www.ietdl.org

Published in IET Microwaves, Antennas & Propagation
Received on 26th December 2011
Revised on 21st May 2012
doi: 10.1049/iet-map.2011.0611

ISSN 1751-8725

Hybrid differential evolution with artificial bee colony and its application for design of a reconfigurable antenna array with discrete phase shifters

X. Li M. Yin

College of Computer Science, Northeast Normal University, Changchun 130117, People's Republic of China
E-mail: Minghao.Yin1@gmail.com

Abstract: This study proposes a new method of designing a reconfigurable antenna with quantised phase excitations using a new hybrid algorithm, called as differential evolution algorithm (DE)/artificial bee colony algorithm (ABC). The main objective of the reconfigurable design problem is to find the element excitation that will result in a sector pattern main beam with low side lobes. The same excitation amplitudes applying to the array with zero phase should be in a high directivity and low side lobe pencil-shaped main beam. The dynamic range ratio is minimised to reduce the effect of mutual coupling between the antenna-array elements. Additionally, compared with the continuous realisation and subsequent quantisation, experimental results indicate that the performance of the discrete realisation of the phase-excitation value can be improved. In order to test the performances of hybrid differential evolution with the artificial bee colony algorithm, the results of some state-of-the-art algorithms are considered. The experimental results indicate the better performance of the DE/ABC.

1 Introduction

The problem of reconfigurable antenna arrays involves radiating multiple patterns using a single power-divided network. During the past decades, this problem has attracted many researchers since the pioneering work of Bucci *et al.* [1, 2]. Moreover, this problem has played an important role in the field of manufacturing and telecommunications science [3, 4]. In order to solve this problem, many methodologies have been proposed to obtain the multi-pattern arrays in the previous literature [4–8].

Evolutionary algorithms perform population-based probabilistic searches with a high speed of convergence rate, and have been proved very successfully in solving the problems of large scale. When it comes to solve reconfigurable antenna problems, compared with the traditional algorithms, evolutionary algorithms have the ability of obtaining excitation phases and amplitudes that can be practically implemented more easily by imposing additional constraints. So it is not uncommon, in the past decade, to see that different kinds of evolutionary algorithms, such as simulated annealing [4], genetic algorithm (GA) [4–6], particle swarm optimisation algorithm [7, 8] and tabu search algorithm [9], have been advanced to handle reconfigurable antenna problem, especially for the problems of large scale. Among them, differential evolution algorithm (DE) [10] and artificial bee colony algorithm (ABC) [11] are the two novel meta-heuristic algorithms introduced recently and have gained significant attentions in the research literatures [12, 13]. DE is a simple and yet powerful population based, direct search algorithm with the generation and test feature for global optimisation problems. The basic idea of DE is to create new candidate solutions by combining the parent individual and several other individuals of the same population, and a candidate solution replaces the parent only if it has better fitness. ABC is a population-based heuristic evolutionary algorithm inspired by the intelligent foraging behaviour of the honeybee swarm.

In the evolutionary algorithm-based antenna-array synthesis producer, phased excitations are always represented by continuous values; however, discrete phase shifters are sometimes used to realise the phase excitation. Therefore the excitation phase values obtained by these approaches are subsequently quantised to the nearest n-bit phase shifter excitation values. In order to solve the reconfigurable antenna array with quantised phase excitations, Baskar proposes a mixed integer optimisation for the first time in an evolution search, namely, the generalised generation-gap model GA (G3-GA) [14]. The objective is to optimise real-valued amplitude excitations and quantised phase excitations [15].

Although meta-heuristic methods have been proved to have superior features to other traditional methods, they also suffer some limitations. Additionally, researchers have found that a skilled combination of two meta-heuristic techniques can improve the performance of the algorithms obviously when dealing with real-world and large-scale problems [16]. Some hybrid heuristic-based optimisation methods have been proposed in the last few years [17–19]. However, this field of study is still in its early days, a large number of future researches are necessary in order to develop hybrid algorithms for optimisation problems.

In this paper, we will use the hybrid DE algorithm with the artificial bee colony to solve reconfigurable antenna-array optimisation with quantised phase excitations. Specifically, a hybrid bee operator of ABC is adopted, combined with mutation, crossover and selection operators of DE to explore and exploit the search space effectively. In order to demonstrate the advantages of the proposed design, the results obtained using continuous-phase excitations followed by quantisation are compared with other algorithms including DE, CoDE, SaDE, jDE and JADE.

