Efficient Global Optimisation of Microwave Antennas Based on a Parallel Surrogate Model-assisted Evolutionary Algorithm

Liu, B., Akinsolu, M.O., Ali, N. and Abd-Alhameed, R.

IET Microwaves, Antennas & Propagation

Research Article

Efficient global optimisation of microwave antennas based on a parallel surrogate model-assisted evolutionary algorithm

ISSN 1751-8725
Received on 7th December 2017
Revised 7th August 2018
Accepted on 5th October 2018
doi: 10.1049/iet-map.2018.5009
www.ietdl.org

Bo Liu[1,2] ✉, *Mobayode O. Akinsolu*[1,2], *Nazar Ali*[3], *Raed Abd-Alhameed*[4]

[1]*School of Applied Science, Computing and Engineering, Wrexham Glyndwr University, Mold Road, Wrexham, UK*
[2]*Department of Electronic, Electrical and Systems Engineering, University of Birmingham, Edgbaston, Birmingham, UK*
[3]*Department of Electrical and Computer Engineering, Khalifa University, Al Zafranah, Abu Dhabi, UAE*
[4]*School of Electrical Engineering and Computer Science, University of Bradford, Richmond Road, Bradford, UK*
✉ *E-mail: liubo168@gmail.com*

Abstract: Computational efficiency is a major challenge for evolutionary algorithm (EA)-based antenna optimisation methods due to the computationally expensive electromagnetic simulations. Surrogate model-assisted EAs considerably improve the optimisation efficiency, but most of them are sequential methods, which cannot benefit from parallel simulation of multiple candidate designs for further speed improvement. To address this problem, a new method, called parallel surrogate model-assisted hybrid differential evolution for antenna optimisation (PSADEA), is proposed. The performance of PSADEA is demonstrated by a dielectric resonator antenna, a Yagi-Uda antenna, and three mathematical benchmark problems. Experimental results show high operational performance in a few hours using a normal desktop 4-core workstation. Comparisons show that PSADEA possesses significant advantages in efficiency compared to a state-of-the-art surrogate model-assisted EA for antenna optimisation, the standard parallel differential evolution algorithm, and parallel particle swarm optimisation. In addition, PSADEA also shows stronger optimisation ability compared to the above reference methods for challenging design cases.

1 Introduction

In recent years, evolutionary algorithms (EAs) are playing an important role in antenna design optimisation [1–4] because of their global optimisation capability, free of an initial design, generality, and robustness. However, the efficiency of the optimisation process is a major challenge due to the computationally expensive electromagnetic (EM) simulations. An effective efficiency improvement method is surrogate model-assisted evolutionary algorithms (SAEAs), which are employed and developed for antenna design optimisation [5, 6]. 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 used to replace potentially numerous computationally expensive EM simulations in optimisation, so as to highly improve the efficiency.

Another speed improvement method is parallel computation. Employing parallel solvers for a single antenna design simulation has already been widely applied. The computing capability (CPU, memory) of desktop workstations is continuously increasing, while the complexity of numerical techniques does not. In addition, with the development of cloud computing, more computing resources can be utilised, and carrying out multiple simultaneous simulations of several antenna designs (parallel solvers are applied to each of them) is a foreseeable tendency. Even at present, the above two-level parallel simulation is already available in some commercial software packages such as the CST Microwave Studio using multi-cores and shared memory [7].

To avoid confusion, the parallel computation in this paper refers to parallel simulations of two to four antenna candidate designs considering the capacity of a standard desktop workstation or the financial cost of using shared computing resources in a cloud. Availability of large-scale high-performance computing facilities is not considered. Parallel solvers are also not the concentration of this paper despite that they may be employed for simulating each antenna candidate design in the optimisation process. Clearly, the above parallelism has difficulty in providing substantial help for standard EA-based antenna optimisation methods, because the required population size (i.e. needed EM simulations in one iteration) is often not small (e.g. 30–50) and the computing overhead is still large even after parallelisation. Therefore, a natural idea is to introduce the above parallelisation into SAEA-based antenna optimisation methods.

Introducing parallel simulations into modern SAEAs is not trivial. In terms of surrogate model management methods, SAEAs can generally be classified into offline SAEAs and online SAEAs. Offline SAEA-based efficient antenna design optimisation methods perform a one-shot initial sampling to build the surrogate model, which is then used as the substitute of the EM simulations. In the optimisation process, the surrogate model is not updated or only updated for local refinements [8]. In this kind of methods, the EM simulations involved in the one-shot initial sampling are independent of each other and are easy to be parallelly performed. However, although more than ten variables can be handled when employing both low- and high-fidelity EM models [9, 10], it is well known that most offline SAEAs have difficulty in handling problems with more than a few variables (e.g. 4–5) for the targeted problem in this paper [11, 12].

To address the scalability problem, online SAEA-based efficient antenna design optimisation methods are introduced. Siah *et al*. [5] employ the efficient global optimisation method [13] for antenna optimisation and a new online SAEA, called surrogate model-assisted hybrid differential evolution for antenna optimisation (SADEA), is proposed in [6] by the authors. However, the terminology 'online' indicates a sequential process: the surrogate model keeps updating, often a single EM simulation is performed in each iteration, and the selected candidate design for that simulation depends on the current surrogate model built by simulation data in previous iterations [5, 6]. Therefore, parallel simulations are difficult to be used. Although Liu *et al*. [6] show up to almost an order of speed improvement compared to standard EAs for various antennas, parallel candidate antenna design simulation, which is straightforward for standard EAs, is not taken into account.

This paper aims to introduce small-scale parallel simulations into online SAEA-based antenna optimisation. A new method, called parallel surrogate model-assisted hybrid differential evolution for antenna optimisation (PSADEA), is proposed. The main innovations are the ensemble of three selected differential evolution (DE) mutation strategies and the self-adaptive employment of them, as well as a new surrogate modelling method. These innovations enable parallelisation with enhanced search capacity and surrogate model quality. The goal of PSADEA is to reduce the several weeks' optimisation time of a standard EA-based antenna optimisation method to a few hours in a small-scale parallel computing environment. An additional benefit is the stronger performance optimisation ability compared to popular standard EAs, which are often considered as the benchmark in terms of solution quality.

