A Quasi-Optimal Shape Design Method for Electromagnetic Scatterers Based on NURBS Surfaces and Filter-Enhanced GWO

Han Wang, Qiang Zou, and Hai Lin, *Member, IEEE*

***Abstract*—This paper presents a quasi-optimal and efficient method for designing the shape of electromagnetic (EM) scatterers. Optimizing the EM scatterer shape to achieve satisfactory scattering properties is a crucial, yet challenging problem due to the complex geometry involved and the high non-linearity in the optimization. In this paper, the geometric complexity is to be handled by modeling EM scatterer shapes as non-uniform rational basis spline (NURBS) surfaces, which enables an easy and intuitive shape control. Using the surfaces' control points as optimization variables, the shape optimization problem is to be solved with the heuristic optimization strategy, which can effectively handle non-linearity and the associated local optima. Specifically, the recently developed grey wolf optimizer (GWO) is employed. Although GWO can effectively avoid local optima and achieve quasi-optimal results, it requires repeated calls to time-consuming EM simulations. To solve this efficiency issue, a filter-enhanced version of GWO is proposed to reduce the times of calling EM simulations, and an efficient, adaptive remeshing method is proposed to accelerate the numerical analysis process within each EM simulation call. Altogether, they lead to a quasi-optimal and efficient method for optimizing EM scatterers. Its effectiveness has also been validated by several examples.**

***Index Terms*—Electromagnetic scatterer optimization; scattering simulation; NURBS surfaces; grey wolf optimizer; quadtree-based remeshing**

I. Introduction

THE shape of an electromagnetic (EM) scatterer is a dominant factor in determining its scattering property, and this property plays an important role in applications such as stealth aircraft design [1]. To have a satisfactory scattering property, shape optimization is quite often a good choice, especially for radar cross section (RCS) reduction tasks [1], [2]. It is, however, not a trivial matter because (1) an EM scatterer could have very complex geometry, implying that manipulation of EM scatterer shape is hard [3], and (2) the objective function involved in the shape optimization is highly non-linear, meaning that the relevant solving algorithms could be trapped at local optima or become unstable when the algorithms' parameters are not tuned properly.

Most existing work related to the above shape optimization problem focused on simple geometries and/or objectives in order to make it more tractable. For example, Bondeson et al. [4] considered 2D sections composed of sinusoidal bump functions; Toivanen et al. [5] analyzed similar 2D geometries, but defined by spline curves; Yang et al. [6] optimized 3D surfaces, but restricting them to be surfaces of revolution; Liu et al. [7] focused only on lengths of components in inter-chip antenna, and Kataja et al. [8] considered the separation of elements of Yagi-Uda antenna, among others. To solve the problems so defined, many utilized the gradient descent algorithm or its variants [4]–[6], [8]–[10]. Gradient descent is, however, prone to local optima. To solve this issue, evolutionary optimizers [11]–[15] have been introduced. As an early version of heuristic optimizers, evolutionary optimizers are able to handle simple or moderate problems but may have difficulties in situations of high non-linearity [16], which is the case for the problem considered in this work.

This paper considers 3D freeform geometries and represents them with non-uniform rational basis spline (NURBS) surfaces. The use of NURBS surfaces in EM simulation is not new and has been promoted in [17]–[19]. Despite their benefits of large design space and easy shape manipulation, NURBS surfaces introduce high non-linearity into the problem, which has yet been seriously studied. As an attempt to solve this issue, this work suggests a combined use of NURBS representation and the swarm intelligence optimizer. Particularly, the recently developed grey wolf optimizer (GWO) [16] is used to carry out the optimization. (The details of GWO will be given in Section III-B.) Compared with previous heuristic optimizers, GWO has demonstrated superior performance of convergence and local optima avoidance in solving highly non-linear, global optimization problems (see Section 4 of Ref. [16]).

Our implementation of the above idea shows that trivially combining NURBS and GWO is very slow due to the many times of calling EM simulations over the course of iterations. To solve this issue, a filter-enhanced version of GWO is further proposed to reduce the calls to EM simulations. Specifically, a nested filter structure is used, where the upper-level filter supports an efficient GWO searching, and the lower-level filter retains an accurate GWO searching. This paper also proposes an efficient, adaptive remeshing method to accelerate the numerical analysis process within each EM simulation call. Altogether, they provide a quasi-optimal[1] and efficient shape

This work was supported in part by the National Key Research and Development Program of China (No. 2020YFC2201302), the NNSF of China (No. 62102355), and the NSF of Zhejiang Province (No. LQ22F020012). *(Corresponding authors: Hai Lin, and Qiang Zou)*

The authors are with the State Key Laboratory of CAD and CG, College of Computer Science and Technique, Zhejiang University, Hangzhou 310058, China (e-mail: wanghanaviva@zju.edu.cn; qiangzou@cad.zju.edu.cn; lin@cad.zju.edu.cn)

[1] This method is deemed as quasi-optimal, not globally optimal, because although GWO can find global optimum in most practices, there is currently no theoretical guarantee that GWO can definitely find them, an issue suffered from by all heuristic optimizers.

optimization method for EM scatterers.

The rest of this paper is organized as follows. The overall framework of the proposed method is given in Section II. Some brief introductions to NURBS surfaces and GWO are given in Section III, based on which the details of filter-enhanced GWO are described in Section IV, and those of the adaptive remeshing presented in Section V. Validation of the method using a series of examples and comparisons can be found in Section VI, followed by conclusions in Section VII.

II. The Overall Framework

The proposed method consists of three modules (Fig. 1): filter-enhanced GWO, EM simulator, and fitness evaluator. Given an initial shape, the filter-enhanced GWO explores the design space and suggests shape modification candidates of high potential. The EM simulator then accepts those candidates and runs numerical analyses, going through two steps: (1) adaptive remeshing based on quadtrees; and (2) scattering evaluation using computational electromagnetic (CEM) methods such as the multilevel fast multipole algorithms (MLFMA) [20] and physical optics (PO) [21]. It inputs simulation results to the fitness evaluator, which in turn compiles them into fitness values to support the comparison and selection of shape candidates in the filter-enhanced GWO module. These three procedures will be repeated until a quasi-optimal result is reached.

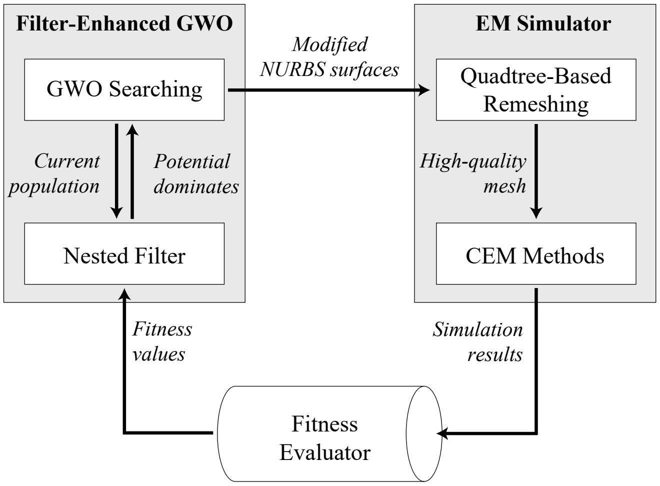


Fig. 1. The framework of the quasi-optimal shape design method, which consists of the filter-enhanced GWO, the EM simulator, and the fitness evaluator. The black arrows indicate the data flow.

