SADEA-II: A Generalized Method for Efficient Global Optimization of Antenna Design

Liu, B., Koziel, S., Ali, N

SADEA-II: A Generalized Method for Efficient Global Optimization of Antenna Design

Bo Liu[a], Slawomir Koziel[b], Nazar Ali[c]

[a] *Department of Computing, Wrexham Glyndwr University and School of Electrical, Electronic and System Engineering, The University of Birmingham, U.K.*
[b] *Engineering Optimization & Modeling Centre, Reykjavik University, Iceland.*
[c] *Department of Electronic and Electrical Engineering, Khalifa University, UAE.*

Abstract

Efficiency improvement is of great significance for simulation-driven antenna design optimization methods based on evolutionary algorithms (EAs). The two main efficiency enhancement methods exploit data-driven surrogate models and / or multi-fidelity simulation models to assist EAs. However, optimization methods based on the latter either need ad-hoc low-fidelity model setup or have difficulties in handling problems with more than a few design variables, which is a main barrier for industrial applications. To address this issue, a generalized three stage multi-fidelity-simulation-model assisted antenna design optimization framework is proposed in this paper. The main ideas include introduction of a novel data mining stage handling the discrepancy between simulation models of different fidelities, and a surrogate-model-assisted combined global and local search stage for efficient high-fidelity simulation model-based optimization. This framework is then applied to SADEA, which is a state-of-the-art surrogate-model-assisted antenna design optimization method, constructing SADEA-II. Experimental results indicate that SADEA-II successfully handles various discrepancy between simulation models and considerably outperforms SADEA in terms of computational efficiency while ensuring improved design quality.

Keywords: Antenna design optimization, antenna design automation,

Email addresses: `b.liu.3@bham.ac.uk`, `b.liu@glyndwr.ac.uk` (Bo Liu), `koziel@ru.is` (Slawomir Koziel), `ntali@kustar.ac.ae` (Nazar Ali)

Surrogate-model-assisted evolutionary algorithm, Expensive optimization, Multi-fidelity, Variable fidelity, Gaussian process

1. Introduction

In recent years, evolutionary algorithms (EAs) have been playing an important role in antenna design optimization [1, 2, 3, 4] due to their global optimization capability, free of an initial design, generality and robustness. High-quality design results have been obtained, but computational efficiency of the optimization process is still a major challenge. Although analytical models [5, 6] and fast electromagnetic (EM) simulation methods [7] address efficient optimization for some particular types of antennas and make significant contributions, a more generalized method employing standard EM simulation is needed to complement the state-of-the-arts [8]. Given several thousands to tens of thousands of EM simulations required by a standard EA to converge, and the cost of several tens of minutes per EM simulation, efficiency improvement without compromising performance is highly desirable.

A general method to improve the optimization efficiency is to introduce data-driven surrogate modeling and coupling it with EAs [9, 10, 11]. Using the antenna design parameters as the input and EM-simulated responses as the output, a computationally cheap surrogate model (which is often based on statistical learning techniques) is constructed and is used to replace potentially numerous computationally expensive EM simulations in optimization. Pioneer methods are [9] and [10], applying the EGO method [12] and the ParEGO method [13] from the computational intelligence field. These pioneer research works largely decrease the number of EM simulations, but the main challenge is that considerable efficiency improvement is difficult to be maintained if the number of design variables is larger than a few [11, 14].

The surrogate-model-assisted differential evolution for antenna optimization (SADEA) method has been proposed in [11] to address this problem. Although SADEA ensures generality, scalability (for up to around 30 design variables) and efficiency (4-8 times speed improvement compared to standard differential evolution (DE) and particle swarm optimization (PSO)), it is more suitable for problems with less than 20 minutes / simulation. In many industrial applications, depending on the structure complexity and other circumstances (e.g., housing), the cost of a reasonably accurate full-wave EM simulation may be 40 minutes or more when using a regular PC machine [8]. Therefore, further substantial efficiency improvement is needed for SADEA for its industrial use, which is the goal of this work.

A straightforward idea for further speed improvement is to introduce multi-fidelity EM-simulation models to SADEA. This concept has been widely used in local antenna optimization [8] and other domains. The general idea is to use cheaper but less accurate low-fidelity EM models to filter out non-promising solutions, and to use expensive but accurate high-fidelity EM models to perform local search around "promising" solutions obtained by the low-fidelity EM model.

The major challenge of the above method is the reliability. The success comes from the basic assumption that the optimal points of landscapes based on low- and high-fidelity EM models are close to each other [14, 15]. Otherwise; local search may be performed in an area far from the true optimum. However, the validity of the above assumption depends on the fidelity used of the low-fidelity model, which is problem dependent. Handling discrepancy between the EM simulation models of different fidelities is the main obstacle for using multi-fidelity antenna optimization methods in industrial software, because the selection and setup of the appropriate low-fidelity model is ad-hoc [15].

This problem has been also a challenge in the computational intelligence field until now. To the best of our knowledge, the only reliable solution is [16], which iteratively updates a co-kriging surrogate model [17] using samples from low- and high-fidelity simulations accumulated over the entire optimization process. This method is general and reliable because a mathematically sound co-kriging surrogate model uses information from multiple fidelity simulation models to address the discrepancy. Moreover, this method has been applied to antenna optimization [18]. However, scalability is the main challenge, since for problems with more than a few design variables, the computational cost of obtaining sufficient number of samples to build a high-quality co-kriging model is often prohibitive.

One of our main objectives is to combine generality, reliability and scalability to handle discrepancy between simulation models of different fidelities. To address this problem, a novel data mining method is developed considering characteristics of the antenna design landscape. Subsequently, a three stage multi-fidelity antenna optimization framework is proposed. This framework is then combined with SADEA for further substantial speed enhancement. The new method is thus called SADEA-II. The major goals of SADEA-II are:

- to considerably reduce computational effort compared to SADEA, so that global optimization can be realized in reasonable timeframe even for problems requiring 40 minutes to 1 hour / high-fidelity EM simulation.
- to provide highly optimized results which are better than SADEA.
- to ensure sufficient generality so that different types of antenna structures and various low-fidelity EM model selections (including various types of discrepancy between the EM models) can be reliably and efficiently handled.

The remainder of this paper is organized as follows. Section 2 provides the basic techniques. Section 3 introduces the SADEA-II framework, including its general structure, the three optimization stages and the parameter setting. Section 4 discusses verification results of SADEA-II. Concluding remarks are presented in Section 5.

2. Basic Techniques

2.1. A Brief Description to Gaussian Process Surrogate Modeling and Lower Confidence Bound Prescreening

In SADEA-II, Gaussian Process (GP) surrogate modeling [19] is used to construct data-driven surrogate models. A brief introduction is as follows. More details are in [19].