The remainder of the paper is organised as follows: Section 2 provides the basic techniques. Section 3 elaborates the PSADEA method. Section 4 presents two typical antenna examples and three mathematical benchmark problems to show the performance of PSADEA. The concluding remarks are presented in Section 5.

2 Basic techniques

2.1 Gaussian process surrogate modelling

The surrogate modelling method selected for PSADEA is Gaussian process (GP) [14]. Given a set of observations $x = (x^1, \dots, x^n)$ and $y = (y^1, \dots, y^n)$, GP predicts a function value $y(x)$ at some design point x by modelling $y(x)$ as a Gaussian distributed stochastic variable with mean μ and variance σ^2. If the function is continuous, the function values of two points x^i and x^j should be close if they are highly correlated. In this work, we use the Gaussian correlation function to describe the correlation between two variables:

$$\mathrm{Corr}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\sum_{l=1}^{d} \boldsymbol{\theta}_l \left| x_i^l - x_j^l \right|^{p_l}\right) \qquad (1)$$
$$\boldsymbol{\theta}_l > 0, 1 \le \boldsymbol{p}_l \le 2$$

where d is the dimension of x and $\boldsymbol{\theta}_l$ the correlation parameter which determines how fast the correlation decreases when x_i moves in the l direction. The smoothness of the function is related to $\boldsymbol{p}_l$ with respect to x^l. To determine the parameters $\boldsymbol{\theta}_l$ and $\boldsymbol{p}_l$, the likelihood function that $y = y^i$ at $x = x^i (i = 1, \dots, n)$ is maximised. The function value $y(x^*)$ at a new point x^* can be predicted as

$$\hat{y}(x^*) = \hat{\mu} + r^{\mathrm{T}} \boldsymbol{R}^{-1}(\boldsymbol{y} - I\hat{\mu}) \qquad (2)$$

where

$$\boldsymbol{R}_{i,j} = \mathrm{Corr}(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i, j = 1, 2, \dots, n \qquad (3)$$

$$r = [\mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_1), \mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_2), \dots, \mathrm{Corr}(\boldsymbol{x}^*, \boldsymbol{x}_n)] \qquad (4)$$

$$\hat{\mu} = (I^{\mathrm{T}} \boldsymbol{R}^{-1} I)^{-1} I^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{y} \qquad (5)$$

The mean square error value of the prediction uncertainty is:

$$\hat{s}^2(x^*) = \hat{\sigma}^2[I - r^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{r} + (I - \boldsymbol{r}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{r})^2 (\boldsymbol{I}^{\mathrm{T}} \boldsymbol{R}^{-1} I)^{-1}] \qquad (6)$$

where

$$\hat{\sigma}^2 = (\boldsymbol{y} - I\hat{\mu})^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{y} - I\hat{\mu}) n^{-1} \qquad (7)$$

Several prescreening methods can be used to evaluate the quality of a candidate design with respect to the predicted value in (2) and the prediction uncertainty in (6) [15]. In PSADEA, the lower confidence bound (LCB) method [16] is used. Given the predictive distribution $N(\hat{y}(x), \hat{s}^2(x))$ for $y(x)$, an LCB prescreening of $y(x)$ can be defined as:

$$y_{\mathrm{lcb}}(x) = \hat{y}(x) - \omega \hat{s}(x)$$
$$\omega \in [0, 3] \qquad (8)$$

where ω is a constant, which is often set to 2 to balance the exploration and exploitation ability [15].

The ooDACE toolbox [17] is used for implementing the GP surrogate model.

2.2 DE algorithm

The DE algorithm [18] is adopted as the search engine in PSADEA. DE outperforms many EAs for continuous optimisation problems [18] and is widely used in EM simulation-driven design optimisation.

Suppose that P is a population. Let $x = (x_1, \dots, x_d) \in R^d$ be an individual solution in P. To generate a child solution $u = (u_1, \dots, u_d)$ for x, a donor vector is first produced by mutation:

$$\boldsymbol{v}^i = \boldsymbol{x}^{r_1} + F \cdot (\boldsymbol{x}^{r_2} - \boldsymbol{x}^{r_3}) \qquad (9)$$

where $\boldsymbol{x}^{r_1}$, $\boldsymbol{x}^{r_2}$, and $\boldsymbol{x}^{r_3}$ are three mutually exclusive solutions randomly selected from P (the current population); $\boldsymbol{v}^i$ is the ith mutant vector in the population after mutation; $F \in (0, 2]$ is a control parameter, often called the scaling factor. The above mutation strategy in (9) is called DE/rand/1, which is one of the popular DE mutation strategies.

Then the following crossover operator is applied to produce the child u:

i. Randomly select a variable index $j_{\mathrm{rand}} \in \{1, \dots, d\}$.
ii. For each $j = 1$ to d, generate a uniformly distributed random number rand from (0, 1) and set:

$$\boldsymbol{u}_j = \begin{cases} \boldsymbol{v}_j, & \text{if (rand} \le \text{CR)} \mid j = j_{\mathrm{rand}} \\ \boldsymbol{x}_j, & \text{otherwise} \end{cases} \qquad (10)$$

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

In PSADEA, two other DE mutation strategies are used:

i. Mutation strategy: DE/current-to-best/1

$$\boldsymbol{v}^i = \boldsymbol{x}^i + F \cdot (\boldsymbol{x}^{\mathrm{best}} - \boldsymbol{x}^i) + F \cdot (\boldsymbol{x}^{r_1} - \boldsymbol{x}^{r_2}) \qquad (11)$$

where $\boldsymbol{x}^i$ is the ith vector in the current population and $\boldsymbol{x}^{\mathrm{best}}$ the best candidate in the current population P.

ii. Mutation strategy: DE/rand/2

$$\boldsymbol{v}^i = \boldsymbol{x}^{r_1} + F \cdot (\boldsymbol{x}^{r_2} - \boldsymbol{x}^{r_3}) + F \cdot (\boldsymbol{x}^{r_4} - \boldsymbol{x}^{r_5}) \qquad (12)$$

where $\boldsymbol{x}^{r_4}$ and $\boldsymbol{x}^{r_5}$ are two different solutions randomly selected from P and are different from $\boldsymbol{x}^{r_1}$, $\boldsymbol{x}^{r_2}$, and $\boldsymbol{x}^{r_3}$.