III. Preliminaries

The following sections will present the detailed algorithms used to implement the modules outlined in Section II. To support the presentation, we now recall some facts about NURBS surfaces and the GWO algorithm.

A. NURBS Surfaces

NURBS is a parametric way of representing surfaces. The shape of the surface is defined in terms of a linear combination of a set of spline basis, where the weights are called control points. Mathematically, a NURBS surface $\boldsymbol{S}(u,v)$ is constructed as follows [22]:

$$\boldsymbol{S}(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}R_{i,j}(u,v)\boldsymbol{P}_{i,j} \quad 0\le u,v\le 1. \tag{1}$$

where $(m+1)(n+1)$ control points $\boldsymbol{P}_{i,j}$ make the control polyhedron of the surface, $(m+1)(n+1)$ basis functions $R_{i,j}$ form the set of spline basis. The basis function $R_{i,j}$ with degree p in the u direction and degree q in the v direction is a piecewise rational basis function of the form

$$R_{i,j}(u,v)=\frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{m}\sum_{l=0}^{n}N_{k,p}(u)N_{l,q}(v)w_{k,l}} \tag{2}$$

where $N_{i,p}$ and $N_{i,q}$ are the nonrational B-spline basis functions, and the $(m+1)(n+1)$ weights $w_{i,j}$ associated with control points define the rational property of $R_{i,j}$.

As can be seen from its definition, every point on a NURBS surface is a linear combination of control points, and because of the compactly supported basis $R_{i,j}$, the control points participating in the linear combination regarding a single point are just a small subset of all control points. As such, modifying one control point only causes local shape changes. Shape manipulation is thus intuitive and predictable. Such a local control property also allows NURBS surfaces to take complex local details, resulting in a quite large design space for EM scatterers.

B. Original GWO

GWO is a recently proposed heuristic optimizer [16], [23], with demonstrated superior ability of local optima avoidance. This algorithm simulates the hunting mechanism of grey wolves in nature, particularly the social hierarchy of grey wolves. It consists primarily of two steps: encircling prey and hunting the prey. In the encircling step, all wolves (candidate solutions) update their positions according to those of leader wolves (elite solutions) to get close to them. This collective behavior has been mathematically modeled using the following equations:

$$\begin{aligned}\boldsymbol{D}&=ABS(\boldsymbol{C}\odot\boldsymbol{x}_{\xi}^{t}-\boldsymbol{x}^{t})\\ \boldsymbol{x}^{t+1}&=\boldsymbol{x}^{t}-\boldsymbol{A}\odot\boldsymbol{D}\end{aligned} \tag{3}$$

where the function ABS returns the absolute value element-wise, operator $\odot$ calculates the element-wise product of two vectors, t indicates the current iteration, $\boldsymbol{x}_{\xi}^{t}$ denotes the leader wolf (wolves), $\boldsymbol{x}^{t}$ is a wolf undergoing position updating. $\boldsymbol{A}$ and $\boldsymbol{C}$ are coefficients used to inject randomness into the position update, which are defined as:

$$\boldsymbol{A}=2\boldsymbol{a}\odot\boldsymbol{r}_1-\boldsymbol{a}, \quad \boldsymbol{C}=2\boldsymbol{r}_2$$

where each component of $\boldsymbol{a}$ is decreasing from 2 to 0 over the course of iterations, and $\boldsymbol{r}_1,\boldsymbol{r}_2$ are random vectors in $[0,1]$. For the physical/social meanings of these coefficients, please refer to [23].

In the hunting step, leader wolves are supposed to guide the hunt because they have better knowledge of the potential optima. It is suggested in the literature [23] that leader wolves are chosen as the three best solutions obtained so far. Let them be represented by $\boldsymbol{x}_{\alpha,t}$, $\boldsymbol{x}_{\beta,t}$, and $\boldsymbol{x}_{\delta,t}$. Other wolves will update their positions accordingly in the following way:

$$\boldsymbol{x}^{t+1} = \frac{\boldsymbol{x}_{\alpha}^{t+1} + \boldsymbol{x}_{\beta}^{t+1} + \boldsymbol{x}_{\delta}^{t+1}}{3} \tag{4}$$

where $\boldsymbol{x}_{\alpha}^{t+1}$ is calculated by substituting $\boldsymbol{x}_{\xi}^{t}$ with $\boldsymbol{x}_{\alpha,t}$ in (3), and the same for $\boldsymbol{x}_{\beta}^{t+1}$, and $\boldsymbol{x}_{\delta}^{t+1}$.

IV. Filter-Enhanced GWO

After choosing the NURBS representation scheme, design parameters of the EM optimization problem consist of the surface's control points and their related weights. Each wolf's position thus takes the form of a vector of those control points and weights, to be denoted as $\boldsymbol{x}$. The fitness or dominance of a wolf is determined by the scattered field $\boldsymbol{E}^s$ associated with it, which can be obtained by solving the electric field integral equation (EFIE):

$$\hat{\boldsymbol{n}} \times (\boldsymbol{E}^i + \boldsymbol{E}^s(\boldsymbol{x})) = \boldsymbol{0} \tag{5}$$

with

$$\boldsymbol{E}^s(\boldsymbol{x}) = -j\omega\mu_0 \int \overline{\overline{G}} \cdot \boldsymbol{J}(\boldsymbol{x}) d\tau' \tag{6}$$

where $\hat{\boldsymbol{n}}$ is the normal of the scatterer's surface, $\boldsymbol{E}^i$ is the incident electric field, ω is the angular frequency of EM wave, μ_0 the permeability in free space, and $\overline{\overline{G}}$ the dyadic Green's function. This is a general formulation of fitness determination. For specific tasks, for example RCS reduction, it will transform into a more specific form:

$$f(\boldsymbol{x}) = -\sum_{i=1}^{N_{fre}} \sum_{j=1}^{N_{deg}} \sigma(\boldsymbol{E}^s(\boldsymbol{x}))_{i,j} \tag{7}$$

where $f(\boldsymbol{x})$ denotes the fitness function (The negative sign in the fitness function indicates that wolf with smaller RCS has better fitness.), N_{fre} is the number of frequency points of interest, N_{deg} is the number of observation angles that can be elevation, azimuth, or any arbitrary combination, and σ is the RCS at i^{th} frequency for j^{th} angle, which is given by:

$$\sigma(\boldsymbol{E}^s(\boldsymbol{x})) = \lim_{r\to\infty} 4\pi r^2 \frac{|\boldsymbol{E}^s(\boldsymbol{x})|^2}{|\boldsymbol{E}^i|^2}. \tag{8}$$

Note that $\boldsymbol{E}^s$ and $\boldsymbol{E}^i$ can be any type of polarization based on the optimization goal of interest.