To model an unknown function $y = f(x), x \in R^d$, GP modeling assumes that $f(x)$ at any point x is a Gaussian random variable $N(\mu, \sigma^2)$, where μ and σ are two constants independent of x. For any x, $f(x)$ is a sample of $\mu + \epsilon(x)$, where $\epsilon(x) \sim N(0, \sigma^2)$. The similarity between two points x^i and x^j can be defined by a correlation function $c(x^i, x^j)$. Hyper-parameters are included in it. By maximizing the likelihood function that $f(x) = y^i$ at $x = x^i$ $(i = 1, \ldots, K)$ (where $x^1, \ldots, x^K \in R^d$ and their f-function values $y^1, \ldots, y^K$ are K training data points), the optimal hyper-parameter values can be obtained. Using best linear unbiased prediction, the predicted value $\hat{f}(x)$ of a new point x is as follows:

$$\hat{f}(x) = \hat{\mu} + r^T C^{-1}(y - \mathbf{1}\hat{\mu}) \tag{1}$$

the mean squared error is:

$$s^2(x) = \hat{\sigma}^2[1 - r^T C^{-1} r + \frac{(1 - \mathbf{1}^T C^{-1} r)^2}{\mathbf{1}^T C^{-1} r}] \tag{2}$$

where

$$\hat{\mu} = (I^T C^{-1} y)^{-1} I^T C^{-1} y \tag{3}$$

$$\hat{\sigma^2} = (y - I\hat{\mu})^T C^{-1} (y - I\hat{\mu}) n^{-1} \tag{4}$$

$r = (c(x, x^1), \ldots, c(x, x^K))^T$. C is a $K \times K$ matrix whose (i, j)-element is $c(x^i, x^j)$. $y = (y^1, \ldots, y^K)^T$ and $\mathbf{1}$ is a K-dimensional column vector of ones.

The above method is called ordinary GP. Blind GP [20] is used in SADEA-II. In blind GP, the linear combination of m basis functions $\sum_{i=1}^{m} \beta_i b_i(x)$ is used to replace $\hat{\mu}$ to capture a portion of the variations. The goal is to represent the general trend of the function to be approximated, so as to alleviate the complexity of the ordinary GP modeling, which handles the residuals. Blind GP often has better approximation ability, especially when the number of design variables is larger [20].

The blind GP modeling consists of the following steps: (1) based on the available training data points, an ordinary GP model is firstly constructed by identifying the hyper-parameter values; (2) given the hyper-parameters and the candidate features, the basis functions $b_i(x)$ are ranked based on the estimated $\beta_i (i = 1, \ldots, m)$. The ranking follows a Bayes variable ranking method [20, 21]. For simplicity and efficiency, only linear, quadratic items and two-factor interactions are considered as the basis functions in this implementation; (3) the most promising candidates among $b_i(x) (i = 1, \ldots, m)$ are selected and an intermediate GP model with the original hyper-parameters is constructed. Its accuracy is subsequently evaluated by a leave-one-out cross-validation method [20]. This step is repeated until no accuracy improvement can be achieved; (4) given the selected $b_i(x)$ and the corresponding coefficients β_i, the likelihood function is re-optimized and the final GP model is obtained. The details can be found in [21].

For a minimization problem, given the predictive distribution $N(\hat{f}(x), s^2(x))$ for $f(x)$, a lower confidence bound (LCB) prescreening of $f(x)$ can be used to promote explorative global search:

$$\begin{aligned} & f_{lcb}(x) = \hat{f}(x) - \omega s(x) \\ & \omega \in [0, 3] \end{aligned} \tag{5}$$

where ω is a constant, which is often set to 2. More details can be found in [22].

2.2. A Brief Description to Differential Evolution Algorithm

Differential Evolution (DE) [23] is a popular population-based metaheuristic algorithm for continuous optimization and is used in SADEA-II. There are a few DE mutation strategies available which lead to various trade-offs between the convergence rate and the population diversity. The properties of different DE mutation strategies under the SADEA framework have been investigated in [24]. Based on [24] and our pilot experiments, DE/current-to-best/1 (6) and DE/best/1 (7) are used in SADEA-II.

Suppose that P is a population and the best individual in P is x^{best}. Let $x = (x_1, \ldots, x_d) \in R^d$ (d is the number of decision variables) be an individual solution in P. To generate a child solution $u = (u_1, \ldots, u_d)$ for x, DE/current-to-best/1 and DE/best/1 work as follows: A donor vector is first produced by mutation:

(1)DE/current-to-best/1

$$v^i = x^i + F \cdot (x^{best} - x^i) + F \cdot (x^{r_1} - x^{r_2}) \tag{6}$$

where x^i is the i^{th} individual in P. x^{r_1} and x^{r_2} are two different solutions randomly selected from P; they are also different from x^{best} and x^i. v^i is the i^{th} mutant vector in the population after mutation. $F \in (0, 2]$ is a control parameter, often referred to as the scaling factor [23].

(2)DE/best/1

$$v^i = x^{best} + F \cdot (x^{r_1} - x^{r_2}) \tag{7}$$

Having the donor vector, the following crossover operator is applied to produce the child u:

1 Randomly select a variable index $j_{rand} \in \{1, \ldots, d\}$,

2 For each $j = 1$ to d, generate a uniformly distributed random number $rand$ from $(0, 1)$ and set:

$$u^i_j = \begin{cases} v^i_j, & \text{if } (rand \leq CR) | j = j_{rand} \\ x^i_j, & \text{otherwise} \end{cases} \tag{8}$$

where $CR \in [0, 1]$ is a constant called the crossover rate.

2.3. The SADEA Method

The SADEA method works as follows. More details can be found in [11].

Step 1: Sample α (often a small number of) candidate designs from the design space $[a, b]^d$ (a and b are the lower and upper bounds of design variables, respectively) using Latin Hypercube Sampling [25], evaluate the objective function values of all these solutions using EM simulations and let them form the initial database.

Step 2: If a preset stopping criterion is met (e.g., a maximum number of allowed EM simulations is exceeded), output the best solution from the database; otherwise go to Step 3.

Step 3: Select the λ best solutions from the database to form a population P.

Step 4: Apply the DE mutation (6) and crossover (8) operations to P to generate λ new child solutions.

Step 5: Select τ nearest candidate designs from the database (based on Euclidean distance in the design space) around the centroid of the λ child solutions. Construct a blind GP surrogate model using the selected candidate designs (i.e., training data points in surrogate modeling).

Step 6: Estimate the λ child solutions generated in Step 4 using the blind GP model and lower confidence bound method.

Step 7: Evaluate the EM simulation model at the estimated best child design candidate from Step 6. Add this candidate design and its objective function value to the database. Go back to Step 2.

The advantages on efficiency and scalability of SADEA come from high-quality surrogate modeling and the balance between exploration and exploitation. In particular, the training samples are located near to the points waiting to be predicted (child population in Step 4), better surrogate model quality and prediction results are therefore obtained with the same number of training data points than surrogate model-assisted EAs with standard EA structures.

It is shown that this framework ensures comparable results but uses considerably fewer number of exact evaluations compared to several popular surrogate model-assisted EA frameworks, as verified using more than ten benchmark test problems [26, 24].

3. The SADEA-II Method

3.1. Key Ideas and General Structure

For simplicity, only two EM models are utilized: the low-fidelity model is referred to as the coarse model, whereas the high-fidelity one is referred to as the fine model. For multi-fidelity optimization, it is essential that useful information must be extracted from the computationally cheap coarse model to support fine model evaluation (FE)-based optimization. Hence, it is worth to study the discrepancy between landscapes based on coarse model-based evaluation (CE) and FE, especially when CE is sufficiently cheap (but is often too expensive for standard EAs).

Six different types of industrial antennas with different EM model fidelities are studied and some optimization runs are carried out using CEs. The following observations are obtained: (1) When CE is sufficiently cheap, the EM response feature is largely misrepresented by the coarse model. (2) The best design based on CE is often far from optimal in terms of FE when CE is sufficiently cheap. (3) Even when the response feature is largely misrepresented, there are still often a small number of optimal designs in terms of CE which are fair in terms of FE. (4) There are also some fair designs in terms of FE among the points visited by the CE-based optimization, although their CE values are poor.

