view. (b) Bottom view.

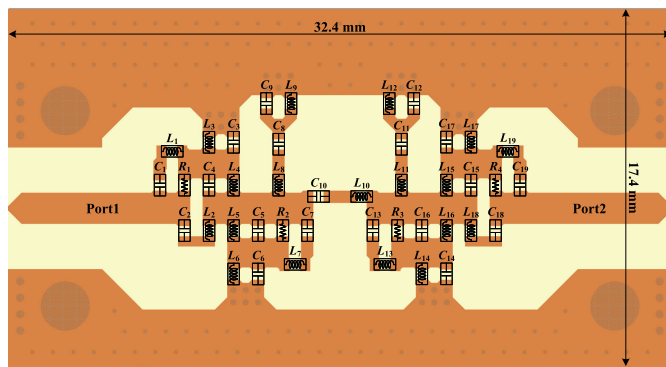


Fig. 20. Layout of the reflectionless wideband BPF.

This filter is fabricated and its photographs are presented in Fig. 21. The calculated, simulated, and measured results are shown in Fig. 22. Fig. 22 exhibits that the measured insertion loss at center frequency is 1.24 dB. The 0.5-dB passband is 372–499 MHz (FBW of 29.2%), while the 3-dB passband is 337–540 MHz (FBW of 46.3%). The stopband is from 590 MHz to 5.65 GHz, while the rejection is better

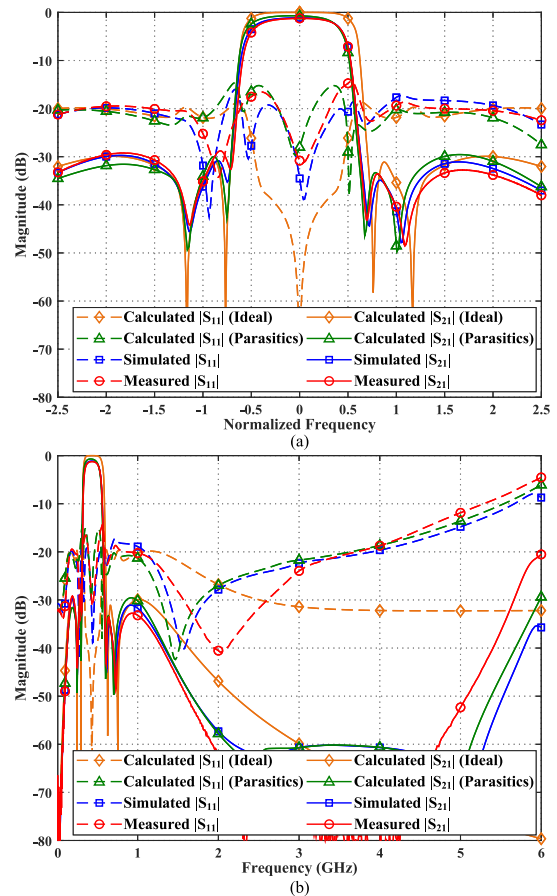


Fig. 22. Calculated, simulated, and measured results of the synthesized reflectionless wideband BPF. (a) Response with frequency normalization using the center frequency of 433 MHz. (b) Response without frequency normalization.

than 30 dB. The measured reflection coefficient is lower than -15 dB from dc to 4.55 GHz, which is more than ten times the center frequency. The calculated, simulated, and measured results show good agreement.

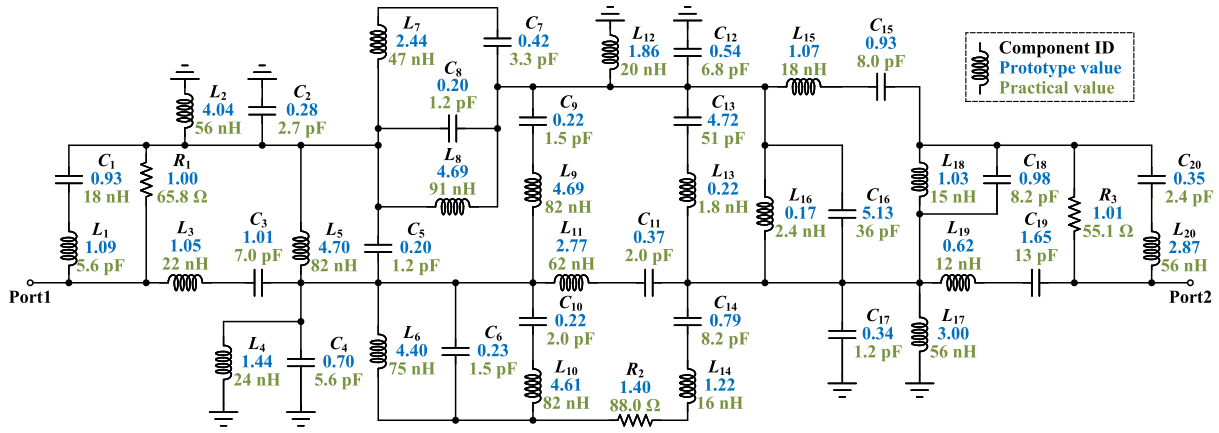


Fig. 23. Schematic of the synthesized reflectionless wideband BPF with equiripple passband.