3 PSADEA method

3.1 SADEA method

PSADEA aims to introduce small-scale parallel simulation (simultaneous simulations of two to four candidate antenna designs) into the authors' SADEA method [6]. Hence, SADEA is firstly introduced as the foundation. SADEA shows up to an order of speed improvement compared to standard EAs for antenna optimisation (not considering parallel simulation of different candidate designs) and is scalable to around 30 design variables [6, 11]. SADEA works as follows:

Step 1: Sample α (often a small number) candidate designs from the design space $[a, b]^d$ (a and b are the lower and upper bounds of design variables, respectively; d is the number of design variables) using Latin hypercube sampling [19]. 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/best/1 mutation (13) and crossover operations [18] 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 GP surrogate model [14] using the selected candidate designs (i.e. training data points in surrogate modelling).

Step 6: Estimate the λ child solutions generated in step 4 using the GP model and the LCB method [15].

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 DE/best/1 mutation strategy used in SADEA is:

$$\boldsymbol{v}^i = \boldsymbol{x}^{\text{best}} + F \cdot (\boldsymbol{x}^{r_2} - \boldsymbol{x}^{r_3}) \tag{13}$$

3.2 Exploration ability enhancement and local surrogate modelling

To include parallel simulations in SADEA, a straightforward idea is to select the top n rather than the top 1 candidate designs (all based on prediction) in step 7 of each iteration and simulate them in parallel. Here, n depends on the computing resource and is often from 2 to 4 as mentioned in Section 1. If this framework is effective, the computational time should be reduced by approximately n times, like simulating n candidate designs simultaneously in standard EAs. However, experiments show far less than three times speed improvement when $n = 3$ using various mathematical benchmark problems [20].

Two observations are: (i) in many iterations, the predicted best candidate is the true best candidate and there are often very few other candidates with similar optimality. (ii) The best candidate may not change for every 10–20 iterations. This explains why simply selecting n top-ranked candidates for parallel simulation is not effective: most top-ranked candidates do not have the required quality. Simulating them does not help much to obtain good quality solutions and improve the surrogate model quality of the optimal region. Hence, improving the exploration ability to obtain more good quality candidates in each iteration becomes the central issue.

In SADEA, the exploration is realised by the DE/best/1 mutation strategy (13) [18] (step 4). The main consideration of using this strategy is its fast convergence property, while the cost is the reduced population diversity. Although experiments show that it can still jump out of local optima, various good quality candidates from different regions of the design space are difficult to be generated, and the parallel simulation of ‘ useful’ candidates, therefore, cannot be realised. Our idea to address this challenge is to ensemble the DE/current-to-best/1 (11), DE/rand/1 (9), and DE/rand/2 (12) strategies. Each DE mutation strategy [(9)–(12)] trades off the exploration ability (population diversity) and convergence speed to a certain extent. DE/current-to-best/1 (11) has the least population diversity consideration (but still higher than DE/best/1), but the convergence speed is the fastest, DE/rand/2 (12) emphasises the promotion of population diversity but with the slowest convergence speed. DE/rand/1 (9) is in the middle in both respects.

When employing the new DE mutation strategies, it is observed that various good quality candidate designs from different regions of the design space can be generated, but many of them are not correctly predicted by the GP model in step 5. SADEA constructs a single surrogate model in each iteration. When the diversity of the generated candidates becomes high thanks to the new DE mutation strategies, this single surrogate model does not have good performance. Therefore, in PSADEA, a local GP surrogate model is built for each child solution using τ nearest samples (based on Euclidean distance). This means that λ separate local GP models are built in each iteration. Pilot experiments show substantially improved prediction ability.

3.3 Self-adaptive DE mutation strategy selection and the PSADEA framework

Given the three selected DE mutation strategies and the surrogate modelling method, the key problem becomes how to optimally use the selected mutation strategies. Experiments show that when evenly employing the three selected DE mutation strategies, although high-quality design solutions can be generated with improved speed, the speed improvement is not as high as n times when n candidate designs are simulated in parallel. Note that DE/rand/1 and DE/rand/2 strategies have low convergence speed.

As said in Section 3.2, each mutation strategy has a different trade-off between the exploration ability (population diversity) and convergence speed. On the one hand, sufficient population diversity is needed. On the other hand, the convergence speed should be as fast as possible. However, the most appropriate trade-off depends on the particular antenna optimisation problem and is difficult to be analysed mathematically. To address this challenge, a self-adaptive method is proposed, which works as follows:

For each child population $i = 1, 2, \ldots, n$:

Step 1: If the algorithm is within the learning period (the current number of iterations is smaller than a threshold L), the rate of using DE/rand/1 (9), DE/current-to-best/1 (11), and DE/rand/2 (12) is 1/3. Otherwise, use the rates in step 5.

Step 2: Perform a roulette wheel selection [21] based on the rates to determine a DE mutation method and generate a child population C_i (λ child solutions).

Step 3: For each child solution, select τ nearest samples (based on Euclidean distance) as the training data points and construct a local GP surrogate model.

Step 4: Compare the predicted value of each solution in $\boldsymbol{C}_i$ and the current best solution (simulated value). Add the number of solutions that are better than the current best solution to N_{s} [the number of successes of (9), (11), or (12)] and add λ to N_{u} [the number of uses of (9), (11) or (12)].

Until all the n groups of child solutions are generated.

Step 5: Update the rates of using DE/rand/1 (9), DE/current-to-best/1 (11), and DE/rand/2 (12) by $N_{\text{s}}/N_{\text{u}}$. Update the number of iterations.

It can be seen that the DE mutation strategies are employed self-adaptively and the most frequently fitted will be used with a greater chance. Experiments show higher than n times time reduction compared to SADEA.