Based on the above observations, it can be seen that the points visited by CE-based optimization represent meaningful positions of the design space. Although the true optima are often not among them, useful patterns which may lead to truly optimal designs in FE-based search are included in those visited points. Note that the useful points cannot be directly detected by CE values due to the discrepancy. Hence, the key questions become how to use as few

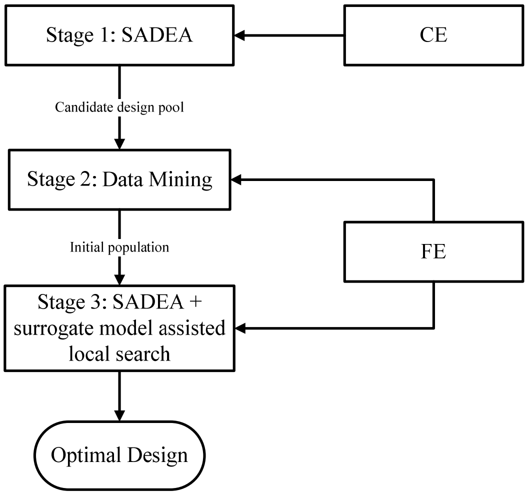

Figure 1: Flow diagram of SADEA-II

number of FEs as possible to detect a portion of the useful points visited by CE-based optimization and how to use them to support FE-based optimization.

To address these two questions, the SADEA-II framework is proposed, which is shown as follows. The flow diagram is shown in Fig. 1.

Stage 1: **Pool Generation**: Construct the pool of candidate designs using SADEA with CEs. All the evaluated candidate designs are included in the pool.

Stage 2: **Data Mining**: Generate the initial population for FE-based optimization by clustering of the candidate design pool from Stage 1, self-development using FEs and performing FEs to some optimal solutions in terms of CEs.

Stage 3: **FE-based optimization**: Carry out a SADEA-based optimization; however, enhanced by a surrogate-model-assisted local search starting from the initial population obtained in Stage 2 using FEs.

Compared to most multi-fidelity optimization frameworks, two distinct differences of the SADEA-II framework are: (1) The initial candidate solutions for

FE-based search are generated based on a data mining process (it tries to generate a good starting population in terms of FE from a data pool that exhibits a distorted landscape but is worth to be studied from Stage 1), instead of a set of selected "promising" candidates based on CEs. (2) Both global and local search are conducted in FE-based search, instead of only using local exploitation. Stage 2 and Stage 3 are introduced as follows.

3.2. Stage 2: Data Mining

The goal of the data mining stage is to provide an initial population as close to the true optimal region as possible to support FE-based optimization (stage 3) using least number of high-fidelity EM simulations. The key challenge is that the true qualities of the candidate design pool are not known beforehand and the number of FEs which can be used is limited. To address this, we design the data mining process with two phases: initial seed population P_s generation and self-development. The former phase aims to extract fair candidate designs in terms of FE from the pool, while the later phase aims to generate the initial population of Stage 3 based on the extracted seed population. Remind that even when the response feature is largely misrepresented, there are still often a small number of optimal designs in terms of CE which are not bad in terms of FE (Section 3.1). Verifying some of these good designs in terms of CE can help both of the above phases, which is called verification and is included in both phases.

The procedure and the flow diagram (Fig. 2) are provided first and clarifications are then followed. Besides the GP modelling and the DE operators in Section 2, some operators used in this stage are defined in Table 1. In the remainder of this paper, $f_c(x)$ represents the performance value in terms of CE, while $f_f(x)$ represents the performance value in terms of FE.

Input: The candidate design pool D_p from Stage 1

Output: The initial population for Stage 3; Training data set with FE values

Table 1: Operators in Stage 2

operator name	input	output	purpose
$Divide$	(1) a design set, (2) a performance set, (3) the number of groups	groups of designs	divide a design set into a defined number of groups (G) evenly based on the performance value (f). The solutions gathered in the $i^{th}(i = 1, 2, \ldots, G)$ group correspond to the performance values in the range $[min(f) + \frac{i-1}{G}(max(f) - min(f)), min(f) + \frac{i}{G}(max(f) - min(f))]$
$iKmeans$	(1) a design set, (2) the number of clusters	(1) the clustered design set, (2) the centroid of each cluster	use the intelligent Kmeans method [27] to cluster a design set (Euclidean distance) into a defined number of clusters
$NearestPoint$	(1) a design set, (2) a reference design	the selected design	select a design from the design set that is the closest to the reference design (Euclidean distance)
FEV	(a) design(s)	the performance of the design(s)	obtain the performance of (a) design(s) in terms of FE(s)
$Elite$	(1) a design set, (2) a performance set, (3) the number of designs in the elite design set	the elite design set	obtain an elite design set which is composed of a defined number of top ranked designs (based on performance)
$Refine$	(1) a design set, (2) a performance set, (3) a threshold	the refined design set	obtain a refined design set by removing designs whose performance values are worse than a defined threshold

Elements in the design set and the performance set are in one to one correspondence.

Step 1: Use the *Divide* operator to divide D_p into G groups based on $f_c(x)$ values.

Step 2: Use the *iKmeans* operator to split each group of designs from Step 1 into 2 clusters to obtain in total $2 \times G$ clusters and $2 \times G$ centroids. Use the $NearestPoint$ operator to obtain $2 \times G$ designs that are the closest to the above centroids. Use the FEV operator to obtain the $f_f(x)$ values of these designs. Remove the evaluated designs from D_p.

Step 3: Select a group D_s (among all the G groups from Step 1) in which the current best $f_f(x)$ (from Step 2) locates. Use the *iKmeans* operator to split D_s into $0.2 \times \lambda$ clusters (λ is the population size, see Section 2.3) and get $0.2 \times \lambda$ centroids. Use the $NearestPoint$ operator to obtain $0.2 \times \lambda$ designs from D_p that are the closest to the above centroids, which form the seed population A: P_{sA}. Use the FEV operator to obtain the $f_f(x)$ values of P_{sA}. Remove the evaluated designs from D_p.

Step 4: Setting D_p as the design set, $f_c(x)$ values as the performance set, $0.5 \times \lambda - ||P_{sA}||$ as the number of designs in the elite design set, use the $Elite$ operator to obtain the elite set, which forms the preliminary seed population B. Use the FEV operator to obtain the $f_f(x)$ values of population B. Setting the population B as the design set, its $f_f(x)$ values as the performance set, 0.75 quartile of P_{sA} as the threshold, use the $Refine$ operator to form P_{sB}. Remove the evaluated designs from D_p.

Step 5: Form the seed population P_s by combining P_{sA} and P_{sB}.

Step 6: Apply DE/best/1 (7) mutation strategy and binomial crossover (8) to P_s to generate $||P_s||$ new child solutions. Use all the solutions in P_s as the training data points to construct a blind GP model and estimate the child solutions. Use the FEV operator to obtain the $f_f(x)$ value of the estimated best child solution. Repeat the above process until $0.1 \times \lambda$ new candidate designs are generated.

Step 7: Setting D_p as the design set, $f_c(x)$ values as the performance set, $0.1 \times \lambda$ as the number of designs in the elite design set, use the *Elite* operator to obtain the elite set. Use the *FEV* operator to obtain the $f_f(x)$ values of the elite set. Remove the evaluated designs from D_p.

Step 8: Combine candidate designs and their $f_f(x)$ values from Step 6 and 7. Use the *Elite* operator to select the top $0.1 \times \lambda$ candidate designs (based on $f_f(x)$ value) and add them to P_s.