With optimization variables (i.e., control points) and the optimization objective (i.e., fitness values) in place, the GWO method is readily applicable. Our direct application, however, shows that it is very slow. The reason for this is that, because of the large number of control points, the method needs to run EM simulations (i.e., solving (8)) many times, which is very time-consuming. In view of this issue, a filter-enhanced version of GWO is proposed to much reduce calls to EM simulations.

The basic idea of the filter-enhanced GWO is to divide wolves into two types: one for those requiring precise fitness values, and the other for which approximated fitness values are enough. Then, EM simulations are only necessary for wolves of the former type. The division is possible because there are two search modes for each wolf in GWO: exploration and exploitation; the exploration mode means that wolves diverge from each other to search for regions potentially containing the optimum, while under the exploitation mode wolves converge to the optimum. In exploration, approximated fitness values can also work because we only want to check the "potential" of a region, a qualitative measure [23]. It is the exploitation mode where precise fitness values (and thus EM simulations) come into play.

The GWO algorithm has already provided a parameter, i.e., the random parameter $\boldsymbol{A}$, to switch between exploration and exploitation. When $|\boldsymbol{A}| \geq 1$, wolves are forced to diverge. This emphasizes exploration and allows the GWO algorithm to search globally to find regions of higher potential. $|\boldsymbol{A}| < 1$ lets wolves focus on a region and converge towards the optimum in this region. As such, we use the value of $\boldsymbol{A}$ to drive the decision on whether a precise or an approximated fitness value is to be used for a search wolf. Roughly speaking, when $|\boldsymbol{A}| < 1$, the fitness value is precisely evaluated (i.e., calling EM simulations); otherwise, an approximated fitness value is generated.

We can further reduce wolves that require precise fitness values. During exploitation (i.e., when $|\boldsymbol{A}| < 1$), leader wolves have better knowledge about the potential location of the optimum [23]. Based on this intuition, an additional condition can be added to narrow down wolves that require precise fitness values to those near leader wolves. That is, by dividing exploitation wolves into elite wolves that are near to leader wolves and ordinary ones that are not, we carry out EM simulations only for elite wolves and use approximations for ordinary ones. This division of wolves, together with the above division into exploration and exploitation, can significantly reduce calls to EM simulations.

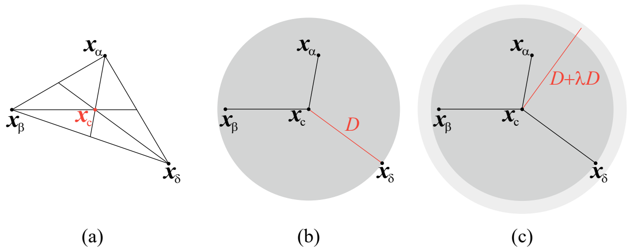
To mathematically model the nearness between a given exploitation wolf and leader wolves, the Euclidean distances between their positions (i.e., vectorized control points) are used. Specifically, let an exploitation wolf be represented by $\boldsymbol{x}$, the three leader wolves at this iteration represented by $\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\delta$. (Note that three leader wolves not just one are chosen at each iteration in GWO, as already noted in Section III-B.) Their nearness is expressed in terms of the Euclidean distance between $\boldsymbol{x}$ and the center of the three wolves, as follows:

$$\begin{aligned} d &= ||\boldsymbol{x} - \boldsymbol{x}_c||_2 \\ \boldsymbol{x}_c &= \frac{\boldsymbol{x}_\alpha + \boldsymbol{x}_\beta + \boldsymbol{x}_\delta}{3} \end{aligned} \tag{9}$$

The threshold used to determine if a given exploitation wolf is near the leader wolves or not is based on the region influenced by them. The circumscribed sphere about the leader wolves is used to define the influence region. This sphere is also slightly enlarged (e.g., by 10% of the circumscribed sphere radius) to include the local influence region of each leader wolf. Specifically, the threshold is given by the following two procedures:

$$\begin{aligned} &D \leftarrow \max ||\boldsymbol{x}_\xi - \boldsymbol{x}_c||_2, \quad \xi = \alpha, \beta, \delta \\ &D \leftarrow D + \lambda D \end{aligned} \tag{10}$$

where D is the threshold, and λ the enlargement parameter.



(a) (b) (c)

Fig. 2. The calculation of the threshold used to determine the elite wolves. (a) The center $\boldsymbol{x}_c$ of the three leader wolves. (b) The threshold D is first defined by the maximum Euclidean distance from the center to the leader wolves. (c) The circumscribed sphere is slightly enlarged based on the enlargement parameter λ.

To generate approximated fitness values, this work employs the Gaussian Process (GP) [24] method to approximate the shape-fitness relationship, based on a set of samples generated by the Latin hypercube sampling (LHS) [25]. (For specific mathematical and algorithmic details of GP, please refer to [24]. We omit the details here since we do not consider applying GP as new results.) The GP method is preferred because it can be automatically adapted to new samples, and this is the case in this work where calling EM simulations produces new samples. As the essence of approximation is to filter out fine, detailed information and extract the rough, structural information underlying the shape-fitness relationship, the proposed method is named filter-enhanced GWO.

Overall, a hierarchical filter structure is designed to enhance GWO. It consists of a global GP and a local one. The global GP generates approximated fitness values for wolves to explore globally. The local GP gives approximated fitness values for non-elite exploitation wolves to converge locally. For those elite exploitation wolves, their fitness values are given by EM simulations. Their simulation results are then input into both GPs to refine the approximations. Details about the two GPs are presented below.

The global GP is prepared at the initialization process for the exploration wolves. Because the exploration wolves could be anywhere in the searching space, training samples for the global GP should cover the whole searching space. However, gathering necessary training samples for the global GP requires numerous precise fitness value evaluations. To address this issue, we generate the training samples by introducing the additional asymptotic fitness values. The asymptotic fitness value is evaluated by asymptotic algorithms (i.e, PO). The asymptotic algorithms can generate low-fidelity simulation results with hundreds of times speedup compared to the precise algorithms (i.e., MLFMA). Based on the fact that the asymptotic fitness value and precise fitness value are physically related, the global GP can be trained with a limited number of training samples with the additional input. Specifically, the input $\boldsymbol{x}_{add}$ of a training sample consists of the design parameters $\boldsymbol{x}$ and its related asymptotic fitness value y_{asy}, and the desired output is the precise fitness value.

The local GP is constructed after the classification of elite and non-elite exploitation wolves. With the operators in GWO, the exploitation wolves are located around the center of leading wolves. Based on this fact, the sampling area of the local GP is much smaller than the searching space. Thus, the local GP only requires a limited number of training samples. Meanwhile, the elite wolves and previous data (i.e., data used for constructing the global GP) within the sampling area can be reused for constructing the local GP. Thus, construction of the local GP requires a few calls of EM simulation, and the local GP is trained directly from precise sampling data (The input of the local GP is the design parameter $\boldsymbol{x}$, and the desired output is the precise fitness value).