The rest of this paper is organised as follows: in Section 2, we will introduce the problem formulation. Section 3 describes the fitness function. Section 4 describes the DE algorithm. Section 5 describes the ABC. Section 6 describes the hybrid DE and artificial bee colony. The corresponding experimental results are given in Section 7. In Section 8, we conclude this paper and point out some future research directions.

2 Problem formulation

The problem described is as follows: in order to design a reconfigurable dual-beam antenna array, an amplitude distribution can generate either a pencil-shaped or a sector power pattern, when the phase distribution of the array is modified appropriately. All excitation phases are set at 0° for the pencil-shaped beam, and varied in the range $-180^\circ \le \phi \le 180^\circ$ for the sector pattern [7]. If the excitation is symmetrical at the centre of the linear array, the array with an even number of uniformly spaced isotropic elements ($2N$) can be written as [14]

$$F(\theta) = 2\sum_{k=1}^{N}(a_{kR}\cos\phi_k - a_{kI}\sin\phi_k) \tag{1}$$

with

$$\phi_k = \frac{2\pi}{\lambda} d_k \sin\theta \tag{2}$$

where d_k is the distance between the position of the kth element and the centre, θ is the scanning angle from the broadside, a_{kR} is the real part of the kth element excitation, a_{kI} is the imaginary part of the kth element excitation and a_{kR} and a_{kI} are set within the range [0, 1] and [−1, 1], respectively. N excitation amplitude and phase coefficients are chosen to optimise the desired pattern. The pencil and sector patterns should have a high directivity, low side lobe pencil-shaped main beam and a wide sector beam.

3 Fitness function evaluation

For the reconfigurable dual-beam optimisation, the objective of the fitness function must qualify the entire array radiation pattern. The calculated pattern can be described in terms of the criteria of the desired pattern. The fitness function for the dual-beam optimisation can be described as follows [7]

$$E(P) = \sum_{i=1}^{3}(P_{i,d}^{(p)} - P_i^{(i)})^2 + \sum_{i=1}^{4}(P_{i,d}^{(s)} - P_i^{(s)})^2 \tag{3}$$

where the superscript p is the design specification for the pencil pattern, the superscript s is the design specification of the sector pattern, the subscript d indicates the desired value of the design specification and P indicates the applicable fitness factor in Table 1. The first part of this fitness function is summarised over the first column of Table 1, and the other part of this function is summarised by the second column. Different from the fitness function of the pencil beam pattern, the pattern ripple needs to be calculated for the sector pattern.

Table 1 Design specifications

Design parameters	Pencil pattern	Sector pattern
side-lobe level (SLL)	−30 dB	−25 dB
half-power bandwidth (HPBW)	6.8°	24°
bandwidth at SLL	20°	40°
ripple	NA	0.5 dB

In order to reduce the effect of coupling between elements, an additional term is included in the objective function (4) [14]. The ratio is used to minimise the coupling effect between the maximum and minimum excitation amplitudes. The minimisation of the amplitude-excitation dynamic range (ADR) can reduce the mutual coupling problem [20, 21]. The objective function can be expressed as follows

$$Ec(P) = \sum_{i=1}^{3}(P_{i,d}^{(p)} - P_i^{(i)})^2 + \sum_{i=1}^{4}(P_{i,d}^{(s)} - P_i^{(s)})^2 + \text{ADR} \tag{4}$$

where ADR is the amplitude–dynamic ratio. The ADR is defined as the ratio between the maximum excitation amplitude to the minimum excitation amplitude. The differences between the excitation amplitudes are minimised by minimising the ADR; therefore the effect of coupling can be minimised.