As shown by the flow diagram in Fig. 1, the PSADEA method can be constructed as follows: (i) steps 1–3 are the same as the SADEA method (Section 3.1). (ii) The above self-adaptive mutation method (new steps 1–5) is used to replace the original steps 4–6 of SADEA. (iii) Step 7 of SADEA is revised to select top n candidate designs from all of $(n \times \lambda)$ the generated child solutions and simulate them in parallel.

Compared to SADEA, there is only one additional algorithm parameter, L, in PSADEA. Empirically, L is suggested to be within [30, 50]. It is clear that L is not sensitive. It is shown that the SADEA method is not sensitive to parameter setting [6] through detailed experimental study. All the parameter setting rules are still useful in PSADEA.

4 Experimental results and comparisons

PSADEA is tested by four real-world antennas, including a hybrid dielectric resonator antenna (DRA), a planar Yagi-Uda antenna (YUA), a broadband microstrip antenna, and an ultrawideband MIMO antenna array. The reference methods include SADEA, DE

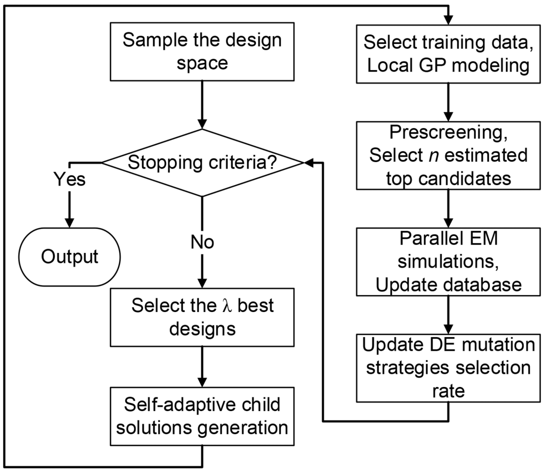
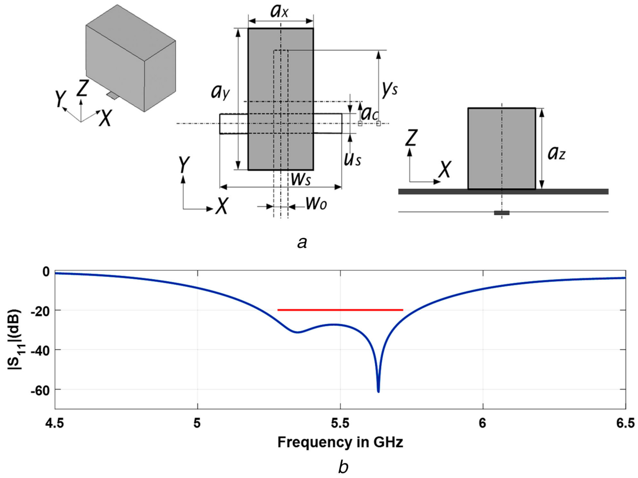

Sample the design space
Stopping criteria?
Yes
No
Output
Select the λ best designs
Self-adaptive child solutions generation
Select training data, Local GP modeling
Prescreening, Select n estimated top candidates
Parallel EM simulations, Update database
Update DE mutation strategies selection rate

Fig. 1 *Flow diagram of the PSADEA method*



a

b

Fig. 2 *Layout of the DRA and a typical optimised performance by PSADEA*
(a) The DRA, ***(b)***S_{11} response

Table 1 Ranges of the design variables (all sizes in mm) (example 1)

Variables	a_x	a_y	a_z	a_c	u_s	w_s	y_s
lower bound	6	12	6	6	0.5	4	2
upper bound	10	16	10	8	4	12	12

Table 2 Optimised $\max |S_{11}|(\text{dB})$ in the operational bandwidth using different methods for example 1

Method	Best	Worst	Mean	Std
PSADEA	−27.32	−24.76	−26.14	0.89
SADEA	−25.70	−22.05	−23.84	1.10
parallel DE	−25.03	−23.15	−24.31	1.01
parallel PSO	−22.58	−19.05	−21.34	1.99

(DE/current-to-best/1 mutation), and particle swarm optimisation (PSO) (the CST Microwave Studio PSO optimiser is used). As SADEA outperforms several popular SAEAs [6], it is used as the reference method in this paper. For all of the test cases, PSADEA successfully obtains highly optimal designs. The comparison results among different methods in terms of solution quality and efficiency are similar except that DE and PSO fail to obtain a satisfactory design for the YUA. The tests are run on a workstation with Intel 4-core i7 CPU and 24 GB RAM. Ten independent runs are carried out for PSADEA and SADEA. Standard DE and PSO are carried out for three runs because multiple runs are not affordable (a single run costs ∼10 days for some test cases).

SADEA does not support parallel simulations of different candidate designs [6]; for the parallel simulations, three candidate antenna designs are simulated in parallel for PSADEA, PSO, and DE (parallel solvers are used to each of them). This is decided by the optimal computing budget allocation algorithm of CST Microwave Studio according to the CPU and shared memory of the workstation. Note that the initial surrogate model in all test cases is constructed by 50 samples/simulations, so the starting time of PSADEA and SADEA is not 0. Based on the suggested values in [6], the following algorithm parameters are used for PSADEA and SADEA: $F = 0.8$, $\text{CR} = 0.8$, $\alpha = 50$, $\lambda = 50$, $\tau = 8 \times d$ (d is the number of design variables). L is set to 30. For DE, the same F, CR, and λ are used. For PSO, the same λ is used and default values set by the CST Microwave Studio optimiser are used for other parameters. In the following, the DRA and YUA are used to demonstrate PSADEA. The time consumption is wall clock time.