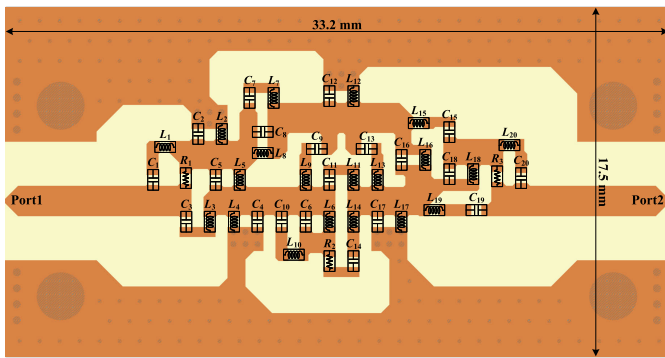


Fig. 24. Layout of the reflectionless wideband BPF with equiripple passband.

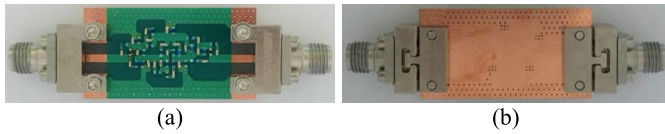


Fig. 25. Photographs of the fabricated reflectionless wideband BPF with equiripple passband. (a) Top view. (b) Bottom view.

C. Reflectionless Wideband Filter With Equiripple Passband

To further verify that filters with wider passband can be synthesized, another result is a reflectionless BPF with a wider and equiripple passband. The same hyper-parameters and initial graphs used in synthesizing the previous filter are utilized. Similar boundary constraints in Fig. 4 are applied with an additional constraint of the 0.5-dB ripple in the passband. The 3-dB FBW is set to 100%. The upper bound of inductors is set to 91 nH and the lower bound of capacitors is set to 1 pF. After 3000 iterations, a suitable topology is obtained. The schematic and corresponding layout are shown in Figs. 23 and 24, respectively. The PCB is implemented using the dielectric substrate of RT5880 with a thickness of 0.508 mm. The component values of the prototype and practical filter are presented in Fig. 23.

This filter is fabricated and its photographs are shown in Fig. 25. As shown in Fig. 26, the measured insertion loss at center frequency is 0.89 dB. The 3-dB passband is 247–709 MHz (FBW of 96.7%). For the upper stop-band starting from 759 MHz, the rejection is better than

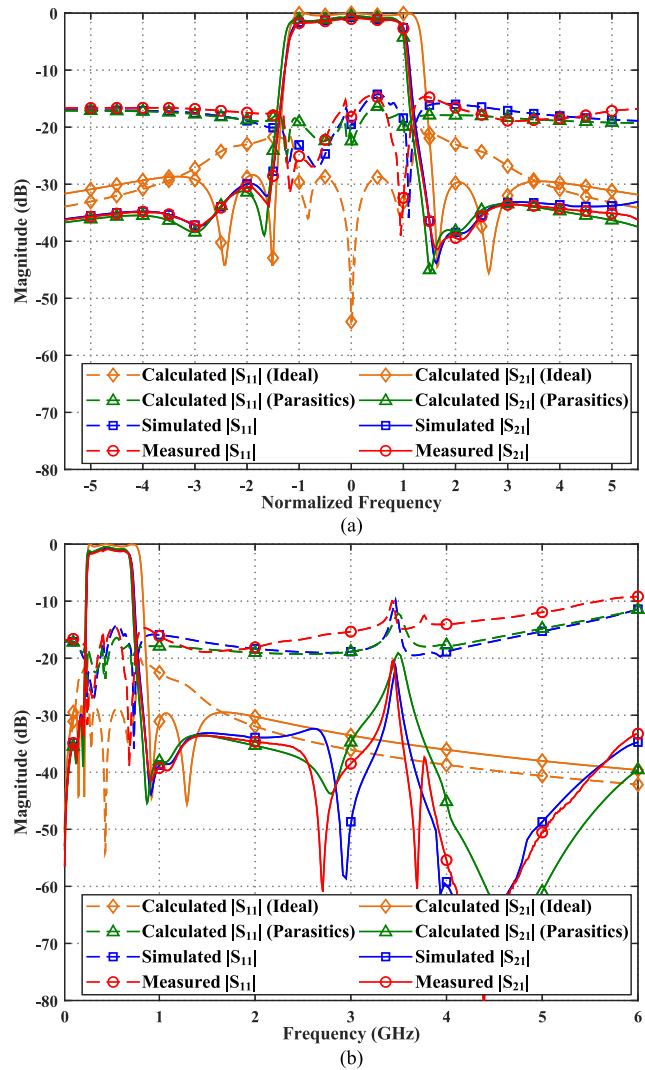


Fig. 26. Calculated, simulated, and measured results of the synthesized reflectionless wideband BPF with equiripple passband. (a) Response with frequency normalization using the center frequency of 433 MHz. (b) Response without frequency normalization.

20 dB. Meanwhile, in the frequency range from 834 MHz to 3.34 GHz, the rejection is better than 30 dB. Due to the peak introduced by the layout, in the range from dc to