Algorithm 1 Filter-Enhanced GWO

```
/* initialization */
$M_g, population \leftarrow initialize()$
$\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\delta \leftarrow filter(population, M_g)$
$t \leftarrow 0$
while $t < max\_iter\_number$ do
  /* update the position of each search agent */
  $exploration, exploitation, elite \leftarrow []$
  for $\boldsymbol{x}$ in $population$ do
    $update\_position(\boldsymbol{x})$
    if $|\boldsymbol{A}| \geq 1$ then
      $append(exploration, \boldsymbol{x})$
    else
      $append(exploitation, \boldsymbol{x})$
      if $||\boldsymbol{x} - \boldsymbol{x}_c|| < D$ then
        $append(elite, \boldsymbol{x})$
      end if
    end if
  end for
  /* evaluate precise fitness values for elite wolves */
  for $\boldsymbol{x}$ in $elite$ do
    $evaluate\_fitness(\boldsymbol{x})$
  end for
  /* update GPs */
  $M_l \leftarrow construct\_local\_GP(elite)$
  $M_g \leftarrow update\_global\_GP(elite)$
  /* filter the leader wolves */
  $P \leftarrow []$
  $append(P, filter(exploration, M_g))$
  $append(P, filter(exploitation, M_l))$
  $\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\delta \leftarrow update\_leader\_wolves(P)$
  $t \leftarrow t + 1$
end while
return $\boldsymbol{x}_\alpha$
```

The pseudo-code of the filter-enhanced GWO is presented in Algorithm. 1. Some details of the proposed method are noted below:

- M_g and M_l indicates the global GP and local GP, respectively. $population$ is the grey wolf population.
- The $filter$ function filters the leader wolves by the expected improvement (EI) [26]. Search agents that have the largest probability to be the potential wolves will be selected from the population by this function.
- The $update\_position$ function updates the position of a search agent by (4).

- In each iteration, the grey wolves are classified based on $|\boldsymbol{A}|$ and the threshold D.
- The fitness values of elite grey wolves are generated through the precise fitness value evaluator. These data are then used for constructing the local GP and updating the global GP.
- The potential leader wolves are filtered from the exploration wolves and exploitation wolves, respectively, and stored in P. Then the precise fitness values of the potential leader wolves filtered by the GPs are computed and used to update the leader wolves in $update\_leader\_wolves$ function.

V. Quadtree-Based Remeshing

In the filter-enhanced GWO, precise fitness values are calculated through EM simulations, i.e., evaluating (6). Such equations are typically solved using CEM methods, which require a mesh of the NURBS-based design surface. As there are repeated EM simulations over the course of shape optimization, many generations of surface meshing are needed, which is time-consuming. To improve efficiency, this work resorts to mesh reuse such that, for a new design parameter, its corresponding mesh is not generated from scratch but a modified version of an existing mesh. This is to be done by adaptively subdividing/coarsening the existing mesh to accommodate the new design, based on a quadtree representation of the mesh.

A. Quadtree Representation

Each NURBS surface is associated with a quadtree structure as illustrated in Fig. 3. The root node of the quadtree represents the $[0,1] \times [0,1]$ parameter domain of NURBS, which is recursively subdivided into two or four child nodes based on the quality of mesh elements. For example, the edge length of the triangular mesh should be no larger than a threshold l_m (i.e, $l_m = 0.1\lambda$).

Four element split schemes are designed to carry out the subdivision. Let a node's corresponding parameter domain be represented by $[u_a, u_b] \times [v_a, v_b]$. For this parameter domain, there exists a corresponding 3D quadrilateral. Further, let l_u denote the length of the longer u-directional edge of the quadrilateral, and similarly for l_v for v-directional edges. Then the four schemes are as follows:

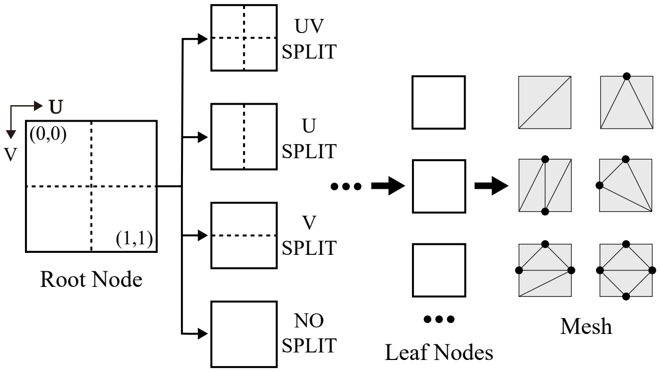


Fig. 3. The quadtree structure and the related mesh generation. The root node representing parameter domain $[0,1] \times [0,1]$ has four children, whose split types are UV SPLIT, U SPLIT, V SPLIT, and NO SPLIT from top to bottom. A set of leaf nodes are produced after the recursive domain subdividing, and the triangular mesh is generated by partitioning the leaf nodes into triangles. Six vertex insertion cases are presented to avoid vertex inconsistencies.

- UV SPLIT: $l_u > l_m, l_v > l_m$. The u-direction and v-direction of the parameter domain are both subdivided, and the node furcates into four children with parameter domains $[u_a, (u_a + u_b)/2] \times [v_a, (v_a + v_b)/2]$, $[u_a, (u_a + u_b)/2] \times [(v_a + v_b)/2, v_b]$, $[(u_a + u_b)/2, u_b] \times [v_a, (v_a + v_b)/2]$ and $[(u_a + u_b)/2, u_b] \times [(v_a + v_b)/2, v_b]$.
- U SPLIT: $l_u > l_m, l_v < l_m$. The u-direction of the parameter domain is subdivided, and the node furcates into two children with parameter domains $[u_a, (u_a + u_b)/2] \times [v_a, v_b]$ and $[(u_a + u_b)/2, u_b] \times [v_a, v_b]$.
- V SPLIT: $l_u < l_m, l_v > l_m$. The v-direction of the parameter domain is subdivided, and the node furcates into two children with parameter domains $[u_a, u_b] \times [v_a, (v_a + v_b)/2]$ and $[u_a, u_b] \times [(v_a + v_b)/2, v_b]$.
- NO SPLIT: $l_u < l_m, l_v < l_m$, there is no need to subdivide the parameter domain and the node is a leaf node.

As described above, the edge lengths l_u and l_v are the key factors affecting the quadtree subdivision process. Mathematical formulations for calculating the edge lengths are given in the following. Let l_{u1} and l_{u2} denote the lengths of the two edges of a quadrilateral along u-direction. Then edge length l_u can be derived by:

$$\begin{aligned} l_u &= \max(l_{u1}, l_{u2}) \\ l_{u1} &= \int_{u_a}^{u_b} \|\boldsymbol{S}_u(u, v_a)\|_2 du = \sum_{i=0}^{K} w_i \|\boldsymbol{S}_u(u_i, v_a)\|_2 \\ l_{u2} &= \int_{u_a}^{u_b} \|\boldsymbol{S}_u(u, v_b)\|_2 du = \sum_{i=0}^{K} w_i \|\boldsymbol{S}_u(u_i, v_b)\|_2 \end{aligned} \tag{11}$$

where K decides the accuracy of the numerical quadrature, the quadrature weight w_i is normalized with respect to the range of quadrature, and the partial derivative with respect to u of NURBS surface $\boldsymbol{S}_u$ can be found in [22]. A similar procedure can be carried out for calculation of l_v.