4 Differential evolution algorithm

DE [10] is a simple and yet powerful heuristic method for solving non-linear, non-differentiable and multimodal optimisation problems. Like other Eas, DE starts with an initial random population and searches towards the global optimum by some iteration operations including mutation, crossover and selection. The main idea behind DE is a scheme for producing trial vectors according to the manipulation of the target vector and difference vector. If the problem is the minimisation problem, the trail vector competes with the current population vector and the better one is selected to enter the next generation. Different kinds of strategies of DE have been proposed based on the target vector selected, and the number of different vectors used. In this paper, we use the strategy, DE/rand/1/bin, described as follows:

For each target vector $\boldsymbol{x}_i(t)$, trail vector $\boldsymbol{v}_i(t)$, $i = 1, \ldots, NP$, let D be the dimension of the target vector, and G be the G generation. The mutant vectors are generated in these DE/rand/1/bin strategies respectively.

For DE/rand/1/bin

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{a,G} + F(\boldsymbol{x}_{b,G} - \boldsymbol{x}_{c,G}) \tag{5}$$

where a, b, c, $d \in [1, \ldots, NP]$ are randomly chosen as

doi: 10.1049/iet-map.2011.0611

integers, and $a \neq b \neq c \neq d \neq i$. F is the scaling factor controlling the amplification of the DE.

Following the mutation phase, the crossover operator is applied to the population. The crossover operator, implements a recombination of the trial vector and the parent vector to produce the offspring. This operator is calculated as

$$u_{j,i,G} = \begin{cases} \boldsymbol{v}_{j,i,G}, & (\text{rand}_j[0, 1] \leq CR) \text{ or } (j = j_{\text{rand}}) \\ \boldsymbol{x}_{j,i,G}, & \text{otherwise} \end{cases} \quad (6)$$

where $j = [1, \ldots, D]$, $\text{rand}_j \in [0, 1]$, $j_{\text{rand}} = [1, \ldots, D]$ is the randomly chosen index, CR is the DE control parameter that is called the crossover rate and is a user-defined parameter within the range [0, 1]. $v_{j,i,G}$ is the differential vector of the jth particle in the ith dimension at the Gth iteration, and $u_{j,i,G}$ denotes the trail vector of the jth particle in the ith dimension at the Gth iteration. The selection operator is used to choose the next population between the trail population and the target population

$$\boldsymbol{x}_{i,G+1} = \begin{cases} u_{i,G}, & f(u_{i,G}) < f(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G}, & \text{otherwise} \end{cases} \quad (7)$$

The standard DE algorithm can be described as Fig. 1

5 ABC algorithm

The artificial bee colony is an evolutionary algorithm first introduced by Karaboga in 2005. This algorithm simulates the foraging behaviour of the bee colony. ABC algorithm is a population-based algorithm to be developed by taking into consideration the thought that how honeybee swarm would find food. In this algorithm, the model of the ABC algorithm consists of three groups of bees: employed bees, onlooker bees and scout bees. Employed bees are responsible for exploiting the nectar sources explored before and sharing their information with the onlookers within the hive. After that the onlookers will select one of the food sources within the neighbourhood of the food source. An employed bee becomes a scout if the food source is abandoned, and then starts to search a new food source randomly [11].

The ABC algorithm is an iterative algorithm. It starts by associating all employed bees with randomly generated food solutions. The initial population is very important in the meta-heuristic algorithms and can be generated by different ways. Each individual is randomly producing and is used in this study. The initial population of solutions is filled with SN number of randomly generated D dimensions.

In order to produce a candidate food position v_{ij} from the old one x_{ij} in the neighbourhood of its present position is as follows

$$\boldsymbol{v}_{ij} = \boldsymbol{x}_{ij} + \varphi_{ij}(\boldsymbol{x}_{ij} - \boldsymbol{x}_{kj}) \quad (8)$$

$$k = \text{int}(\text{rand} * SN) + 1$$

where $\varphi_{ij} = (\text{rand} - 0.5) \times 2$ is a uniformly distributed real random number within the range $[-1, 1]$, $i \in \{1, 2, \ldots, SN\}$, $k \in \{1, 2, \ldots, SN\}$ and $k \neq i$ and $j \in \{1, 2, \ldots, n\}$ are randomly chosen indexes. After producing the new solution v_i, it will be evaluated and compared with x_i. If the objective fitness of v_i is smaller than the fitness of x_i, v_i is accepted as a new basic solution. Otherwise x_i would be obtained.