4.1 Example 1

The first example is a hybrid DRA and the layout is shown in Fig. 2*a* [22]. The DRA is modelled in CST Microwave Studio using a mesh density of 12 cells per wavelength and ∼22,000 mesh cells. It is implemented on a RO4003C substrate with a relative permittivity of 3.38, loss tangent of 0.0027, and thickness of 0.5 mm. The excitation mode is $TE^{x}_{\delta 11}$ and each simulation costs from 20 s to 1 min. As shown in Table 1 and (14), the design exploration goal is to adjust the dimensions of the DR brick and slot, length of the microstrip slab, and position of the DR relative to the slot, so that the bandwidth of the DRA is to be centred at 5.5 GHz from 5.28 to 5.72 GHz. ($a_c \le 0.5 \times a_y$) is the geometric constraint used to ensure the slot remains under the DRA in all possible cases. To make all the methods converge, the computing budget is as follows: 360 parallel EM simulations for PSADEA, 1080 sequential EM simulations for SADEA (the same number of total EM simulations with PSADEA), and 2000 parallel EM simulations each for PSO and DE.

$$\text{minimise} \quad \max |S_{11}| \quad 5.28\,\text{GHz} - 5.72\,\text{GHz} \tag{14}$$

The results are shown in Table 2. A typical response of PSADEA is shown in Fig. 2*b* with the obtained optimal design of $a_x = 9.47\,\text{mm}$, $a_y = 12.61\,\text{mm}$, $a_z = 9.92\,\text{mm}$, $a_c = 4.54\,\text{mm}$, $u_s = 3.45\,\text{mm}$, $w_s = 8.38\,\text{mm}$, and $y_s = 2.00\,\text{mm}$. The following observations can be made: (i) in all the ten runs, PSADEA provides very satisfactory results even for the worst one, using only 2.8 h on average. (ii) PSADEA shows good robustness. (iii) PSADEA shows a clear advantage over SADEA, parallel DE, and parallel PSO in terms of solution quality.

The convergence trends of all methods are shown in Fig. 3. From Fig. 3*a*, it can be seen that to obtain the average best objective function value of the SADEA algorithm (−23.84 dB at 8.5 h), PSADEA needs 2 h. Hence, PSADEA shows 4.3 times speed improvement compared to SADEA when three candidate designs are simulated in parallel. To obtain the average best objective function values of parallel DE (−24.31 dB at 16.1 h) and parallel PSO (−21.34 dB at 13.2 h), PSADEA needs 2.1 h and 1.6 h, respectively. Hence, PSADEA offers 7.7 times speed improvement over parallel DE and 8.3 times speed improvement over parallel PSO, while obtaining better results. This speed improvement comes from both parallel simulations of three candidate designs and the self-adaptive SAEA framework. Fig. 3*b* shows the convergence trends for PSADEA and SADEA when not considering parallel simulations. (The parallel simulations of PSADEA are considered as sequential.) It can be seen that to obtain the average best objective function value of SADEA (−23.84 dB at 1059 sequential EM simulations), PSADEA needs 708 sequential EM simulations. This speed improvement is only contributed by the self-adaptive SAEA framework.

4.2 Example 2

As shown in Fig. 4*a*, the second example is a planar YUA [23] modelled in CST Microwave Studio using a mesh density of 15

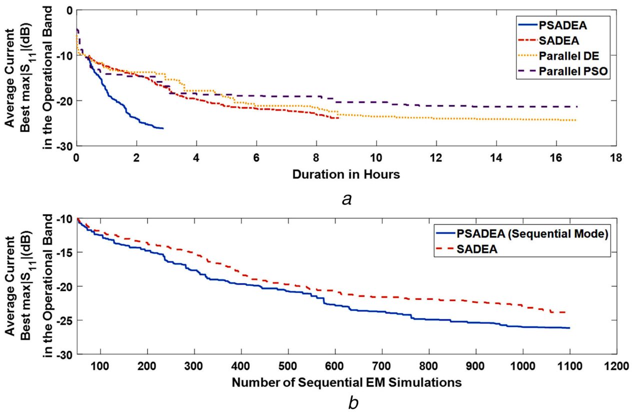


Fig. 3 *Convergence trends for example 1*
(a) Convergence trends for PSADEA and SADEA (average of ten runs), parallel DE, and parallel PSO (average of three runs), ***(b)*** Comparison of PSADEA (in sequential mode) and SADEA

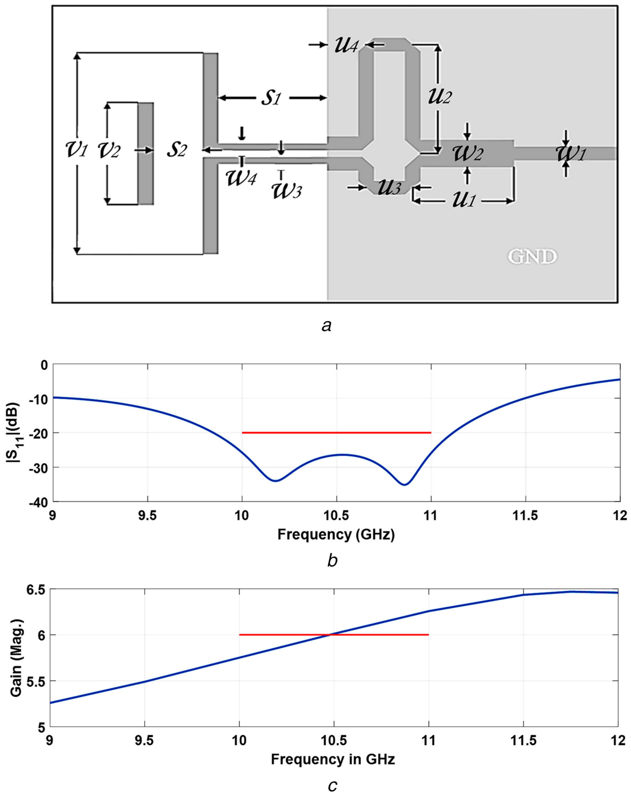


Fig. 4 *Layout of the YUA and typical optimised performances by PSADEA*
(a) The YUA, ***(b)***S_{11} response, ***(c)*** Gain response

cells per wavelength and over 86,000 mesh cells. The YUA is implemented on an RT6010 substrate with the relative permittivity of 10.2, loss tangent of 0.0033, and thickness of 0.635 mm. The driven element of the YUA is fed by a 50 Ω microstrip-to-slot balun via a power divider and each simulation costs ∼2 min. As shown in Table 3 and (15), the design exploration goal is to adjust the dimensions of the YUA for an operational bandwidth of 10–11 GHz at an average gain not <6 (7.78 dB) over the bandwidth. To make all the methods converge, the computing budget is as follows: 220 parallel EM simulations for PSADEA, 660 sequential EM simulations for SADEA (the same number of total EM simulations with PSADEA), and 1500 parallel EM simulations each for PSO and DE.