Algorithm 2 Quadtree Construction

```
1: $current\_layer \leftarrow [root]$
2: while $current\_layer$ is not empty do
3:     $next\_layer \leftarrow []$
4:     for $node$ in $current\_layer$ do
5:         $node.split\_scheme \leftarrow calc\_split\_scheme(node)$
6:         $append(next\_layer, generate\_child(node))$
7:     end for
8:     $clear(current\_layer)$
9:     $current\_layer \leftarrow next\_layer$
10: end while
```

The pseudo-code of the quadtree construction method is presented in Algorithm. 2. In line 1, the $current\_layer$ array is initialized by the $root$ node of the quadtree. In lines 4-7, each node in the $current\_layer$ is subdivided through one of the four split schemes, and the new child nodes will be stored in $next\_layer$. In lines 8-9, the $current\_layer$ is updated by the $next\_layer$. Finally, the loop ends when the $current\_layer$ is empty.

The above procedure results in a regular mesh because each triangle is constructed based on the edge length criterion threshold l_m. The regular mesh is suitable for precise algorithms such as MLFMA [27], but it is a waste for asymptotic algorithms such as PO and results in an efficiency reduction [28], [29]. To obtain the optimal mesh for PO, namely, coarser mesh in areas with a higher radius of curvature and finer mesh in areas with a lower radius of curvature, a simple modification of the proposed construction method is enough. Specifically, take the radius of curvature as the meshing criteria, and change the four split schemes based on the radius of curvature of the parametric domain. In this implementation, only the edge length criterion is considered because the procedure has to fulfill the requirements of the precise algorithm, and the efficiency reduction resulting from the regular mesh in asymptotic algorithms does not have a significant impact on the final optimization results.

B. Quadtree-Based Remeshing

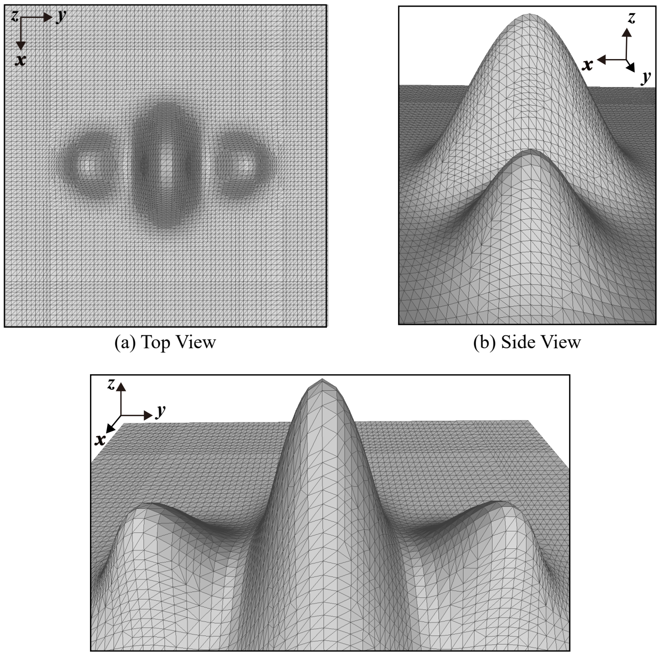
For a new design parameter, the original quadtree structure may not meet the mesh quality requirements. The reason is that the edge lengths in (11) are changed after the modification of NURBS surfaces, which will influence the split scheme of each node. Thus, in the remeshing process, the structure of the quadtree nodes should be adaptively adjusted at first. The basic idea of the adjustment is to subdivide leaf nodes whose new edge lengths are larger than l_m, delete the subtrees of non-leaf nodes whose new edge lengths are smaller than l_m, and rebuild the subtrees of non-leaf nodes whose split scheme should be changed.

The pseudo-code of the adjustment is presented in Algorithm. 3, where the $create\_subtree$ function creates a new subtree for the $node$ based on the new split scheme, and the $delete\_subtree$ function deletes the subtree and related triangular mesh for the $node$. After this structure adjustment, triangular meshes are generated only for those changed nodes, based on the same meshing scheme illustrated in Fig. 3.

In the practical implementation, using the quadtree structure described above could lead to vertex inconsistencies on the adjacent edges of two leaf nodes if they happen to have different split types, or they are at different levels. To solve this issue, we can insert new vertices at the inconsistency regions as illustrated in the right column of Fig. 3. The six vertex insertion cases are classified based on the number and positions of inconsistencies. After the vertex insertion, triangular mesh can be obtained by connecting vertices in each quadrilateral. Note that there may be several different choices to connect the vertices (Fig. 3 only shows one of them for each case), we can adaptively choose one that can generate high-quality triangles. Fig. 4 presents the mesh generated by the above procedures. The modified NURBS surface is obtained by moving the surface's control points. We can see that high-quality mesh is obtained even at the protrusion parts.

Algorithm 3 Quadtree Adjustment

```
$current\_layer \leftarrow [root]$
**while** $current\_layer$ is not empty **do**
  $next\_layer \leftarrow []$
  **for** $node$ in $current\_layer$ **do**
    $old\_scheme \leftarrow node.split\_scheme$
    $new\_scheme \leftarrow calc\_split\_scheme(node)$
    $node.split\_scheme \leftarrow new\_scheme$
    **if** $new\_scheme$ is the same as $old\_scheme$ **then**
      $append(next\_layer, node.child\_nodes)$
    **else if** $old\_scheme$ is NO_SPLIT **then**
      $create\_subtree(node)$
    **else if** $new\_scheme$ is NO_SPLIT **then**
      $delete\_subtree(node)$
    **else**
      $delete\_subtree(node)$
      $create\_subtree(node)$
    **end if**
  **end for**
  $clear(current\_layer)$
  $current\_layer \leftarrow next\_layer$
**end while**
```



(a) Top View

(b) Side View

(c) Front View

Fig. 4. The mesh of a modified NURBS surface generated by the quadtree-based remeshing. The modified NURBS surface is obtained by moving the control points' positions of a square surface. (a) The top view of the modified NURBS surface. (b) The side view of the protrusion parts. (c) The front view of the protrusion parts.