Step 9: If $||P_s|| = \lambda$, output P_s (the initial population for Stage 3) and $f_f(x)$ values of candidate designs in P_s. Output all the evaluated candidate designs and their $f_f(x)$ values as the initial training data points for Stage 3. Otherwise; go back to Step 6.

Steps 1-5 of the above procedure realize extraction of fair candidate designs in terms of FE to form the initial seed population P_s. Due to the limitation of the number of FEs, clustering technique, which is essential for selecting representative(s) from a group of candidate designs, are used. However, the candidate design pool from Stage 1 should not be directly clustered. In Stage 1, the search gradually transforms from emphasizing global exploration to emphasizing local exploitation. Hence, the solutions visited earlier exhibit much larger distances between each other than those visited later. When directly clustering the candidate design pool (based on the distance), the earlier visited solutions will dominate the clustering; however, one cannot expect that many promising subregions will be identified using the candidate designs visited in early exploration. Our method to address this is to split the candidate design pool into groups (with the distances between candidates kept on the same level within each group); subsequently, the clustering is carried out in each group separately. $f_c(x)$ value is selected as a reference to approximately reflect different phases of the search. Candidate designs visited in each phase are gathered to a group.

For each group, the design solution clustering is realized by means of the iKmeans clustering, which is to prevent the uncertainty of the standard Kmeans

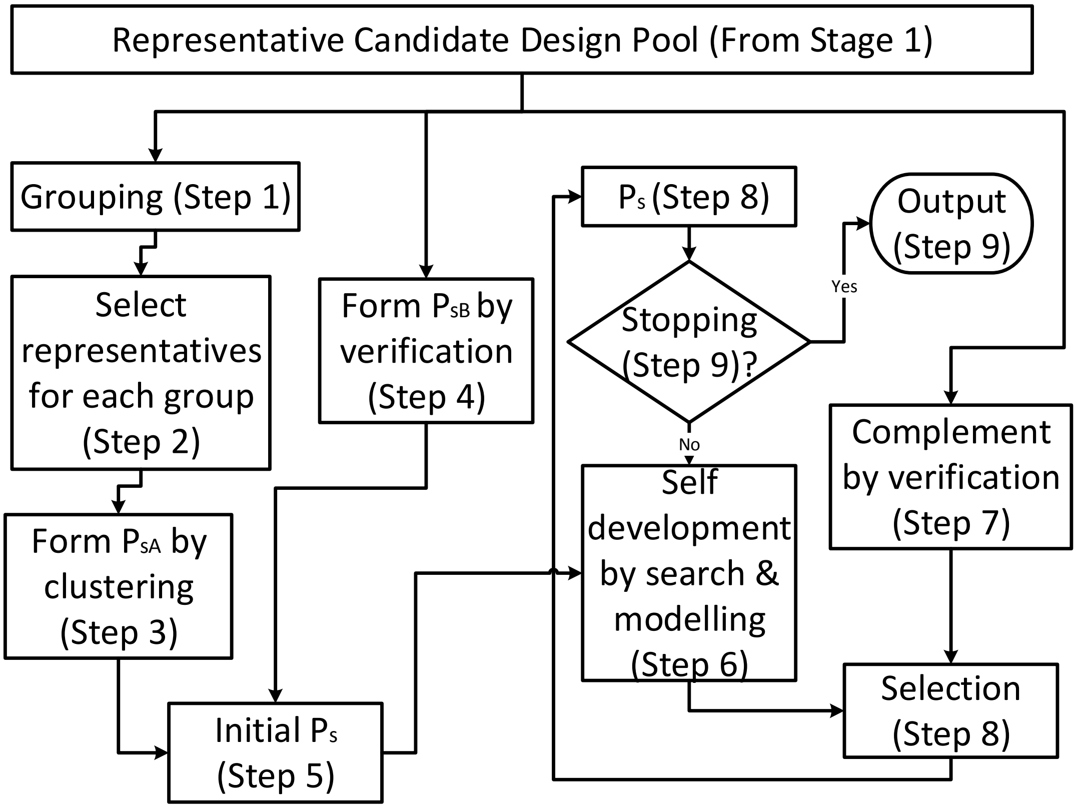
Representative Candidate Design Pool (From Stage 1)
Grouping (Step 1)
Select representatives for each group (Step 2)
Form P_{sA} by clustering (Step 3)
Form P_{sB} by verification (Step 4)
Initial P_s (Step 5)
P_s (Step 8)
Stopping (Step 9)?
Yes
No
Output (Step 9)
Self development by search & modelling (Step 6)
Complement by verification (Step 7)
Selection (Step 8)

Figure 2: Flow diagram of the data mining process

clustering [27]. Furthermore, the members of the population P_s are also selected from truly optimal candidates through verification of the “optimal” solutions using the CE-based ranking.

Steps 6-9 implement the self-development process. Note that both the number and the quality of the extracted designs in the initial seed population are not expected to be sufficient, because the good enough designs in terms of FE may not exist in the pool and a portion of them may not be found due to limited allocated FEs. Therefore, instead of being directly used as the initial population of Stage 3, a self-development process using FEs based on them is necessary. Step 6 generates new promising candidates based on P_s, which is *not* affected by the discrepancy between the coarse and the fine models. The DE/best/1 strategy is used here with the main objective being to yield a good solution at a low computational cost. Similarly to the initialization of P_s, verification of “optimal” solutions based on CE ranking is also used to update the P_s (Steps 7-8) for the next round of the self-development process.

Combination of self-development and verification is especially useful for antenna optimization. The maximum value of a response (e.g., reflection) over certain frequency band of interest is a common way to evaluate antenna performance, such as $max|S_{11}|$ from 3.1 GHz to 10.6 GHz (UWB range). However, such minimax-type of design specifications are analytically less tractable: while single frequency response is normally a smooth function of antenna geometry parameters, the minimax objective is continuous but not differentiable. As a result, a larger number of training data points (i.e., FEs) are necessary to construct a good quality mathematical approximation model (in particular, blind GP model) if only depending on Step 6. Therefore, verification of solutions “optimality” using the CE-based ranking is generally recommended because of the observation in Section 3.1. Although the success rate may be low due to the model discrepancy, a few decent candidate designs can be very helpful for improving the quality of the intermediate population P_s.

Using or not using verification steps are compared using six real-world antenna optimization problems, four of which have coarse EM models of intentionally low fidelity, leading to much discrepancy (The coarse models of the other two are analytical models, whose fidelity cannot be changed). The above data mining stage shows clear advantages on efficiency for all the test problems. Considering the extreme case when there is no fair design among optimal designs in terms of CE, the discarding of low quality design in Step 4 prevents the data mining to be failed.

There are a few fixed numbers in the process, such as using 2 clusters for the initial test in Step 2, generating 10% of the population size in each round of self-development (step 6-8). They are empirical settings. Once set, they never change and experimental results on all real-world antenna test problems show success.

3.3. Stage 3: SADEA Enhanced by Local Search

Stage 3 yields the final optimal design using computationally expensive FEs. Clearly, SADEA can be directly applied, but Stage 3 aims to further reduce the number of FEs compared to SADEA taking advantage of the initial population P_s. Compared to Stage 1, the candidate designs in P_s have good quality and it is reasonable to assume that P_s is in a largely reduced subregion of the search space (which is also verified by pilot experiments). Hence, in many cases, a surrogate-model-assisted local search with reduced exploration ability may improve the design quality using fewer FEs than that required by SADEA. On the other hand, because most landscapes of antenna are multimodal [28], the solution may be trapped in local optima when only performing local search. To balance the global search ability and fast convergence, a surrogate-model-assisted local search is used to assist SADEA. The surrogate-model-assisted local search method of choice is ORBIT [29]. ORBIT is a very successful radial basis function-assisted trust-region method. More details can be found in [29]. Clearly, other successful surrogate-model-assisted local search methods can also be adopted.