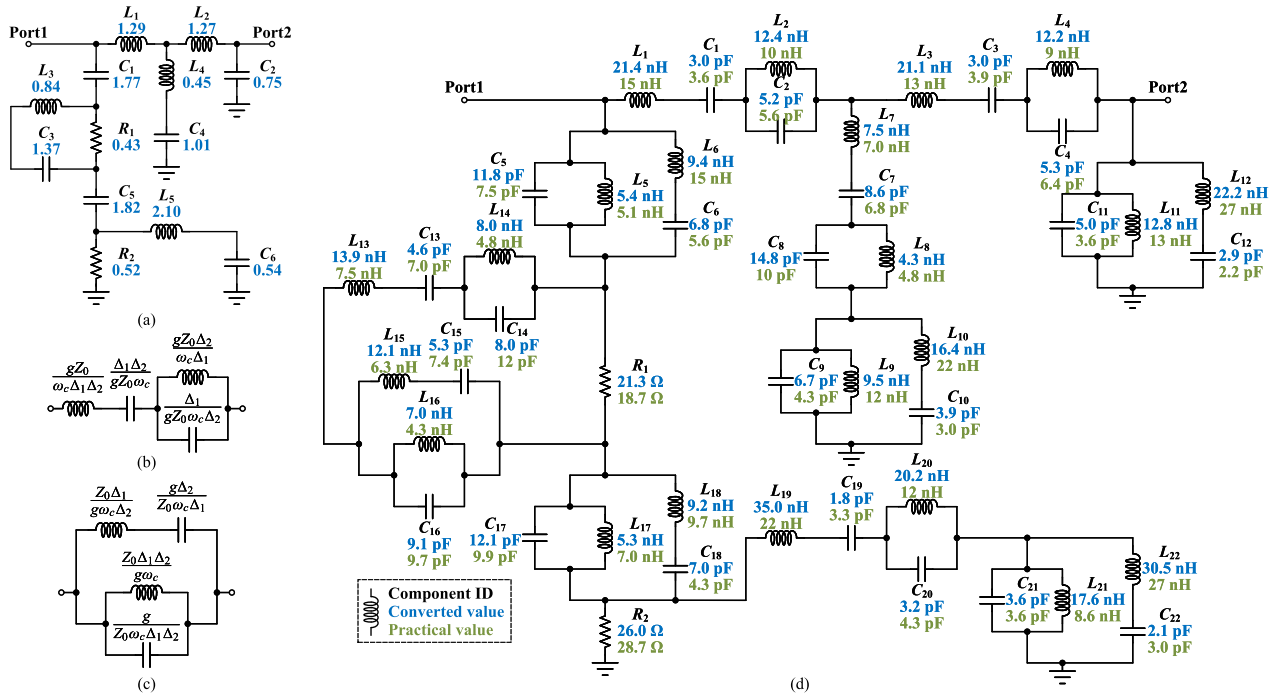


Fig. 27. Schematic of the synthesized LPF prototype, the low-pass-to-dual-band transformation, and the obtained dual-band BPF. (a) Schematic of the reflectionless LPF prototype. (b) Network converted from an inductor in the transformation. (c) Network converted from a capacitor in the transformation. (d) Schematic of the reflectionless dual-band BPF transformed from the prototype.

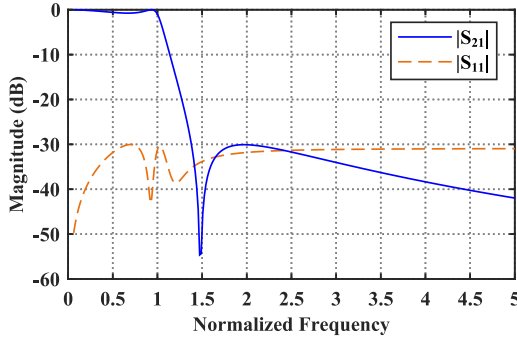


Fig. 28. Calculated response of the prototype.

3.11 GHz, the measured return loss is larger than 15 dB. The peak of return loss is controlled to 10 dB by gradient-based fine-tuning, which extends the frequency range of 10-dB return loss up to 5.59 GHz. A good agreement among the calculated, simulated, and measured results is achieved.

D. Dual-Band Reflectionless BPF

To synthesize a reflectionless dual-band BPF, an efficient method is using the low-pass-to-bandpass transformation twice. Compared with synthesizing the filter directly, which needs a topology of a considerably large scale, only a small-scale prototype is required when transformations are applied. This section provides an example of synthesizing dual-band BPF using such operations. Firstly, an LPF prototype is synthesized, and the schematic is presented in Fig. 27(a). The passband insertion loss is limited to 1 dB, the stopband rejection is set to 30 dB, and a TZ is added to the stopband. The max reflection coefficient is constrained to

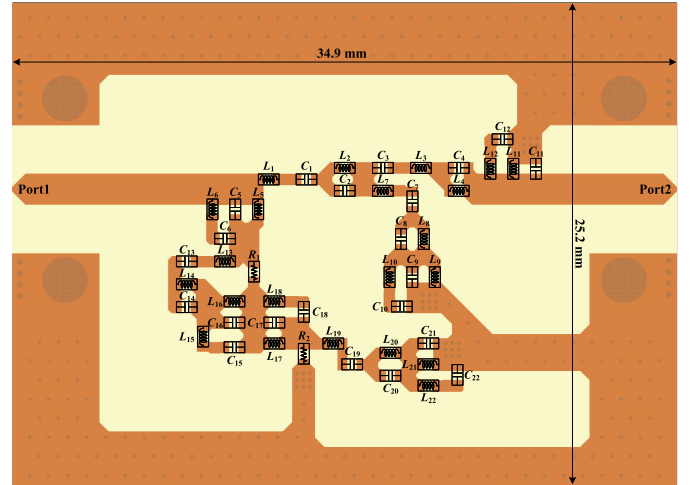


Fig. 29. Layout of the reflectionless dual-band BPF.

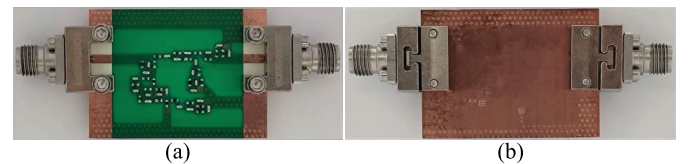


Fig. 30. Photographs of the fabricated reflectionless dual-band BPF. (a) Top view. (b) Bottom view.