VI. Results and Discussions

Several shape optimization examples are presented in this section to validate the proposed method. The EM simulator used in these examples is our own implementation, whose accuracy and efficiency have been validated in [30], [31]. In these examples, comparisons with the trivial GWO verify

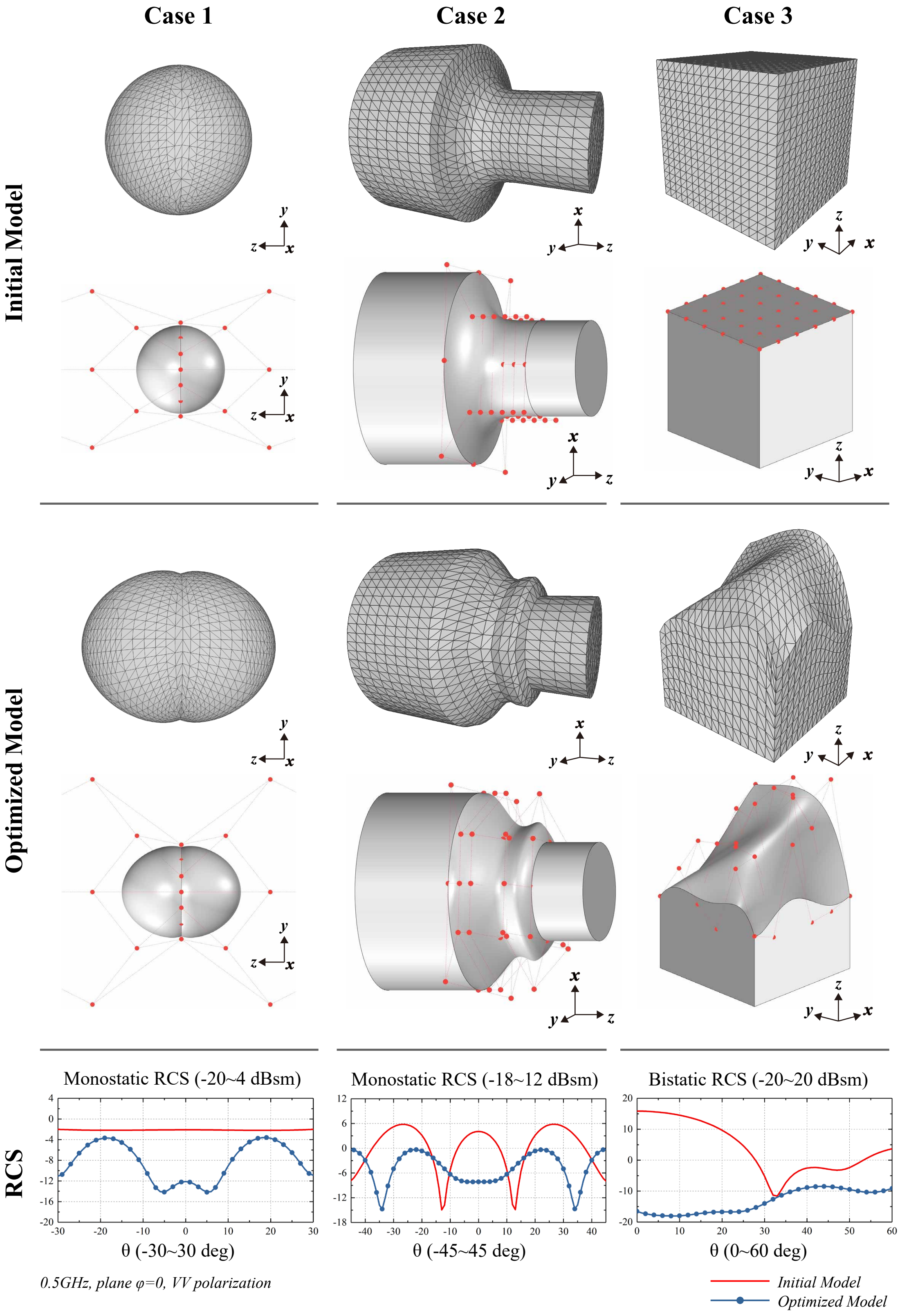


Fig. 5. The initial models, models optimized by the proposed method, the related mesh, and the RCS profiles of the three cases. The red dots around the models indicate the control points of NURBS surfaces, whose locations or related weights form the design parameters (For the sake of clarity, only the significant part of the NURBS control polygon is given in case 1). The design parameter in case 1 is formed by combining weights and locations of the control points. In other cases, only the locations of control points are optimized. In addition, the incident wave is transmitted in the opposite direction of the z-axis in case 3.

TABLE I
STATISTICAL RESULTS AND COMPUTATIONAL RESOURCES REQUIREMENTS OF THE FIVE OPTIMIZATION EXPERIMENTS IN THE THREE CASES.

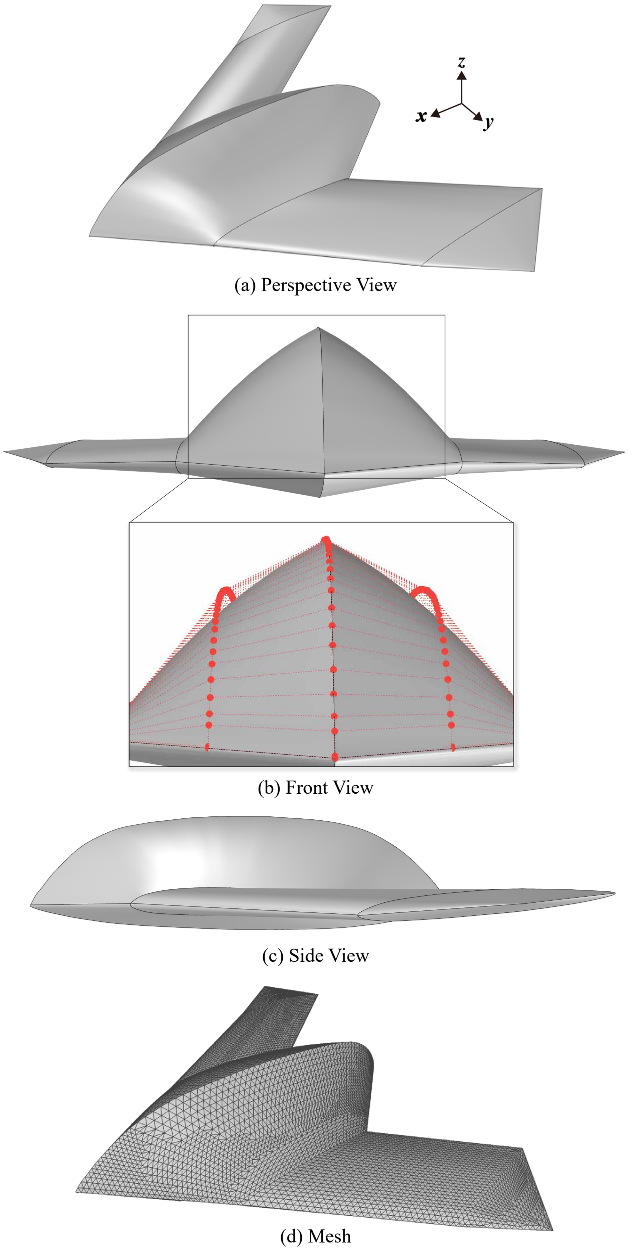
Case	Dimension[1]	Variables[2]	Fitness (dB)[3]				Simulation Calls[4]			Optimization Time (h)		
			Initial	GWO	Proposed	Difference[5]	GWO	Proposed	Percentage Decrease[6]	GWO	Proposed	Speedup[7]
Case 1	3	50	1.56	7.42	7.41	0.01	1515	356	76.5%	11.38	2.11	5.74
Case 2	5	45	-2.20	3.28	3.22	0.06	2525	376	85.1%	13.42	2.07	6.48
Case 3	36	36	-3.97	11.56	11.36	0.20	18180	2601	85.7%	30.60	3.63	8.43