When all employed bees finish this process, an onlooker bee can obtain the information of the food sources from all employed bees and choose a food source depending on the probability value associated with the food source, using the

```
procedure Algorithm description of DE algorithm
begin
Step 1: Set the generation counter G=0; and randomly initialise a
population of NP individuals X_i. Initialise the parameter F, CR
Step 2: Evaluate the fitness for each individual in P.
Step 3: while stopping criteria is not satisfied do
      for i= 1 to NP
            select randomly  a ≠ b ≠ c ≠ d ≠ i
              for j=1 to D
                j_rand = ⌊rand(0,1) * D⌋
                If rand(0,1) ≤ CR or j== j_rand  then
                      u_i,j = x_a,j + F × (x_b,j − x_c,j)
                Else
                      u_i,j = x_i,j
                end if
              end for
     end for
      for i=1 to NP do
      Evaluate the offspring  u_i
         If  u_i  is better than  P_i  then
              P_i = u_i
         end if
      end for
       Memorise the best solution achieved so far
Step 4: end while
end
```

Fig. 1 *Procedure algorithm description of DE algorithm* following expression

$$p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \tag{9}$$

where fitness_i is the fitness value of the solution i evaluated by its employed bee. Obviously, when the maximal value of the food source decreases, the probability with the preferred source of an onlooker bee decreases proportionally. Then the onlooker bees produce a new source according to (9). The new source will be evaluated and compared with the primary food solution. If the new source has a better nectar amount than the primary food solution, it will be accepted.

After all the onlookers have finished this process, the sources are checked to determine whether they are to be abandoned. If the food source does not improve after a determined number of the trails 'limit', the food source is abandoned. Its employed bee will become a scout and then will search for a food source randomly as follows

$$\boldsymbol{x}_{ij} = LB_j + (UB_j - LB_j) \times r \tag{10}$$

where r is a uniform random number in the range [0, 1].

After the new source is produced, another iteration of the ABC algorithm will begin. The whole process repeats again till the termination condition is met.

6 Our approach: DE/ABC

In this section, different steps of DE/ABC approach are described below:

6.1 Hybrid bee operator

The main operator of DE/ABC is the hybrid bee operator, which hybridises the DE operator with the ABC operation. Crossover operation of the DE algorithm is applied to each pair of the target vector $\boldsymbol{x}_{j,i,G}$ and its corresponding mutate vector $\boldsymbol{v}_{j,i,G}$ to generate a trail vector. In the standard DE algorithm, DE employs the binomial crossover defined as follows

$$u_{j,i,G} = \begin{cases} \boldsymbol{v}_{j,i,G}, & (\text{rand}_j[0, 1] \le CR) \text{ or } (j = j_{\text{rand}}) \\ \boldsymbol{x}_{j,i,G}, & \text{otherwise} \end{cases}$$

The crossover operator can find the globally optimal region. However, it is cannot converge rapidly to the globally optimal solution. Employed bees fly onto the source which they are exploiting. In order to solve this problem, we tackle by integrating an employed bee operator of ABC to maintain the diversity and obtain good solution rapidly at the same time (Fig. 2)

The crucial idea behind DE is a scheme for producing trial vectors according to the manipulation of the target vector and difference vector. From the algorithm, we can find that the proposed bee operation is based on the main update operator of ABC. The core idea of the proposed hybrid bee colony is based on two considerations. On the one hand, the employed bee colony exploited the nectar sources explored before and gave the information to the waiting bees in the hive about the quality of the food source sites, which they are exploiting and the onlooker bees waiting in the hive watch the dances advertising the profitable sources and choose a source site depending on the frequency of a dance proportional to the quality of the source. On the other hand, the mutation operator of the DE is able to explore the new space. From the analysis, it can be seen that the hybrid bee colony can balance the exploration and the exploitation effectively.

```
procedure hybrid bee operator of DE/ABC
begin
    for i= 1 to NP
        select randomly  a ≠ b ≠ c ≠ d ≠ i
        for j=1 to D
            j_rand = ⌊rand(0,1) * D⌋
            If rand(0,1) ≤ CR or j== j_rand  then
                u_i,j = x_a,j + F × (x_b,j − x_c,j)
            else
                If rand<0.2
                    u_i,j = x_i,j + (x_i,j − x_d,j) *   i,j
                end if
            end if
        end for
    end for
end.
```

Fig. 2 *Procedure hybrid bee operator of DE/ABC*

6.2 Main procedure of DE/ABC

By incorporating the above-mentioned hybrid bee operator into DE, the DE/ABC has been developed as a new algorithm. The hybrid method is described as Fig. 3.