−30 dB. The calculated response of this prototype is shown in Fig. 28. Then, the low-pass-to-bandpass transformation is applied twice, which forms the low-pass-to-dual-band transformation. In this transformation, each inductor of the prototype is converted into the small network of Fig. 27(b), while

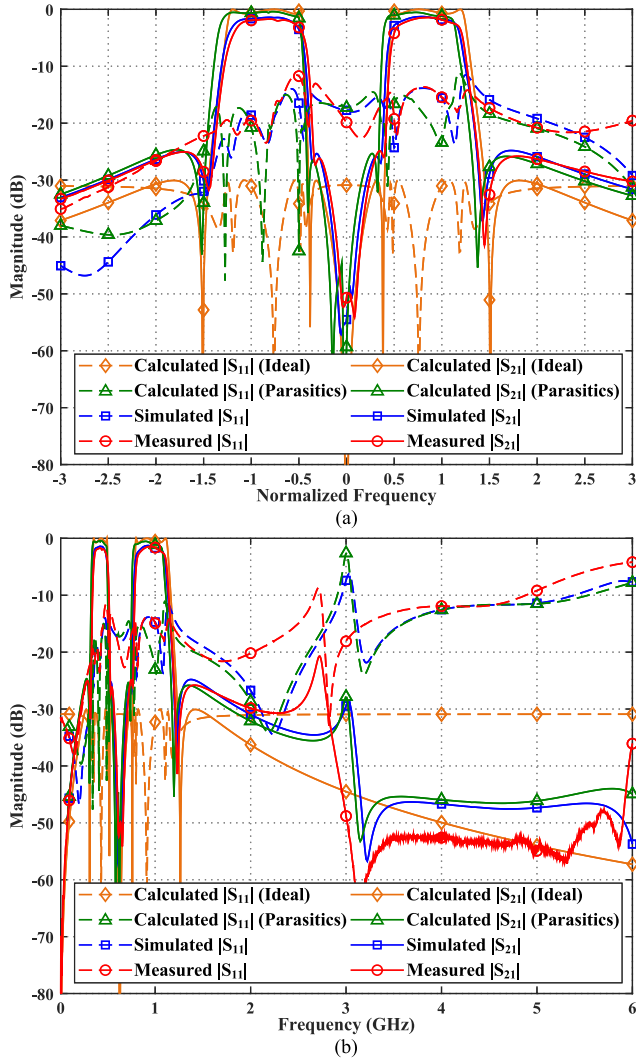


Fig. 31. Calculated, simulated, and measured results of the reflectionless dual-band BPF. (a) Response with frequency normalization using the center frequency of 629.4 MHz. (b) Response without frequency normalization.

each capacitor is converted into the network of Fig. 27(c). After that, the schematic presented in Fig. 27(d) is obtained. Note that in Fig. 27(b) and (c), g is the prototype value of the corresponding component, ω_c is the center frequency of the transformation and Δ_1 and Δ_2 are the FBW in the first and second transformations, respectively. Δ_1 only affects the bandwidths of the passbands, while Δ_2 affects both the bandwidths and the center frequencies of the two passbands. If the center frequencies of the two passbands are ω_1 and ω_2 , and $\omega_1 < \omega_2$, the value of ω_c is calculated as

$$\omega_c = \sqrt{\omega_1 \omega_2}. \quad (21)$$

At frequencies of ω_1 and ω_2 , the original frequency Ω before the second low-pass-to-bandpass transformation should satisfy the equation $|\Omega| = 1$. Thus, Δ_2 is derived as

$$\Delta_2 = \left(\sqrt{\frac{\omega_2}{\omega_1}} - \sqrt{\frac{\omega_1}{\omega_2}} \right). \quad (22)$$

In this case, the center frequencies are set to 433 and 915 MHz. By using (21), center frequency f_c is 629.4 MHz,

and ω_c is calculated by $2\pi f_c$. Meanwhile, Δ_2 is calculated from (22), and the value is 0.766. The calculated values of all components are labeled in blue in Fig. 27(d). Note that C_1 and C_5 in Fig. 27(a) can be simplified into a single capacitor. However, these components are kept in the transformation to provide more flexibility in fine-tuning after introducing parasitics.

The design layout of this reflectionless dual-band BPF is shown in Fig. 29. All values of components labeled in green in Fig. 27(d) are used in fabricating this filter, and the photographs are presented in Fig. 30. This PCB is implemented using the dielectric substrate of RO4350B with a thickness of 0.762 mm. The calculated, measured, and simulated results are shown in Fig. 31. In the measured response, the insertion loss at the center frequency of the lower passband is 1.70 dB, while it is 1.41 dB in the upper passband. The 3-dB passbands are 356–503 MHz (FBW of 34.2%) and 805–1119 MHz (FBW of 32.6%). The measured maximum peaks of insertion loss of the lower, middle, and higher stopbands near passbands are 25.2, 25.0, and 25.8 dB, respectively. The maximum reflection coefficient near passbands is -11.6 dB.