Stage 3 works as follows:

Step 1: Calculate the Euclidean distances between each individual in the initial population (provided by Stage 2) and the centroid of it. Set the average and the largest distance values as the initial radius and the maximum radius of the trust-region, respectively.

Step 2: Carry out ORBIT starting from the current best design (in terms of FE) of the initial population using N_{orbit} FEs and / or if the RBF gradient is smaller than a given tolerance. Update the current best design. Add all the FE results to the training data set.

Step 3: Carry out Steps 3-7 in Stage 1 (Section 2.3) using k FEs. Update the current best design. Add all the FE results to the training data set. Go back to Step 1 until the stopping criterion (e.g., predetermined computational budget setting) is met.

Note that a surrogate-model-assisted local search method has (to some extent) ability to escape from the local optima because surrogate modeling itself smoothens the landscape. To promote this ability, a reasonably large initial trust-region radius is used, as is shown in Step 1.

3.4. Parameter Setting

SADEA and ORBIT are the components of SADEA-II. Parameter setting rules for SADEA and ORBIT are investigated and those parameters are shown to be insensitive by experimental verifications. More details are provided in [11, 29]. For parameters with a suggested fixed value, we follow [11] and [29]. For parameters with suggested ranges, we use $\alpha = 50$, $\lambda = 50$, $\tau = 8 \times d$ for all test problems. The new parameters introduced in SADEA-II are shown in Table 2.

The recommended setting rules are as follows:

- N_{ce}: Clearly, this number does not need to be precise. We suggest: (1) Use $25 \times d$ CEs as the minimum value. (2) Subsequently, stop Stage 1 when there is no improvement with respect to the best fitness value or

Table 2: New parameters in SADEA-II

N_{ce}	the number of CEs used in Stage 1
G	the number of groups the pool is splitted into based on $f_c(x)$ in Stage 2
k	the threshold number of FEs to trigger ORBIT in Stage 3
N_{orbit}	the number of FEs assigned to each ORBIT run in Stage 3

only slight improvement is recorded after 200 consecutive CEs. According to the results of various test cases, this setting is suitable to build a good candidate design pool. Given that CEs are much cheaper than FEs, this process is also not expensive for SADEA (although it is often still too expensive for standard EAs).

- G: The value of G should be neither too small (otherwise the distances in each group are still not on the same level) nor too large (otherwise FEs will be wasted in later steps). We suggest to set it between [4,6].
- k: The setting of k depends on the computational budget. When the number of FEs is at the level of 100 to 300 (or more), which is typical in practice, k can be set to 50. In case the computational budged only allows a few FEs, k can be set quite small to trade off the global optimization capability for efficiency.
- N_{orbit}: N_{orbit} is recommended to be within the range [20,40] according to empirical test on mathematical benchmark problems and real-world antenna problems. Note that ORBIT may terminate before using N_{orbit} simulations when the tolerance is less than the threshold 1e-4.

It can be seen that the above parameters either do not need to be precise or the suggested ranges are narrow. This ensures that the parameter setting is not a practical problem. In the experiments, we use $G = 5$, $k = 50$ and $N_{orbit} = 40$. Note that the same parameters are used throughout all test problems to verify the robustness of SADEA-II.

3.5. Discussions on selecting surrogate models

In SADEA-II, two kinds of surrogate models, which are the GP model and the RBF model, are used. An interesting question is that can other kinds of surrogate modelling methods, such as Artificial Neural Network, Support Vector Machine, be used in the SADEA-II framework. We do not recommend using other surrogate modelling methods. The reasons are as follows: (1) Stage 1 implements SADEA. In SADEA, the LCB prescreening is important to make SADEA jump out of local optima and the LCB presceening is only applicable to GP modelling. (2) In Stage 2, the number of available training data points is often insufficient. GP modelling has advantages on tractability for such problems because of its sound mathematical foundation. [30] provides more details on comparisons with Artificial Neural Network. (3) ORBIT is used in Stage 3. The method to select the next point for evaluation in ORBIT relies on a property of the RBF model. [29] provides more details.

4. Experimental Results and Comparisons

SADEA-II has been tested by six real-world antennas and all of them showed success. To cover as much information as possible, two very different antennas from the view of multi-fidelity optimization and design challenges are used in this section to demonstrate the operation and performance of SADEA-II. The test cases include: a linear microstrip antenna array (LMA) and a Yagi-Uda antenna (YUA). The coarse model for the first example is an analytical model, whereas the coarse model for the second example is a coarsely-discretized EM-simulation model. The fine models for both test problems are high-fidelity EM models. For the sake of SADEA-II verification, the fidelity (discretization level) of the coarsely-discretized EM models is intentionally selected so that some important response features of the fine model are misrepresented to a large extent.

Because of using of superposition model to replace the actual expensive EM simulations in the LMA example, it was possible to execute a comparison between SADEA and the standard DE based on 30 runs of each algorithm. This also enables us to verify the robustness of SADEA-II for real-world antenna problems. On the other hand, because of the high CPU cost of individual EM simulations, it is difficult to run DE or PSO for the YUA example within a reasonable timeframe. Hence, SADEA is used as the reference method based on two runs. Note that in [11], the optimization capability and robustness of SADEA is verified by comparing to DE and PSO using multiple runs with less expensive antenna optimization problems. Because advantages of SADEA compared to popular EAs and some surrogate model assisted EAs are shown in [11], such comparisons will not be repeated when using SADEA as the reference method. SADEA-II and SADEA share some of the parameters (Section 3). For DE, a population size of 50 is used, which is a common setting [23], whereas the other parameters ($F = 0.8$ and $CR = 0.8$) are the same as in SADEA-II and SADEA.

The ranges of the design variables are set by the experience of the designer, which are reasonably wide and without any case specific investigation. These examples are run on a PC with 2.7GHz Xeon CPU with 6GB RAM. The time consumptions reported refer to the clock time.

For antenna examples, the discrepancy between the coarse and fine models is difficult to be quantified analytically. To study the performance of SADEA-II for even more complex problems with analytically quantized discrepancy, a mathematical benchmark problem is constructed to mimic the low-fidelity EM models of various levels with increasing difficulty, which is described in Section 4.3.

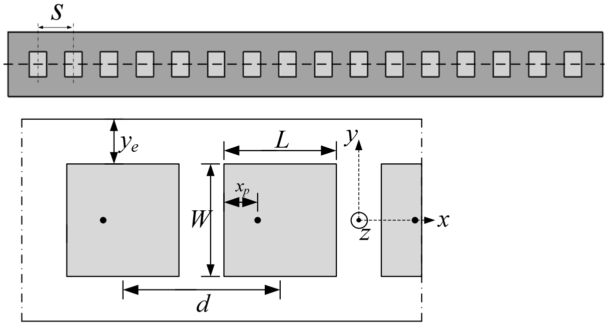

Figure 3: Geometry of 16-variable microstrip patch antenna array

4.1. Example 1: Linear Microstrip Antenna Array

The first example is a 10 GHz 16-element microstrip patch antenna array shown in Fig. 3, implemented on a finite 1.575-mm-thick Rogers RT5880 dielectric substrate ($\varepsilon = 2.2$), which extends laterally beyond the patch edges by $x_e = 18.4$mm in the x-direction and $y_e = 9.2$mm in the y-direction. The patches have dimensions $L = W = 9.2$mm and the spacing between their centers is $dc = 15$mm. Each patch is independently fed by a wire probe, situated at a distance $x_p = 6.3$mm from the leftmost patch edge. There are 16 design variables, which are excitation amplitudes $a_k, k = 1, 2, \dots, 16$ with a range of $[0, 1]^{16}$. The objective is minimization of the side lobes assuming ±8 degree of the main beam:

$$\text{minimize} \quad SLL \tag{9}$$

where SLL is the sidelobe level, i.e., the maximum relative power for the angles 0 to 82 degrees and 98 to 180 degrees.