As we know, the standard DE algorithm is good at exploring the search space and locating the region of the global minimum, but it is relatively slow at the exploitation of the solution. On the other hand, the standard ABC algorithm is usually quick at the exploitation of the solution though its exploration ability is relatively poor. Therefore, in this literature, a hybrid meta-heuristic algorithm by integrating the artificial bee colony into DE, called as DE/ABC is used to solve the problem of reconfigurable antenna array. The difference between DE/ABC and DE is that the hybrid bee operator is used to replace the original DE mutation operator. In this way, this method can explore the new search space by the mutation of the DE algorithm and exploit the population information with the employed onlooker bee operator of ABC, and therefore can overcome the lack of exploitation of the DE algorithm.

7 Experimental results

7.1 Compared DE/ABC with ABC and DE for global optimisation problem

To evaluate the performances of the DE/ABC and ABC and DE, we apply them to six classical benchmark functions. These functions are presented in Table 2. In Table 2, Range denotes a subset of Range^D. D is the dimension of classical functions. The global minimum values of six classical benchmark functions of Table 2 are zeros. Functions 1 and 2 are unimodal high-dimensional functions. Functions 3 and 6 are multimodal high-dimensional functions.

The performances of the DE/ABC are compared with those of the DE and ABC algorithms. For all the functions, the population size is 100 and the maximum number of generations is 1500. For the parameter of DE/ABC, F is

doi: 10.1049/iet-map.2011.0611

```
procedure Algorithm description of DE/ABC
begin
Step 1: Set the generation counter G=0; and randomly initialise a
population of NP individuals X_i. Initialise the parameter F, CR, limit.
Step 2: Evaluate thefitness for each individual in P.
Step 3: while stopping criteria is not satisfied do
    %% hybrid employed bee colony
    for i= 1 to NP
        select randomly  a ≠ b ≠ c ≠ d ≠ i
         for j=1 to D
           j_rand = ⌊rand(0,1) * D⌋
           If rand(0,1) ≤ CR or j== j_rand  then
               u_i,j = x_a,j + F × (x_b,j − x_c,j)
           else
               If rand<0.2
               u_i,j = x_i,j + (x_i,j − x_d,j) *   i,j
               end if
           end if
         end for
   end for
    %% hybrid Onlooker bee colony
   i=1;
   t=0;
    which t<NP
     if rand<prob(i)
         t=t+1;
        select randomly  a ≠ b ≠ c ≠ d ≠ i
         for j=1 to D
           j_rand = ⌊rand(0,1) * D⌋
           If rand(0,1) ≤ CR or j== j_rand  then
               u_i,j = x_a,j + F × (x_b,j − x_c,j)
           else
               If rand<0.2
               u_i,j = x_i,j + (x_i,j − x_d,j) *   i,j
               end if
           end if
         end for

    end if
      i=i+1;
   if i==NP+1
      i=1;
  end if
  end while
 %%Scout bee colony
    If max(trial_i >limit)
        Replace x_i with a new randomly produced solution by
        x_ij = l_j + (u_j − l_j) × r
    End if
       Memorise the best solution achieved so far
Step 4: end while
end
```

Fig. 3 *Procedure algorithm description of DE/ABC*

0.5, *CR* is 0.9 and the limit is 100. For the parameter of DE, *F* is 0.5 and *CR* is 0.9. For the ABC algorithm, the limit is 100.

In the experiment, the mean results of 30 independent runs are summarised in Table 3. Compared with the DE algorithm, as can be seen in Table 3, we can find that the DE/ABC is significantly better than DE on four functions except *f05*. For the multimodal functions with many local minimums, that is, *f*03–*f*06, it is clear that the best results are obtained by DE/ABC. DE may trap into the local minima for two out of four functions. The DE/ABC can find better solutions than the DE algorithm within the maximum number of generations. This result illustrates that the algorithm has better ability to escape from poor local optima and locate a good near-global optimum.

Compared with ABC, from Table 3, it is obvious that DE/ABC performs better solutions than ABC except *f*02. For the multimodal function, the DE/ABC can provide better solutions than ABC for all functions.