$$\begin{array}{ll} \text{minimise} & \max |S_{11}| \quad 10\,\text{GHz} - 11\,\text{GHz} \\ \text{s.t.} & \text{mean}(G) \geq 6 \end{array} \tag{15}$$

Table 3 Ranges of the design variables (all sizes in mm) (example 2)

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

Table 4 Optimised max $|S_{11}|$(dB) in the operational bandwidth using different methods for example 2

Method	Best	Worst	Mean	Std
PSADEA	−25.87	−22.82	−23.98	0.90
SADEA	−24.53	−22.52	−23.63	0.70
parallel DE	−6.27	−5.19	−5.57	0.61
parallel PSO	−15.65	−7.53	−12.26	4.23

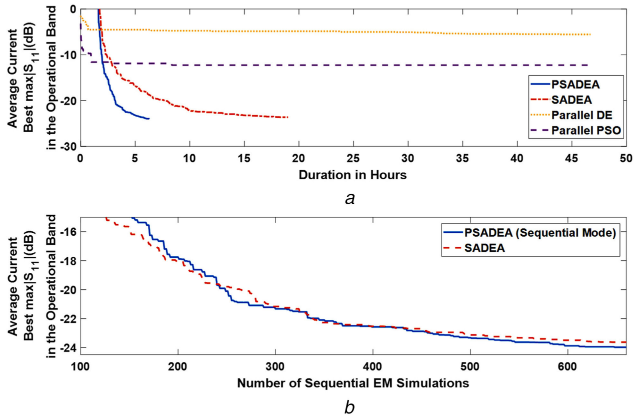


Fig. 5 *Convergence trends for example 2*
(a) Convergence trends for PSADEA and SADEA (average of ten runs), parallel DE, and parallel PSO (average of three runs), ***(b)*** Comparison of PSADEA (in sequential mode) and SADEA

The results are shown in Table 4. Typical responses of PSADEA are shown in Figs. 4*b* and *c* with the obtained optimal design of $s_1 = 5.06$ mm, $s_2 = 5.95$ mm, $v_1 = 7.51$ mm, $v_2 = 2.95$ mm, $u_1 = 4.22$ mm, $u_2 = 1.50$ mm, $u_3 = 3.80$ mm, and $u_4 = 2.78$ mm. For this example, the following observations can be made: (i) PSADEA and SADEA obtain successful results, while parallel PSO and parallel DE fail to satisfy the constraints on gain and provide poor S_{11} responses. (ii) In all the ten runs, PSADEA provides very satisfactory results even for the worst one, using only 6.3 h on average. (iii) PSADEA shows good robustness and reveals an added advantage over SADEA in terms of solution quality.

The convergence trends of all methods are shown in Fig. 5. From Fig. 5*a*, it can be seen that to obtain the average objective function value of the SADEA algorithm (−23.63 dB at 18.6 h), PSADEA needs 5.5 h. Hence, PSADEA shows over 3.4 times speed improvement than SADEA when three candidate designs are simulated in parallel. Again, when the parallel simulations of PSADEA are considered as sequential (Fig. 5*b*), to obtain the average best objective function value of SADEA (−23.63 dB at 645 sequential EM simulations), PSADEA needs 546 sequential EM simulations. This speed improvement is only contributed by the self-adaptive SAEA framework.

4.3 Benchmark problem tests

To complement the antenna examples, the performance of PSADEA is demonstrated by three mathematical benchmark problems [20, 24]. Mathematical benchmark problems are computationally cheap analytical functions. Using them, the algorithm performance can be directly evaluated because the global optimum is known, and more reliable statistical comparisons can be performed because carrying out a number of runs is computationally cheap. To compare the methods in different circumstances, a problem with a simple landscape (*F*1: the

Table 5 Statistics of the best objective function values obtained by all methods for $F1$ (over 30 runs)

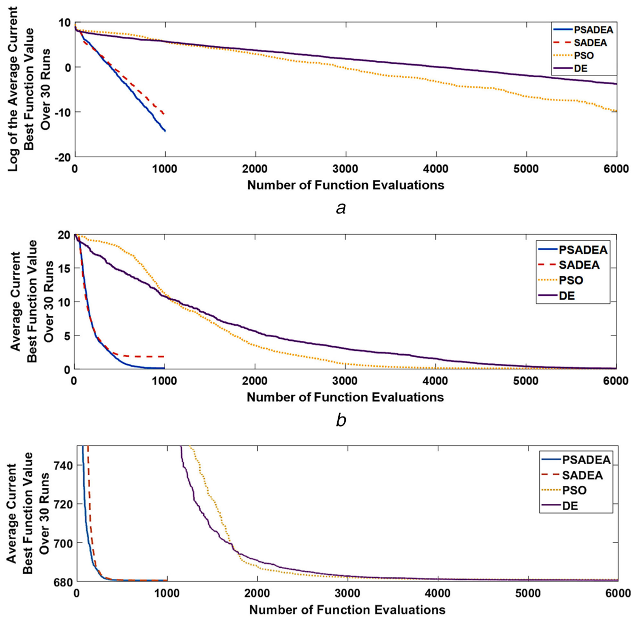
Problem	Best	Worst	Mean	Median	Std.	p-value
PSADEA	3.65×10^{-10}	2.91×10^{-6}	5.91×10^{-7}	1.87×10^{-7}	8.06×10^{-7}	N/A
SADEA	4.81×10^{-7}	9.69×10^{-5}	1.68×10^{-5}	6.51×10^{-6}	2.20×10^{-5}	1.92×10^{-6}
DE	0.0054	0.0844	0.0221	0.0182	0.0154	1.73×10^{-6}
PSO	3.86×10^{-7}	8.94×10^{-4}	5.74×10^{-5}	1.32×10^{-5}	1.64×10^{-4}	1.92×10^{-6}