The coarse model is an analytical array factor model assuming ideal isotropic radiators, for which each calculation costs about 5×10^{-3}s. For the fine EM model, hexahedral mesh is used and the maximum cell density is 40 cells per wavelength and total number of cells is about 900,000. The simulation time is about 30 minutes. A superposition model is built as superposition of indi-

Table 3: Statistics of the best function values (dB) obtained by SADEA-II (500 FEs), SADEA (1000 FEs), standard DE (30,000 FEs)

Method	best	worst	average	std
SADEA-II	-22.87	-21.61	-22.45	0.27
SADEA	-22.24	-19.82	-21.61	0.59
Standard DE	-23.14	-23.06	-23.12	0.02

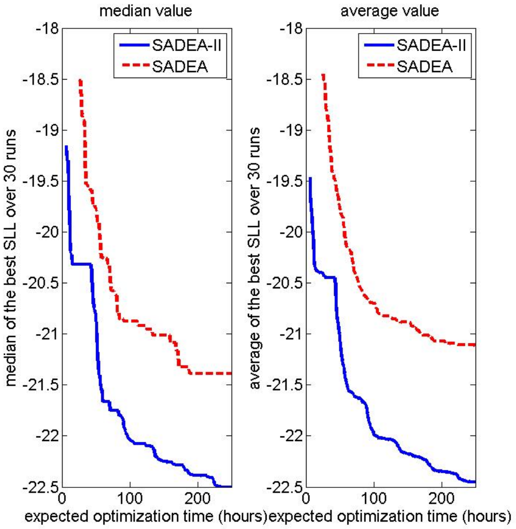

Figure 4: The convergence trends of SADEA-II and SADEA (LMA)

465 vidually simulated far fields of all array elements. Each element is simulated within the array in order to take into account electromagnetic couplings with all other elements. Hence, we can use the computationally cheap superposition model to replace EM simulation and compare SADEA-II, SADEA and DE in a statistical way. For SADEA-II, a total of 500 FEs are used. The statistics of 470 30 runs are shown in Table 3. Fig. 4 shows the convergence trend of SADEA-II and SADEA using 500 FEs. The response of the best design is shown in Fig. 5.

The following conclusions can be drawn for this example using the data gathered in Table 3: (1) SADEA-II exhibits good optimization quality (i.e., the

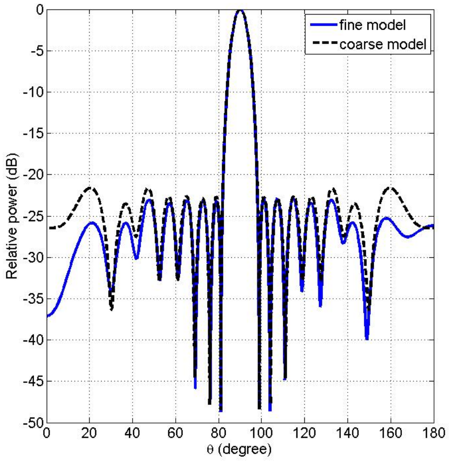

Figure 5: Response of the best solution obtained by SADEA-II (LMA)

quality of the final design), which is better than SADEA and slightly worse than but comparable to DE. (2) SADEA-II exhibits good robustness. (3) From Fig. 4, it can be seen that when using 500 FEs, SADEA-II shows much faster convergence rate than SADEA. To obtain the objective function value of SADEA-II using 500 FEs, the standard DE needs 6300 FEs. Hence, more than an order of speed improvement is obtained by SADEA-II compared to the standard DE.

To investigate the discrepancy between the coarse and fine models and the function of the data mining stage, the best candidate design obtained by Stage 1 and the final optimal design from Stage 3 in each run are compared. Results showed that among the 16 design variables over 30 runs, the maximum average difference between them is 10% of the search range, and the maximum difference spreads from 17.7% to 48.3% to some design variables. This shows that the basic assumption in Section 1 is not valid and the true optimum will be lost if following the traditional multi-fidelity optimization method.

4.2. Example 2: Yagi-Uda Antenna

The second example is a planar YUA [31] implemented on Rogers RT6010 ($\varepsilon_r = 10.2$, $tan\delta = 0.0023$, $h = 0.635$mm). The structure comprises a driven

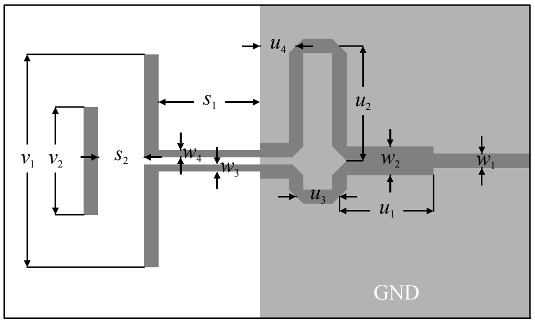

Figure 6: Geometry of 8-variable, planar Yagi-Uda antenna

Table 4: Ranges of the design variables (all sizes in mm) for antenna optimization (YUA)

$Variables$	s_1	s_2	v_1	v_2	u_1	u_2	u_3	u_4
Lower bound	3	1	5	2	2	2	1	1
Upper bound	7	6	12	12	6	6	5	5

element and one director fed by 50Ω microstrip-to-slot line balun based on a power divider (Fig. 6). Design variables are $x = [s_1, s_2, v_1, v_2, u_1, u_2, u_3, u_4]$. Their ranges are shown in Table 4. Other parameters are fixed: $w_1 = w_3 = w_4 = 0.6$, $w_2 = 1.2, u_5 = 1.5, s_3 = 3, v_3 = 17.5$ (all in mm). The design objective is to minimize the maximum reflection coefficient and the constraint is that the average gain should not be smaller than 6 ($15.6dB$) in the 10 to 11 GHz frequency range. The objective function is as follows:

$$\begin{aligned} &\text{minimize} \quad max|S_{11}| \\ &s.t. \quad mean(G) \geq 6 \end{aligned} \tag{10}$$

For both coarse and fine EM models, hexahedral mesh is used. For the coarse model, the maximum cell density is 15 cells per wavelength and total number of cells is 85,680. For the fine model, the maximum cell density is 45 cells per wavelength and total number of cells is 1,512,000. The simulation time of the coarse model and the fine model are about 2 minutes and 40 minutes, respectively. The fidelity of the coarse model is selected to introduce considerable discrepancy in reflection response. Furthermore, the simulated gain of the coarse model is lower than that of the fine model. Consequently, the optimal solutions obtained using CEs are infeasible in terms of FE. A total of 110 FEs are used for SADEA-II. For this constrained optimization problem, a penalty function method is used. The penalty coefficient is set to 100. Note that the surrogate models of the two performances (i.e., those concerning reflection and gain) are constructed separately, rather than directly modeling the penalized function values. The purpose is to avoid modeling an aggregated objective function (i.e., the main objective + the penalty term, which is very not smooth), which would reduce the performance of blind GP modeling.