In general, the performance of DE/ABC is highly competitive with DE, especially for the high-dimensional problems. Moreover, DE/ABC is better than ABC for some functions.

Table 2 Benchmark functions based in our experimental study

Test function	D	Range	Optimum
$f_{01}=\sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_{02}=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[−30, 30]	0
$f_{03}=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)+20+e$	30	[−32, 32]	0
$f_{04}=\frac{1}{400}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[−600, 600]	0
$f_{05}=\frac{\pi}{D}\left\{\begin{array}{l}10\sin^2(\pi y_i)+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)]\\+(yD-1)^2+\sum_{i=1}^{D}u(x_i,10,100,4)\end{array}\right\}$	30	[−50, 50]	0
$y_i=1+((x_i+1)/4)\ u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\0 & -a<x_i<a\\k(-x_i-a)^m & x_i<-a\end{cases}$			
$f_{06}=0.1\left\{10\sin^2(\pi y_i)+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)]+(yD-1)^2\right\}+\sum_{i=1}^{D}u(x_i,10,100,4)$	30	[−50, 50]	0

Table 3 Comparisons of ABC, DE and DE/ABC

F	ABC Mean	Standard	DE Mean	Standard	DE/ABC Mean	Standard
f01	3.4163e−020	2.1396e−020	5.2833e−014	3.5135e−014	5.7621e−049	1.5839e−048
f02	0.1243	0.1082	16.6681	1.0638	25.0887	18.7927
f03	1.1904e−009	4.4766e−010	2.2021e−008	6.0555e−009	4.7961e−015	1.1234e−015
f04	1.1853e−009	4.1106e−010	7.3727e−008	3.1395e−008	0	0
f05	7.1651e−022	7.1565e−022	6.9083e−015	8.2614e−015	2.1351e−030	6.7022−030
f06	4.1637e−020	5.9332e−020	2.5765e−014	1.9767e−014	1.8428e−032	9.2968e−033

7.2 Reconfigurable antenna-array design and parameter setting

To evaluate the performance of the DE/ABC, the benchmark problems for the experiments are also used in [14]. In the first experiment, there are 20 design parameters with continuous values. In the second experiment, there are also 20 design parameters. Among them, ten-phase coefficients are represented as discrete variables, and the other ten-phase coefficients are represented as continuous variables.

In experiment 1, the results of the excitation phases cannot be used and approximated to the nearest values for an n-bit phase. In this paper, we will compare DE/ABC with other algorithms including DE, G3-GA, CoDE, SaDE, jDE and JADE based on these two experiments. In experiment 2, ten-phase excitations are indicated as quantised values corresponding to the n-bit phase shifter is used. Therefore the values of the phase excitation are quantised between -180° and 180° with 5.625° per step. For simulating DE/ABC, the population size *NP* is 20. The maximum function evaluations are 20 000. The crossover rate *CR* is 0.9. The scale factor *F* is 0.5. For simulating DE algorithm and generalised generation gap GA (G3-GA), the population size *NP* is 20, the maximum function evaluations are 20 000, the crossover rate *CR* is 0.9 and the scale factor *F* is 0.5. In G3-GA, the number of the offspring is $\lambda = 6$, the maximum function evaluations are 20 000, the population size *NP* is 500 and $\sigma_\alpha = \sigma_\beta = 0.25$. In order to compare

Table 4 Optimum results of experiment 1 and 2 without ADR

Element number	Experiment 1		Experiment 1 after quantisation		Experiment 2	
	Amplitude	Phase, deg.	Amplitude	Phase, deg.	Amplitude	Phase, deg.
1/20	0.203	−179.8	0.203	−180.0	0.147	−8.6
2/19	0.159	−141.9	0.159	−140.0	0.170	−25.7
3/18	0.243	−157.3	0.243	−157.1	0.281	−37.1
4/17	0.363	−107.2	0.363	−105.7	0.310	−60.0
5/16	0.456	−103.5	0.456	−100.0	0.451	−77.1
6/15	0.598	−81.2	0.598	−77.1	0.581	77.1
7/14	0.746	90.9	0.746	94.3	0.670	−111.4
8/13	0.835	97.2	0.835	100.0	0.787	−94.3
9/12	0.912	101.9	0.912	105.7	0.893	82.9
10/11	0.954	96.0	0.954	100.0	0.918	88.6
ADR	5.99		5.99		6.23	
fitness value	0.16		5.0317		0.16	