V. CONCLUSION

In this article, an adaptive synthesis procedure using lumped components is proposed and applied for synthesizing reflectionless filters. The algorithm can automatically select the suitable topology of given constraints through the specifically enhanced HGAPSO. The proposed HGAPSO is embedded with random neighborhoods and coordinates the search for improving the heuristics. Besides, a sorting-based classification is introduced for accelerating convergence, and a probabilistic method is used for boosting iteration. Components' values are fine-tuned using gradient-based optimization with parasitics of lumped components and layout being considered. To verify the effectiveness of the proposed algorithm, four reflectionless filters are synthesized and implemented. The results calculated with parasitics, the simulated results, and the measured results are matched accurately. The synthesized filters fulfill the design goals, which proves the effectiveness of the proposed methodology.

APPENDIX A FACTORS AFFECTING THE CONVERGENCE OF THE PROPOSED HGAPSO

Under a properly-configured hyperparameter, the proposed procedure is adaptive to select suitable topologies for synthesizing circuits with different response constraints. In this section, several factors affecting the convergence of the proposed HGAPSO are investigated by comparison. Each comparison is based on the setup in synthesizing the reflectionless BPF shown in Fig. 19. Note that the reflection coefficient is constrained to -15 dB for faster convergence during the benchmark. Every convergence curve is the median result of all tests.

All hyper-parameters configured in synthesizing the flat-passband BPF are used as a reference in testing the effect of different hyper-parameters. The PSO parameters w_g and

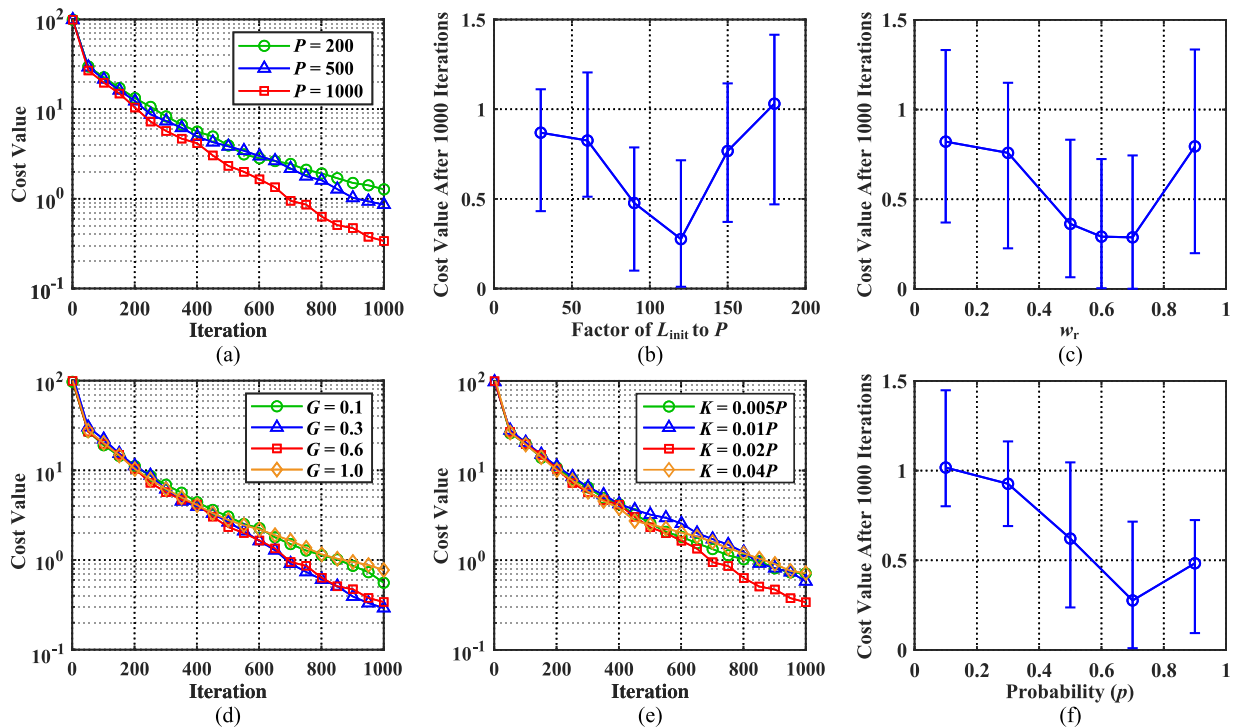


Fig. 32. Comparisons of the effect of different hyperparameters on the convergence. (a) Number of total particles P . (b) Factor of the lifespan of each particle L_{init} to P . (c) Factor w_r of the max step of random neighborhood search to the range of the classifier in each dimension. (d) Value G of the minimum gap between the upper and the lower bound of the classifier in each dimension. (e) Factor of K - P , where K is the number of optimal particles in generating the optimal zone. (f) Probability p of generating reinitialized particles in the optimal zone.

w_l are fixed to 0.024 and 0.016, respectively. Fig. 32 shows the effect of hyper-parameters introduced by the proposed HGAPSO. The convergence curves of different total number of particles (P) are shown in Fig. 32(a). It can be seen that more particles provide greater search capability, but the time consumption of each iteration is increased. The lifespan of each particle (L_{init}) is set proportionally related to parameter P . Since the lifespan of a particle is reduced by its rank in each iteration, the factor of L_{init} to P is the minimum searching time of each particle. As shown in Fig. 32(b), $L_{\text{init}} = 120 \cdot P$ is used to achieve the high particle efficiency during the convergence. In Fig. 32(c), the optimal weight factor w_r locates in the range of 0.6–0.7, which provides the proper searching range for random neighborhood search. The convergence curves of different values for the parameter G are exhibited in Fig. 32(d). This parameter defines the minimum gap between the upper bound and the lower bound in each dimension of the classifier. Setting G between 0.3 and 0.6 would now significantly affect the convergence. The population of optimal particles K is set proportionally related to parameter P . Using particles of the top 2% to update the classifier and define the optimal zone can obtain the best convergence performance, as shown in Fig. 32(e). According to the results in Fig. 32(f), the best probability of regenerating new particles in the optimal zone is around 0.7.