Table 6 Statistics of the best objective function values obtained by all methods for $F2$ (over 30 runs)

Problem	Best	Worst	Mean	Median	Std.	p-value
PSADEA	3.48×10^{-4}	2.0134	0.1477	0.0018	0.4575	N/A
SADEA	2.44×10^{-4}	12.8622	1.8521	0.5915	3.5976	0.0073
DE	0.0440	0.1579	0.1026	0.1005	0.0322	0.0028
PSO	2.25×10^{-4}	1.6463	0.0947	8.32×10^{-4}	0.3608	0.0687

Table 7 Statistics of the best objective function values obtained by all methods for $F3$ (over 30 runs)

Problem	Best	Worst	Mean	Median	Std.	p-value
PSADEA	680.6328	680.6517	680.6406	680.6389	0.0057	N/A
SADEA	680.6350	680.6768	680.6492	680.6467	0.0104	0.0049
DE	680.6829	680.8392	680.7477	680.7442	0.0432	3.02×10^{-11}
PSO	680.6680	684.0611	681.0029	680.8468	0.6155	3.02×10^{-11}



a

b

c

Fig. 6 *Convergence trends of all methods*
(a) Problem $F1$, ***(b)*** Problem $F2$, ***(c)*** Problem $F3$

Ellipsoid function, which is a unimodal function), a problem with a complex landscape ($F2$: the Ackley function, which has numerous local optima), and a problem with tight constraints ($F3$: two out of four non-linear constraints are active at the global optimum) are used. More details on the problems can be found in the Appendix. We expect that the landscape complexity of most antenna optimisation problems is between $F1$ and $F2$. $F3$ is used to imitate antenna optimisation problems with stringent performance specifications, such as example 2 (Section 4.2). The computing budget is 1000 exact function evaluations for PSADEA and SADEA, and 6000 exact function evaluations for DE and PSO, on all problems. Parallel evaluation is *not* considered for all the methods. The statistics (over 30 independent runs) are shown in Tables 5–7.

In terms of solution quality, it can be seen that PSADEA shows excellent results for $F1$ and $F3$ compared to their known global optima of 0 and 680.63, respectively. (All methods satisfy the constraints for $F3$ at convergence.) PSADEA shows a good result for $F2$ compared to the known global optimum of 0. For $F2$, the average value of PSADEA is 0.1477, but 90% of the obtained objective function values are <0.025. This can also be seen from the median value, 0.0018.

A Wilcoxon test [25] is carried out using the optimal function values obtained by all methods (over 30 independent runs) as data samples. The null hypothesis is that the data samples of PSADEA have an equal median with the data samples of the reference methods. The confidence level is 95%. The p-values obtained from the Wilcoxon test between PSADEA and other methods on $F1$, $F2$, and $F3$ are shown in Tables 5–7, respectively. It can be seen that for $F1$ and $F3$, the null hypothesis is rejected for all the reference methods, showing PSADEA obtains statistically better performance compared to SADEA, DE, and PSO. For $F2$, the null hypothesis is rejected for SADEA and DE and it is accepted for PSO. Hence, the PSADEA result is statistically better than that of SADEA and DE, and comparable to PSO for $F2$.

In terms of efficiency, the average convergence trends of all methods when not considering parallel evaluations are shown in Figs. 6*a–c*. It can be seen that for $F1$, PSADEA shows 1.8, 10.7, and 7.4 times speed improvement over SADEA, DE, and PSO, respectively (considering the number of functions evaluations used by PSADEA to obtain the results of the reference methods). For $F2$, PSADEA shows 1.2 times speed improvement over SADEA, and 5.7 and 4.2 times speed improvement over DE and PSO, respectively. For $F3$, PSADEA shows 1.2, 12.1, and 11.0 times speed improvement over SADEA, DE, and PSO, respectively. The comparisons using mathematical benchmark problems show the same conclusions in Sections 4.1 and 4.2. Although the goal of PSADEA is to enable SADEA to benefit from parallel antenna simulations, it can be seen that the self-adaptive SAEA framework (DE mutation strategies, self-adaptive mechanism, and the new surrogate modelling method) also contributes to the efficiency excluding parallel computation as well as clearly improves the optimisation ability.

5 Conclusions

In this paper, the PSADEA method has been proposed for efficient global optimisation of microwave antennas taking advantage of parallel simulations of a few candidate antenna designs, which is a tendency in the foreseeable future. To the best of our knowledge, introducing parallel simulations into online SAEA-based antenna optimisation methods have not yet existed in the literature.

Experiments show that PSADEA is several times faster than SADEA and much more efficient than standard parallel DE and PSO. Thanks to the ensemble of the three selected DE mutation strategies and the self-adaptive employment of them, as well as the new surrogate modelling method, the antennas are highly optimised in a very reasonable time using a standard desktop 4-core workstation. In addition, the solution quality (i.e. optimisation ability) is clearly improved compared to the reference methods, showing PSADEA's potential for challenging design cases. Future work will include developing PSADEA-based antenna design exploration tools (will be available from www.ai-dac.com) and investigating the performance of PSADEA for other microwave devices.

6 Acknowledgments

This work was partially funded by the UK Engineering and Physical Science Research Council under project EP/M016269/1. The authors would like to thank Prof. Slawomir Koziel, Reykjavik University, Iceland, for test examples.