Fig. 7 shows the convergence trend of SADEA-II and SADEA. In the two runs of SADEA-II (using 110 FEs), one result is $max|S_{11}| = -22.43dB$, $mean(gain) = 6.00$, and the other result is $max|S_{11}| = -21.96dB$, $mean(gain) = 6.03$. The result of SADEA using 450 FEs is $max|S_{11}| = -22.24dB$, $mean(gain) = 6.05$. It can be seen that SADEA-II is much faster than SADEA and yields a better final design even when a feasible candidate design cannot be found in the candidate design selection pool, verifying the capability of the data mining stage. The response of the optimized YUA is shown in Fig. 8.

The best candidate design obtained by Stage 1 and the final optimal design from Stage 3 are also compared. Results showed that among the 9 design variables, the maximum average difference between them is 16.9% of the search range. Again, directly performing a local search from the optimal point of the coarse model is hard to lead to the true optimum. SADEA-II, in contrast, successfully handles the discrepancy.

4.3. Benchmark Problem Tests

To test SADEA-II with analytically quantified discrepancy between the coarse and the fine models, we construct a mathematical benchmark problem-based test instance. The basic function is the 20-dimensional Ackley function [32] (see Appendix). The Ackley function has a nearly flat outer region with a

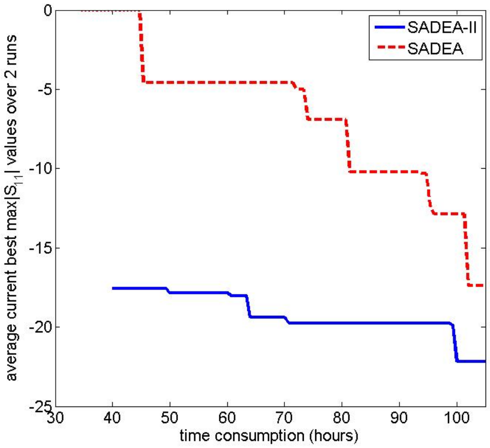

Figure 7: The convergence trends of SADEA-II and SADEA (YUA)

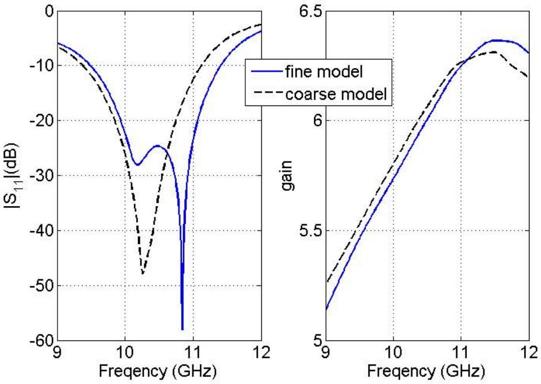

Figure 8: Response of the solution obtained by SADEA-II (YUA): $max|S_{11}| = -22.43dB$, $mean(gain) = 6.00$

Table 5: Mathematical Benchmark Problems with Increasing Discrepancy

Problem	missing peaks	spatial shift
P1	5%	0% of the search ranges
P2	10%	5% of the search ranges
P3	15%	10% of the search ranges

narrow peak, which is similar to some landscapes obtained by EM simulation. On the other hand, the landscape of the Ackley function is highly multimodal (numerous local optima) which is often much more complex than antenna problems. When the optimum is shifted, the numerous local optima bring more difficulties for the data mining method to locate the truly optimal area.

In the constructed test problems, the Ackley function is served as the coarse model. [33] provides an effective method to construct test problems for multi-fidelity optimization, which analytically quantifies the discrepancy: $f_c(x)$ and $f_f(x)$ are the coarse and fine model evaluations, respectively.

$$f_f(x) = f_c(s_f \times (x - s_s)) \tag{11}$$

where $f_c(x)$ (also $f_f(x)$) is a periodic function, and there exist minimal and maximal values in each period. s_f mimics the loss of the peaks of $f_c(x)$. In our problem, this is similar to missing of some resonances. For example, $f_f(x) = cos(s_f \times (x - s_s))$ and $f_c(x) = cos(x)$. When s_f is set to 1.15, it indicates that 15% of the peaks are lost by $f_c(x)$. s_s shifts the positions of the optimal points. In our problem, this is similar to the response shifts in frequency. Based on this method, three problems with increasing difficulties are constructed, which is shown by Table 5. The formulas can be found in the Appendix. The s_s number are randomly generated according to the requirements of Table 5.

20 runs have been performed for each problem using SADEA-II. The computing budget of Stage 1 is 500 CEs and that of Stage 2 is 350 FEs. The results are shown in Fig. 9 and in Table 6. It can be seen that for the 20-dimensional

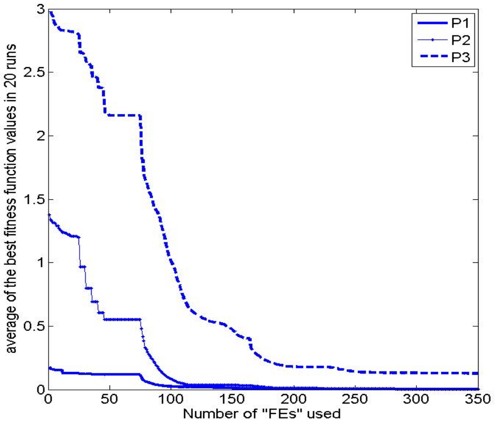

Figure 9: The convergence trends (Stage 2 and Stage 3) of SADEA-II for three mathematical benchmark problems

Table 6: Statistics of the best function values obtained by SADEA-II over 20 runs

Method	best	worst	average	std
P1	0.0005	0.0019	0.0010	0.0004
P2	0.0006	0.0091	0.0025	0.0017
P3	0.0020	1.4470	0.1274	0.3842

Ackley problem, which is often much more complex than antenna optimization problems, the discrepancy between CE and FE are successfully handled by SADEA-II. For P1 and P2, all the final results are close to the global optimum. For P3, only in 2 runs over 20 runs, SADEA-II is trapped in a reasonably good local optimum. Pilot experiments showed that by removing the verification steps using optimal solutions in terms of $f_c(x)$ and using DE mutation strategies which can provide larger population diversity (DE/best/2 [23]) in the data mining stage, even larger discrepancy than that of P3 can be well handled at the cost of slower convergence. However, given the necessity (i.e., experiments show that the function landscapes of antenna optimization are often not as complex as the 20-dimensional Ackley problem) and the high cost of EM simulations in antenna optimization problems, such method is not recommended for antenna optimization.

5. Conclusions

In this paper, the SADEA-II method has been proposed. Comprehensive experimental verification indicates that SADEA-II successfully handles various kinds and extents of discrepancy between simulation models of different fidelities without problem specific fidelity study (it is difficult to be realized for global optimization) and is scalable. Therefore, SADEA-II has addressed the main challenge for multi-fidelity optimization-based antenna design. With SADEA-II, there is a large flexibility for the coarse EM model setup, which does not need to be ad-hoc. This is because of the new three-stage multi-fidelity optimization framework and the data mining methods specially designed for antenna design optimization problems.

Thanks to the co-working of data-driven surrogate models and mutli-fidelity EM simulation models in a reliable way, SADEA-II performs as expected according to the description in Section 1 by: (1) obtaining even better result than SADEA (a state-of-the-art method for antenna optimization) using much less computing effort, addressing antenna global optimization problems with long EM simulation time (e.g., 40 minutes per high-fidelity simulation) within a practical timeframe for the first time; (2) ensuring sufficient generality for handling various low-fidelity models reliably and efficiently. Also, SADEA-II inherits the scalability of SADEA, which is able to handle 30 design variables. This is sufficient for most antenna design optimization problems. Consequently, SADEA-II is suitable for industrial use. The future work will focus on the software tools implementing SADEA-II.