Meanwhile, constraints are factors that affect the convergence of the synthesis. Fig. 33(a) shows the effect of limiting the value boundaries (i.e., V_{min} and V_{max}). Limiting the range of components can avoid extreme values and reduce the

performance degradation caused by the parasites. However, once prototype values are limited to a very narrow range, the convergence is slow. Another factor is the constraint on the reflection coefficient. As shown in Fig. 33(b), the convergence is slower under tighter constraints. Fig. 33(c) shows the effect of the graph scale on the convergence. In Fig. 33(c), $E(G_0)$ and $C(G_0)$ denote the number of edges (i.e., the number of branches) and the number of components of the graph G_0 , respectively. If the scale of the graph used in the synthesis is larger than the graph that it actually needs, the synthesis requires more time to find a suitable topology. But a graph of a larger scale has more subgraphs, which ensures more circuits be obtained in the synthesis.

In short, setting hyper-parameters to unsuitable values will result in slower convergence. However, as shown in Fig. 32, slightly deviated parameters do not deteriorate the convergence significantly. If the synthesis constraints are changed while hyper-parameters are not configured to optimal values, suitable results can still be obtained after more iterations.

APPENDIX B TIME CONSUMING AND THE ACCELERATION OF PARALLELIZATION

The proposed HGAPSO is implemented in C programming language, with a simple built-in thread pool for multithreading. The platform is built by an Intel-D1581 with Ubuntu 20.04, and the program is compiled by GCC 10.3.0. The synthesis of the flat-passband BPF (Fig. 19) is used as a benchmark.

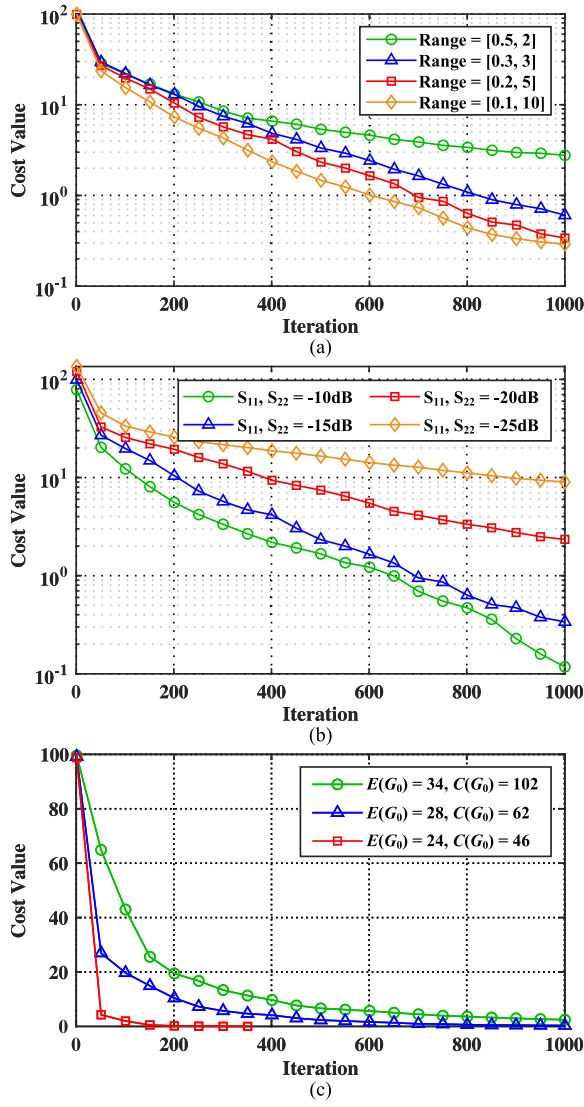


Fig. 33. Convergence curves of different constraints in synthesis. (a) Different limits on the range of prototype value. (b) Different constraints on reflection coefficient. (c) Initial graphs of different scales.

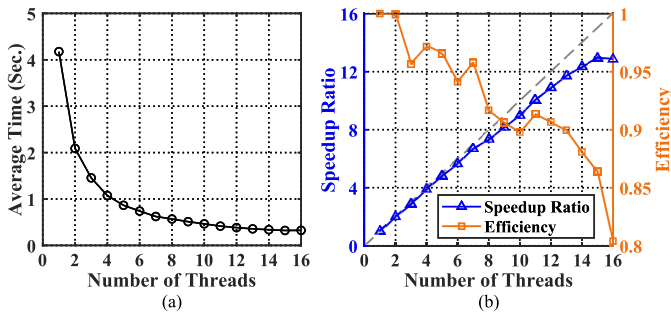


Fig. 34. Acceleration obtained by multithreading. (a) Average time consumption of each iteration versus the number of threads. (b) Speedup ratio and efficiency versus the number of threads.