[1] The dimension of the design parameter. Each design candidate in the design parameter is varied within the range of variation to constraint deformation.
[2] The number of control points' locations or the related weights varied during the optimization. The dimension and the number of variables are different because one design candidate may control several variables to preserve the general shape. For example, to preserve the central symmetry in case 1, control points and their related weights are divided into three central symmetric groups, and each group is controlled by one design candidate.
[3] The fitness values of the initial model and optimized models calculated by (7).
[4] The number of times to call precise EM simulation during the optimization process.
[5] The optimized fitness difference between the trivial GWO and the proposed method.
[6] The ratio of the simulation calls reduction of the proposed method to the trivial GWO.
[7] The optimization time speedup of the proposed method over the trivial GWO.

the efficiency of the proposed method, and a realistic shape optimization case demonstrates its ability in solving the practical problem. The design parameters in these examples are chosen based on the optimization requirements, which are the weights, locations of control points, or any of their arbitrary combinations. In the implementation, the size of the grey wolf population is $5d$ where d is the dimension of the search agent, and the maximum iteration number is 100. In addition, it should be noted that the angles θ and ϕ in these examples follow the definition of the spherical coordinate system, where the elevation angle θ is measured from the z-axis, and the azimuth angle ϕ is the horizontal angle measured from the x-axis.

At first, we choose three RCS reduction cases to compare our results with the trivial GWO. To eliminate the influence of randomness, we made five experiments on each optimization method for these cases. The optimization results of the proposed method are given in Fig. 5. It can be seen that the RCS values of the three cases are reduced after the optimization. Note that special features such as the symmetry of the optimized models in these cases cannot be generated/preserved automatically without specific deformation constraints. The symmetry in case 1 and case 2 are reserved by linking symmetric control points and weights to the same design candidate in the design parameter. Instead, no deformation constraint is imposed to maintain the symmetry in case 3, but its asymmetric result with a 15.33 fitness value increment conforms to the quasi-optimal declaration of the proposed method as well. Moreover, the statistical results and computational resources requirements are presented in Table I. These results indicate that the proposed method can provide similar results to the trivial GWO even at 36-dimensional problems. The slight difference between the two methods is owing to the error of approximate fitness values evaluated by the GPs. Meanwhile, the proposed method has at least 5 × speedup over the trivial GWO. We can conclude that the proposed method can provide competitive results with greatly improved efficiency.

Second, to verify the practical application ability of the proposed method, the shape of a realistic aircraft is optimized for RCS reduction (see Fig. 6). There are 30 design candidates controlling the locations or weights of 244 control points. The aircraft is excited by a V-polarized plane wave. The operating frequency ranges from 1.0 GHz to 2.0 GHz with a step size



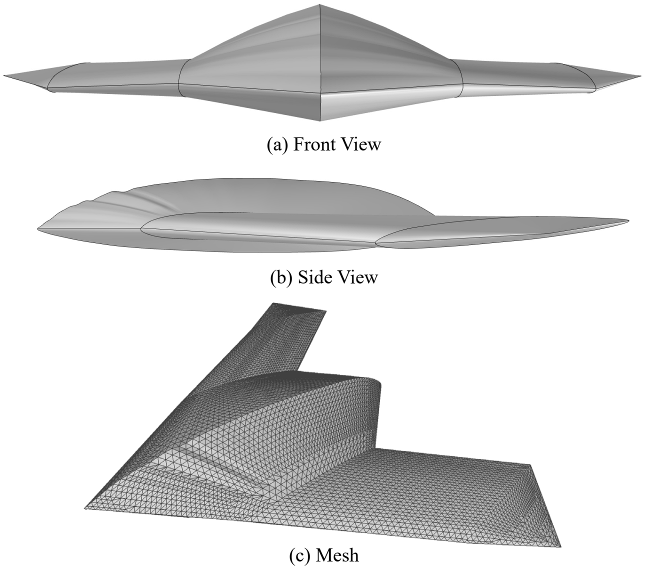
(a) Perspective View

(b) Front View

(c) Side View

(d) Mesh

Fig. 6. The initial realistic aircraft model. The red dots around the model indicate the control points of NURBS surfaces, whose locations form the design parameter. The design parameter of the aircraft consists of 30 design candidates controlling locations of 244 control points.



(a) Front View

(b) Side View

(c) Mesh

Fig. 7. The aircraft model optimized by the proposed method.

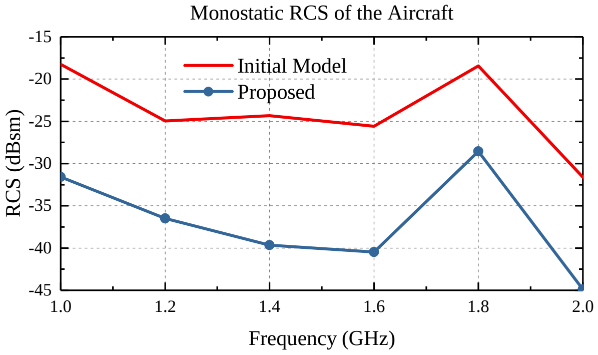


Fig. 8. The bistatic RCS profiles of the initial model and the model optimized by the proposed method. The fitness value of the initial model is 23.86 dB, and the fitness value of the optimized model is 36.95 dB.

of 0.2 GHz. And the one monostatic angle considered in this case is $\theta = 90^0, \phi = 0^0$. The model optimized by the proposed method is presented in Fig. 7. The RCS values of the initial model and the optimized models are shown in Fig. 8. After optimization, the fitness value increases by 13.09 dB. We call 2677 times precise EM simulation in total and the optimization time is 115.23 hours. The optimization result of the trivial GWO is not presented due to the unacceptable optimization time (It requires 15150 simulation calls).

From these numerical results, it can be concluded that the proposed method can provide efficient and quasi-optimal results for the EM shape optimization problems. The efficiency profits from the reduced simulation times and reused meshes. And the quasi-optimality results from the uncertainty of the heuristic optimizer and the error of the approximated fitness value. While increasing the training samples of the global GP and the enlargement parameter λ in (10) can improve the optimization result, it is at the expense of efficiency. The choice of the two parameters is the trade-off between efficiency and the optimality of the results. In addition, based on the fact that the stable mesh is reused for fast numerical analysis, strategies of reusing previous simulation results can also help fast recomputing in EM simulation (i.e., constructing credible initial guess when solving linear matrix equations by iterative solvers).