7 References

[1] Deb, A., Roy, J.S., Gupta, B.: 'Performance comparison of differential evolution, particle swarm optimisation and genetic algorithm in the design of circularly polarized microstrip antennas', *IEEE Trans. Antennas Propag.*, 2014, **62**, (8), pp. 3920–3928

[2] Sato, Y., Campelo, F., Igarashi, H.: 'Meander line antenna design using an adaptive genetic algorithm', *IEEE Trans. Magn.*, 2013, **49**, (5), pp. 1889–1892

[3] Gregory, M.D., Bayraktar, Z., Werner, D.H.: 'Fast optimisation of electromagnetic design problems using the covariance matrix adaptation evolutionary strategy', *IEEE Trans. Antennas Propag.*, 2011, **59**, (4), pp. 1275–1285

[4] Di Barba, P., Dughiero, F., Forzan, M., *et al.*: 'Migration-corrected NSGA-II for improving multi-objective design optimisation in electromagnetics', *Int. J. Appl. Electromagn. Mech.*, 2016, **51**, (2), pp. 161–172

[5] Siah, E.S., Sasena, M., Volakis, J.L.: 'Fast parameter optimisation of large-scale electromagnetic objects using DIRECT with Kriging metamodelling', *IEEE Trans. Microw. Theory Tech.*, 2004, **52**, (1), pp. 276–285

[6] Liu, B., Aliakbarian, H., Ma, Z., *et al.*: 'An efficient method for antenna design optimisation based on evolutionary computation and machine learning techniques', *IEEE Trans. Antennas Propag.*, 2014, **62**, (1), pp. 7–18

[7] 'CST microwave studio: technical specification'. Available at www.cst.com, accessed October 2017

[8] Liu, B., Gielen, G.G.E., Fernandez, F.V.: '*Automated design of analog and high-frequency circuits*' (Springer-Verlag, Berlin, Heidelberg, 2014, 1st edn.)

[9] Koziel, S., Bekasiewicz, A., Zieniutycz, W.: 'Expedited EM-driven multi-objective antenna design in highly-dimensional parameter spaces', *IEEE Antennas Wirel. Propag. Lett.*, 2014, **13**, pp. 631–634

[10] Koziel, S., Bekasiewicz, A.: 'Multi-objective optimisation of expensive electromagnetic simulation models', *Appl. Soft Comput.*, 2016, **47**, pp. 332–342

[11] Liu, B., Zhang, Q., Gielen, G.G.E., *et al.*: 'A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimisation problems', *IEEE Trans. Evol. Comput.*, 2014, **18**, (2), pp. 180–192

[12] Liu, B., Koziel, S., Zhang, Q.: 'A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimisation problems', *J. Comput. Sci.*, 2016, **12**, pp. 28–37

[13] Jones, D.R., Schonlau, M., Welch, W.J.: 'Efficient global optimisation of expensive black-box functions', *J. Glob. Optimisation*, 1998, **13**, (4), pp. 455–492

[14] Santner, T.J., Williams, B.J., Notz, W.I.: '*The design and analysis of computer experiments*' (Springer-Verlag, New York, 2003, 1st edn)

[15] Emmerich, M.T.M., Giannakoglou, K.C., Naujoks, B.: 'Single-and multiobjective evolutionary optimisation assisted by Gaussian random field metamodels', *IEEE Trans. Evol. Comput.*, 2006, **10**, (4), pp. 421–439

[16] Dennis, J.E., Torczon, V.: 'Managing approximation models in optimisation', in Alexandrov, N.M., Hussaini, M.Y. (Eds.): '*Multidisciplinary design optimisation: state-of-the-art*' (SIAM, Philadelphia, PA, USA, 1997), pp. 330–347

[17] Couckuyt, I., Forrester, A., Gorissen, D., *et al.*: 'Blind Kriging: implementation and performance analysis', *Adv. Eng. Softw.*, 2012, **49**, pp. 1–13

[18] Price, K., Storn, R.M., Lampinen, J.A.: '*Differential evolution: a practical approach to global optimisation*' (Springer-Verlag, Berlin, Heidelberg, 2005, 1st edn.)

[19] Stein, M.: 'Large sample properties of simulations using Latin hypercube sampling', *Technometrics*, 1987, **29**, (2), pp. 143–151

[20] Jamil, M., Yang, X.: 'A literature survey of benchmark functions for global optimisation problems', *Int. J. Math. Model. Numer. Optimisation*, 2013, **4**, (2), pp. 150–194

[21] Goldberg, D.E., Deb, K.: 'A comparative analysis of selection scheme used in genetic algorithms', *Found. Genet. Algorithm*, 1991, **1**, pp. 69–93

[22] Koziel, S., Ogurtsov, S.: 'Multi-point response correction for cost-efficient antenna and microwave design optimisation'. Loughborough Antennas Propagation Conf., Loughborough, UK, November 2013, pp. 548–552

[23] Koziel, S., Ogurtsov, S.: 'Multi-objective design of antennas using variable-fidelity simulations and surrogate models', *IEEE Trans. Antennas Propag.*, 2013, **61**, (12), pp. 5931–5939

[24] Michalewicz, Z., Schoenauer, M.: 'Evolutionary algorithms for constrained parameter optimization problems', *Evol. Comput.*, 1996, **4**, (1), pp. 1–32

[25] Wilcoxon, F.: 'Individual comparisons by ranking methods', *Biom. Bull.*, 1945, **1**, (6), pp. 80–83

8 Appendix

8.1 F1: 10-d Ellipsoid problem

$$F(x) = \sum_{i=1}^{d} i x_i^2 \qquad x \in [-20, 20], \quad i = 1, \ldots, 10 \tag{16}$$

8.2 F2: 10-d Ackley problem

$$F(x) = -20\mathrm{e}^{-0.2\sqrt{(1/d)\sum_{i=1}^{d} x_i^2}} - \mathrm{e}^{(1/d)\sum_{i=1}^{d}\cos(2\pi x_i)} + 20 + e \qquad x \in [-32, 32], \quad i = 1, \ldots, 10 \tag{17}$$

8.3 F3: 7-d polynomial problem with non-linear constraints

$$\begin{aligned} F(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\ &\cdots + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\ &x \in [-10, 10], \quad i = 1, \ldots, 7 \\ \text{s.t.}& \\ &127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0 \\ &282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0 \\ &196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0 \\ &-4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11_x 7 \geq 0 \end{aligned} \tag{18}$$

The first and the last constraints are active (i.e. equals to zero) at the global optimum.