The test result is shown in Fig. 34. The speedup ratio S_p and efficiency E_p are calculated by following:

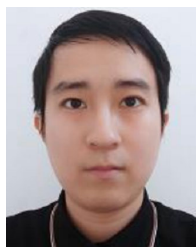
$$\begin{cases} S_p = T_1/T_p \\ E_p = S_p/p \end{cases} \quad (23)$$

where T_p is the average time consumption of each iteration using p threads. Because the cost value of each particle can be calculated individually, the speedup ratio of processing particles should be ideally linear. However, due to the additional process of submitting thread work, thread-switching in the system, and nonparallelized parts (i.e., sorting the cost value, calculating life span, and reinitializing particles), the speedup ratio is below the ideal value. Therefore, the efficiency of acceleration gradually decreases along with the total number of threads. Because the platform only has 16 CPU cores, the average time consumption is slightly longer when using 16 threads compared with 15 threads. Even so, the algorithm is benefited from parallelization. Besides, when 16 threads are used, the program dominates 15 MB of RAM in total (from the “RES” column in the “htop” processes viewer). In this test, the speedup ratio can reach up to 13 when 15 cores are used.

REFERENCES

- [1] D. M. Pozar, *Microwave Engineering*, 4th ed. New York, NY, USA: Wiley, 2012.
- [2] J.-S. Hong, *Microstrip Filters for RF/Microwave Applications*, 2nd ed. New York, NY, USA: Wiley, 2011.
- [3] R. R. Singh, *Circuit Theory and Networks—Analysis and Synthesis*, 2nd ed. McGraw-Hill, 2018.
- [4] M. A. Morgan and T. A. Boyd, “Theoretical and experimental study of a new class of reflectionless filter,” *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 5, pp. 1214–1221, May 2011.
- [5] M. A. Morgan and T. A. Boyd, “Reflectionless filter structures,” *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 4, pp. 1263–1271, Apr. 2015.
- [6] J.-Y. Shao and Y.-S. Lin, “Millimeter-wave bandstop filter with absorptive stopband,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, 2014, pp. 1–4.
- [7] B. Mini-Circuits, “Reflectionless filters improve linearity and dynamic range,” *Microw. J.*, vol. 58, no. 8, pp. 42–50, Aug. 2015.
- [8] H.-S. Yang, “A pulse-modulated RF power amplifier system with output direct absorptive band-pass filter connection,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2021, pp. 632–634.
- [9] R. Gomez-Garcia, D. Psychogiou, J. Munoz-Ferreras, and L. Yang, “Avoiding RF isolators: Reflectionless microwave bandpass filtering components for advanced RF front ends,” *IEEE Microw. Mag.*, vol. 21, no. 12, pp. 68–86, Dec. 2020.
- [10] M. Khalaj-Amirhosseini and M.-M. Taskhiri, “Twofold reflectionless filters of inverse-Chebyshev response with arbitrary attenuation,” *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 11, pp. 4616–4620, Nov. 2017.
- [11] C. Liu, Z. Deng, X. Liu, and X. Luo, “A wideband bandpass filter with broad stopband and ultra-wide reflectionless range for 5G applications,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2019, pp. 834–837.
- [12] W. Yu, Y. Rao, H. J. Qian, and X. Luo, “Reflectionless filtering 90° coupler using stacked cross coupled-line and loaded cross-stub,” *IEEE Microw. Wireless Compon. Lett.*, vol. 30, no. 5, pp. 481–484, May 2020.
- [13] W. Feng, X. Ma, Y. Shi, S. Shi, and W. Che, “High-selectivity narrow- and wide-band input-reflectionless bandpass filters with intercoupled dual-behavior resonators,” *IEEE Trans. Plasma Sci.*, vol. 48, no. 2, pp. 446–454, Feb. 2020.
- [14] J. Lee and J. Lee, “Transmission-line bandpass filter structures with infinite reflectionless range,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 6, pp. 2387–2398, Jun. 2022.
- [15] A. R. Djordjevic and A. G. Zajic, “Low-reflection bandpass filters with a flat group delay,” *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 4, pp. 1164–1167, Apr. 2005.
- [16] Y. Zhang, Y. Wu, W. Wang, and J. Yan, “High-performance common- and differential-mode reflectionless balanced band-pass filter using coupled ring resonator,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 974–978, Mar. 2022.
- [17] K. Zhao, R. Gómez-García, and D. Psychogiou, “Tunable quasi-reflectionless bandpass filters using substrate integrated coaxial resonators,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 379–383, Feb. 2022.