VII. Conclusions

A new method has been presented in this paper to design the shape of electromagnetic (EM) scatterers. The main features of this method include the quasi-optimality of the designed EM scatterer and efficiency of the algorithm. These features are essentially achieved by (1) combining Gaussian processes with GWO to avoid being trapped in local minimums while keeping the computation load low, and (2) reusing meshes for fast numerical analysis. New/improved methods have been presented to implement these two steps, and a series of case studies and comparisons have been conducted to validate the method.

It should be noted that the proposed method is unable to handle multi-objective optimization. This is because the hierarchical filter structure lacks the ability of filtering the potential leader wolves through various trade-offs between the objectives. Future work will focus on solving this issue by applying the multi-objective version of the GWO algorithm and an improved filter strategy.

References

[1] M. Li, J. Chen, X. Feng, F. Qu, and J. Bai, “An efficient adjoint method for the aero-stealth shape optimization design,” *Aerospace Science and Technology*, vol. 118, p. 107017, 2021.

[2] L. Zhou, J. Huang, Z. Gao, and W. Zhang, “Three-dimensional aerodynamic/stealth optimization based on adjoint sensitivity analysis for scattering problem,” *AIAA Journal*, vol. 58, no. 6, pp. 2702–2715, 2020.

[3] J. S. Ryu, Y. Yao, C. S. Koh, S. Yun, and D. S. Kim, “Optimal shape design of 3-D nonlinear electromagnetic devices using parameterized design sensitivity analysis,” *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1792–1795, 2005.

[4] A. Bondeson, Y. Yang, and P. Weinerfelt, “Shape optimization for radar cross sections by a gradient method,” *International Journal for Numerical Methods in Engineering*, vol. 61, no. 5, pp. 687–715, 2004.

[5] J. I. Toivanen, R. A. Mäkinen, J. Rahola, S. Järvenpää, and P. Ylä-Oijala, “Gradient-based shape optimisation of ultra-wideband antennas parameterised using splines,” *IET Microwaves, Antennas & Propagation*, vol. 4, no. 9, pp. 1406–1414, 2010.

[6] Y. Yang, T. Hallerod, D. Ericsson, A. Hellervik, A. Bondeson, and T. Rylander, “Gradient optimization of microwave devices using continuum design sensitivities from the adjoint problem,” *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1780–1783, 2005.

[7] B. Liu, H. Aliakbarian, Z. Ma, G. A. Vandenbosch, G. Gielen, and P. Excell, “An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques,” *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 1, pp. 7–18, 2013.

[8] J. Kataja, S. Järvenpää, J. I. Toivanen, R. A. Mäkinen, and P. Ylä-Oijala, “Shape sensitivity analysis and gradient-based optimization of large structures using MLFMA,” *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 11, pp. 5610–5618, 2014.

[9] R. Safian, N. K. Nikolova, M. H. Bakr, and J. Bandler, “Feasible adjoint sensitivity technique for EM design exploiting Broyden’s update,” in *IEEE MTT-S International Microwave Symposium Digest, 2003*, vol. 1. IEEE, 2003, pp. 299–302.

[10] M. S. Dadash, N. K. Nikolova, and J. W. Bandler, “Analytical adjoint sensitivity formula for the scattering parameters of metallic structures,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 9, pp. 2713–2722, 2012.

[11] A. Hoorfar, “Evolutionary programming in electromagnetic optimization: a review,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 523–537, 2007.

[12] S. Koulouridis and J. L. Volakis, “A novel planar conformal antenna designed with splines,” *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 34–36, 2008.

[13] N. Baatar, D. Zhang, and C.-S. Koh, “An improved differential evolution algorithm adopting λ-best mutation strategy for global optimization of electromagnetic devices,” *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2097–2100, 2013.

[14] Z. He, Y.-S. Li, P.-F. Gu, K. W. Leung, and R.-S. Chen, “A space-mapping-based optimal EM design of RCS reduction for electrically large targets,” *IEEE Transactions on Antennas and Propagation*, 2021.

[15] B. Liu, M. O. Akinsolu, C. Song, Q. Hua, P. Excell, Q. Xu, Y. Huang, and M. A. Imran, “An efficient method for complex antenna design based on a self adaptive surrogate model-assisted optimization technique,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 4, pp. 2302–2315, 2021.

[16] S. Mirjalili, I. Aljarah, M. Mafarja, A. A. Heidari, and H. Faris, “Grey wolf optimizer: theory, literature review, and application in computational fluid dynamics problems,” *Nature-Inspired Optimizers*, pp. 87–105, 2020.

[17] M. Chen, Y. Zhang, X.-W. Zhao, and C.-H. Liang, “Analysis of antenna around NURBS surface with hybrid MoM-PO technique,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 2, pp. 407–413, 2007.

[18] H. Yuan, N. Wang, and C. Liang, “Combining the higher order method of moments with geometric modeling by NURBS surfaces,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 11, pp. 3558–3563, 2009.

[19] K. Wang, Z. He, D. Ding, and R. Chen, “Uncertainty scattering analysis of 3-D objects with varying shape based on method of moments,” *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 4, pp. 2835–2840, 2019.

[20] J. Song, C.-C. Lu, and W. C. Chew, “Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 10, pp. 1488–1493, 1997.

[21] M. N. Vesperinas, *Scattering and Diffraction in Physical Optics*. World Scientific Publishing Company, 2006.

[22] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.

[23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[24] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[25] M. Stein, “Large sample properties of simulations using Latin hypercube sampling,” *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

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

[27] M.-K. Li and W. C. Chew, “Multiscale simulation of complex structures using equivalence principle algorithm with high-order field point sampling scheme,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2389–2397, 2008.

[28] J. Asvestas, “The physical optics fields of an aperture on a perfectly conducting screen in terms of line integrals,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 9, pp. 1155–1159, 1986.

[29] E. Martini, G. Pelosi, and S. Selleri, “Line integral representation of physical optics scattering from a perfectly conducting plate illuminated by a Gaussian beam modeled as a complex point source,” *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 10, pp. 2793–2800, 2003.

[30] Y. Zhang and H. Lin, “A new preconditioning scheme for MLFMA based on null-field generation technique,” *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 289–292, 2014.

[31] Y. Zhang and H. Lin, “MLFMA-PO hybrid technique for efficient analysis of electrically large structures,” *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 1676–1679, 2014.

Han Wang received the B.Eng. degree in software engineering from Wuhan University, Wuhan, China, in 2019. She is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. Her current research interests include numerical methods, fast algorithms, and shape optimization.

Qiang Zou is currently an assistant professor of Computer Science at Zhejiang University, China. He got his Ph.D. from University of British Columbia, Canada. His research interests lie in design modeling and manufacturing simulation

Hai Lin received the B.Eng. and M.Eng. degrees from Xidian University, Xi’an, China, in 1987 and 1990, respectively, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2000.

He was a Research Fellow in medical visualization with De Montfort University, Leicester, U.K., from 2000 to 2003. He was a Visiting Professor with the Department of Computing and Information Systems, University of Bedfordshire, Luton, U.K., from 2013 to 2016. He is currently a Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His current research interests include graphical electromagnetic computing, wireless channel modeling, computer graphics, scientific visualization, and volume rendering.