Acknowledgement

The authors thank CST AG for making CST Microwave Studio available. The authors would like to thank Dr. Renato Cordeiro de Amorim, Glyndwr University, UK for valuable discussions.

Ackley Problem and the constructed multi-fidelity optimization problems

$$\begin{aligned}
&f_c(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)}\\
&f_f(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} (x_i - s_{si})^2}}\\
&-e^{\frac{1}{d}\sum_{i=1}^{d} cos(2\times s_f \times \pi x_i - s_{si})}\\
&i = 1, \ldots, d\\
&x \in [-5, 5]^{20}\\
&P1 : s_f = 1.05, s_s = 0\\
&P2 : s_f = 1.1, s_s = [0.1, 0.2, -0.2, 0.2, 0.1,\\
&-0.1, -0.1, 0, -0.1, 0, -0.2, 0, 0.2, 0, 0.1,\\
&-0.2, 0, 0.1, 0.2, 0.3]\\
&P3 : s_f = 1.15, s_s = [0.2, -0.3, 0.4, 0.4, 0.2,\\
&0.3, 0.3, -0.1, 0.2, -0.3, 0.2, -0.1, -0.2, -0.1, -0.4,\\
&0.3, 0.2, -0.2, 0.5, -0.5]
\end{aligned} \tag{.1}$$

References

[1] M. D. Gregory, Z. Bayraktar, D. H. Werner, Fast optimization of electromagnetic design problems using the covariance matrix adaptation evolu-

tionary strategy, Antennas and Propagation, IEEE Transactions on 59 (4) (2011) 1275–1285.

[2] A. Deb, J. Roy, B. Gupta, Performance comparison of differential evolution, particle swarm optimization and genetic algorithm in the design of circularly polarized microstrip antennas, Antennas and Propagation, IEEE Transactions on 62 (8) (2014) 3920–3928.

[3] Y. Sato, F. Campelo, H. Igarashi, Meander line antenna design using an adaptive genetic algorithm, IEEE Transactions on Magnetics 49 (5) (2013) 1889–1892.

[4] P. Di Barba, F. Dughiero, M. Forzan, E. Sieni, Migration-corrected nsga-ii for improving multiobjective design optimization in electromagnetics, International Journal of Applied Electromagnetics and Mechanics (Preprint) 1–12.

[5] S. K. Goudos, K. A. Gotsis, K. Siakavara, E. E. Vafiadis, J. N. Sahalos, A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution, Antennas and Propagation, IEEE Transactions on 61 (6) (2013) 3042–3052.

[6] M. Ohira, A. Miura, M. Taromaru, M. Ueba, Efficient gain optimization techniques for azimuth beam/null steering of inverted-f multiport parasitic array radiator (mupar) antenna, Antennas and Propagation, IEEE Transactions on 60 (3) (2012) 1352–1361.

[7] Y. Sato, F. Campelo, H. Igarashi, Fast shape optimization of antennas using model order reduction, IEEE Transactions on Magnetics 51 (3) (2015) 1–4.

[8] S. Koziel, S. Ogurtsov, Antenna Design by Simulation-Driven Optimization, Springer, 2014.

[9] E. S. Siah, M. Sasena, J. L. Volakis, P. Y. Papalambros, R. W. Wiese, Fast parameter optimization of large-scale electromagnetic objects using di-

rect with kriging metamodeling, Microwave Theory and Techniques, IEEE Transactions on 52 (1) (2004) 276–285.

[10] M. John, M. Ammann, Antenna optimization with a computationally efficient multiobjective evolutionary algorithm, Antennas and Propagation, IEEE Transactions on (2009) 24.

[11] B. Liu, H. Aliakbarian, Z. Ma, G. Vandenbosch, G. Gielen, P. Excell, An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques, Antennas and Propagation, IEEE Transactions on 62 (1) (2014) 7–18.

[12] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (4) (1998) 455–492.

[13] J. Knowles, Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, Evolutionary Computation, IEEE Transactions on 10 (1) (2006) 50–66.

[14] B. Liu, D. Zhao, P. Reynaert, G. G. Gielen, Synthesis of integrated passive components for high-frequency rf ics based on evolutionary computation and machine learning techniques, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 30 (10) (2011) 1458–1468.

[15] S. Koziel, S. Ogurtsov, Model management for cost-efficient surrogate-based optimisation of antennas using variable-fidelity electromagnetic simulations, IET Microwaves, Antennas & Propagation 6 (15) (2012) 1643–1650.

[16] A. I. Forrester, A. Sóbester, A. J. Keane, Multi-fidelity optimization via surrogate modelling, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 463, The Royal Society, 2007, pp. 3251–3269.

[17] M. C. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (1) (2000) 1–13.

[18] S. Koziel, A. Bekasiewicz, I. Couckuyt, T. Dhaene, Efficient multi-objective simulation-driven antenna design using co-kriging, Antennas and Propagation, IEEE Transactions on 62 (11) (2014) 5900–5905.

[19] T. Santner, B. Williams, W. Notz, The design and analysis of computer experiments, Springer, 2003.

[20] V. R. Joseph, Y. Hung, A. Sudjianto, Blind kriging: A new method for developing metamodels, Journal of mechanical design 130 (2008) 031102.

[21] I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Blind kriging: Implementation and performance analysis, Advances in Engineering Software 49 (2012) 1–13.

[22] M. Emmerich, K. Giannakoglou, B. Naujoks, Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels, Evolutionary Computation, IEEE Transactions on 10 (4) (2006) 421–439.

[23] K. Price, R. Storn, J. Lampinen, Differential evolution: a practical approach to global optimization, Springer-Verlag New York Inc, 2005.

[24] B. Liu, Q. Chen, Q. Zhang, G. Gielen, V. Grout, Behavioral study of the surrogate model-aware evolutionary search framework, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 715–722.

[25] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics (1987) 143–151.

[26] B. Liu, Q. Zhang, G. Gielen, A gaussian process surrogate model assisted evolutionary algorith for medium scale expensive optimization problems, IEEE Transactions on Evolutionary Computation 18 (2) (2014) 180–192.

[27] L. Rutkowski, Clustering for data mining: A data recovery approach, Psychometrika 72 (1) (2007) 109–110.

[28] P. Rocca, G. Oliveri, A. Massa, Differential evolution as applied to electromagnetics, Antennas and Propagation Magazine, IEEE 53 (1) (2011) 38–49.

[29] S. M. Wild, R. G. Regis, C. A. Shoemaker, Orbit: Optimization by radial basis function interpolation in trust-regions, SIAM Journal on Scientific Computing 30 (6) (2008) 3197–3219.

[30] B. Liu, G. Gielen, F. Fernández, Automated design of analog and high-frequency circuits: A computational intelligence approach.

[31] Y. Qian, W. Deal, N. Kaneda, T. Itoh, Microstrip-fed quasi-yagi antenna with broadband characteristics, Electronics Letters 34 (23) (1998) 2194–2196.

[32] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, International Journal of Mathematical Modelling and Numerical Optimisation 4 (2) (2013) 150–194.

[33] M. A. El-Beltagy, A. Keane, A comparison of various optimization algorithms on a multilevel problem, Engineering Applications of Artificial Intelligence 12 (5) (1999) 639–654.