- [18] C. Luo et al., "Quasi-reflectionless microstrip bandpass filters using bandstop filter for out-of-band improvement," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 10, pp. 1849–1853, Oct. 2020.
- [19] Y.-H. Zhu, J. Cai, and J.-X. Chen, "Quasi-reflectionless double-sided parallel-strip line bandpass filter with enhanced selectivity," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 339–343, Feb. 2022.
- [20] X. Wu, Y. Li, and X. Liu, "High-order dual-port quasi-absorptive microstrip coupled-line bandpass filters," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1462–1475, Apr. 2020.
- [21] Z. Cao, X. Bi, and Q. Xu, "Tunable reflectionless filter with independently controllable dual passbands and absorbed harmonic signals," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 11, pp. 3416–3420, Nov. 2021.
- [22] X. Wu, Y. Li, and X. Liu, "Quasi-reflectionless microstrip bandpass filters with improved passband flatness and out-of-band rejection," *IEEE Access*, vol. 8, pp. 160500–160514, 2020.
- [23] M. A. Morgan, W. M. Groves, and T. A. Boyd, "Reflectionless filter topologies supporting arbitrary low-pass ladder prototypes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 2, pp. 594–604, Feb. 2019.
- [24] A. Guilabert, M. A. Morgan, and T. A. Boyd, "Reflectionless filters for generalized elliptic transmission functions," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 12, pp. 4606–4618, Dec. 2019.
- [25] X. Luo, B. Yang, and H. J. Qian, "Adaptive synthesis for resonator-coupled filters based on particle swarm optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 2, pp. 712–725, Feb. 2019.
- [26] J. Jin, F. Feng, J. Zhang, J. Ma, and Q.-J. Zhang, "Efficient EM topology optimization incorporating advanced matrix Padé via Lanczos and genetic algorithm for microwave design," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 8, pp. 3645–3666, Aug. 2021.
- [27] F. Feng, C. Zhang, W. Na, J. Zhang, W. Zhang, and Q. Zhang, "Adaptive feature zero assisted surrogate-based EM optimization for microwave filter design," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 1, pp. 2–4, Jan. 2019.
- [28] W.-B. Du, W. Ying, G. Yan, Y.-B. Zhu, and X.-B. Cao, "Heterogeneous strategy particle swarm optimization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 4, pp. 467–471, Apr. 2017.
- [29] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Dec. 1995, pp. 1942–1948.
- [30] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern., B*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.
- [31] F. Grimaccia, M. Mussetta, and R. E. Zich, "Genetical swarm optimization: Self-adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 781–785, Mar. 2007.
- [32] W. T. Li, X. W. Shi, Y. Q. Hei, S. F. Liu, and J. Zhu, "A hybrid optimization algorithm and its application for conformal array pattern synthesis," *IEEE Trans. Antennas Propag.*, vol. 58, no. 10, pp. 3401–3406, Oct. 2010.
- [33] J. Zhang, K. Ma, F. Feng, and Q. Zhang, "Parallel gradient-based local search accelerating particle swarm optimization for training microwave neural network models," in *IEEE MTT-S Int. Microw. Symp. Dig.*, May 2015, pp. 1–3.
- [34] Y. Yu, H. Qian, and Y.-Q. Hu, "Derivative-free optimization via classification," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2286–2292.
- [35] Y.-Q. Hu, H. Qian, and Y. Yu, "Sequential classification-based optimization for direct policy search," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2029–2035.
- [36] W. Chen, *Active Network Analysis: Feedback Amplifier Theory*, 2nd ed. Singapore: World Scientific, 2016.
- [37] M. A. Diniz-Ehrhardt, J. M. Martínez, and M. Raydan, "A derivative-free nonmonotone line-search technique for unconstrained optimization," *J. Comput. Appl. Math.*, vol. 219, no. 2, pp. 383–397, Oct. 2008.
- [38] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Rev.*, vol. 45, no. 3, pp. 385–482, Jan. 2003.



Yifan Li (Graduate Student Member, IEEE) received the B.E. degree in electronic science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2020, where he is currently pursuing the M.S. degree in electronic science and technology.

His research interests include circuit synthesis, digital-assisted radio frequency (RF) circuits, and data converters.



Xun Luo (Senior Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2005 and 2011, respectively.

From 2010 to 2013, he was with Huawei Technologies Company Ltd., Shenzhen, China, as the Project Manager to guide research and development projects of multiband microwave/millimeter-wave (mm-wave) integrated systems for backhaul and wireless communication. Before joining UESTC,

he was an Assistant Professor with the Department of Microelectronics, Delft University of Technology, Delft, The Netherlands. Since 2015, he has been with UESTC as a Full Professor, where he has been appointed as the Executive Director of the Center for Integrated Circuits. Since 2020, he has been the Head of the Center for Advanced Semiconductor and Integrated Micro-System (ASIS), UESTC. He has authored or coauthored more than 150 journals and conference papers. He holds 46 patents. His research interests include RF/microwave/mm-wave integrated circuits, multiple-resonance terahertz (THz) modules, multibands backhaul/wireless systems, reconfigurable passive circuits, smart antennas, and system in package.

Dr. Luo is a Technical Program Committee Member of multiple IEEE conferences, including the IEEE International Solid-State Circuits Conference (ISSCC), the IEEE Custom Integrated Circuits Conference (CICC), and the IEEE Radio Frequency Integrated Circuits (RFIC) Symposium. He is also the IEEE MTT-Society Technical Committee Member of MTT-4 on Microwave Passive Components and Transmission Line Structures, MTT-5 on Filters, and MTT-23 on Wireless Communications. He was bestowed by China as the China Overseas Chinese Contribution Award in 2016 and was selected by the IEEE MTT Society for the IEEE Outstanding Young Engineer Award in 2022. He with the Center for ASIS was a recipient of the UESTC Outstanding Team for Teaching and Education Award in 2021 and the UESTC Excellent Team for Postgraduate Supervision Award in 2021. He also won the UESTC Distinguished Innovation and Teaching Award in 2018 and the UESTC Outstanding Undergraduate Teaching Promotion Award in 2016. His Research Group BEAM X-Laboratory received multiple best paper awards and design competition awards, including the IEEE RFIC Best Student Paper Award in 2021, the IEEE RFIT Best Student Paper Award in 2016 and 2019, the IEEE IWS Best Student Paper Award in 2015 and 2018, the IEEE IMS Student Design Competition Award from 2017 to 2019, the IEEE IMS Sixty-Second Presentation Competition Award in 2019, and multiple best paper award finalists from the IEEE conferences. He was the TPC Co-Chair of the IEEE IWS in 2023, the IEEE RFIT in 2019, and the IEEE IWS in 2018. He is the Vice-Chair of the IEEE MTT-Society Chengdu Chapter. He serves as an Associate Editor for *IET Microwaves, Antennas and Propagation*. He was a Track Editor of IEEE MICROWAVE AND WIRELESS COMPONENTS LETTERS from 2018 to 2021.