IET Microw. Antennas Propag., 2012, Vol. 6, Iss. 14, pp. 1573–1582
doi: 10.1049/iet-map.2011.0611

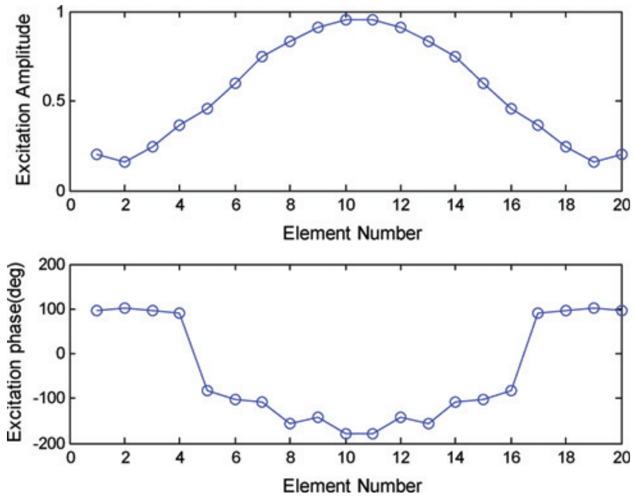

Fig. 4 *Amplitude and phase excitation (experiment 1)*

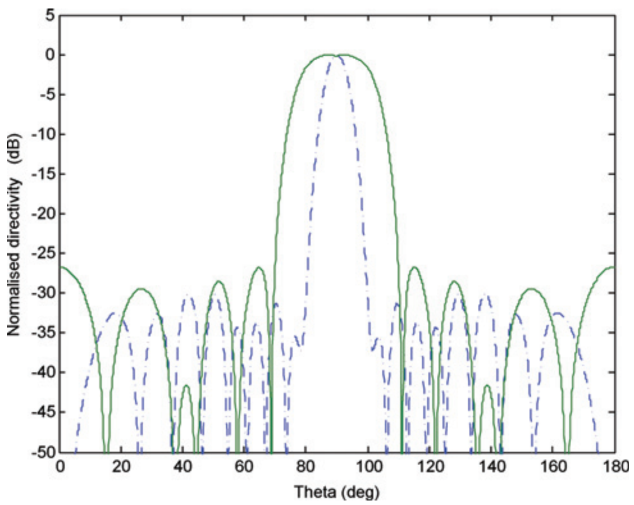

Fig. 5 *Dual-beam array patterns (experiment 1)*

fairly, we set these algorithms with the same fitness evaluations. In ABC, the population size is 20. The limit is 100.

7.3 Optimisation without ADR

In this section, we will use DE/ABC for the reconfigurable antenna-array design without the coupling effects. The objection function (3) is as the fitness. Table 4 shows the experiment results of the excitation amplitude and phase.

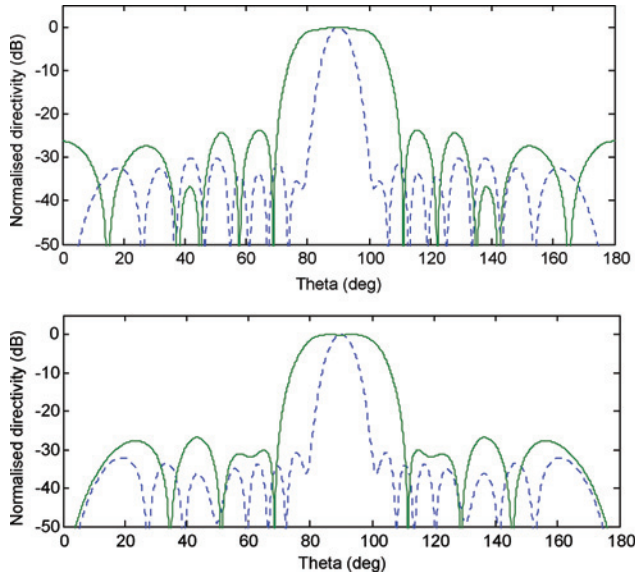

Fig. 6 *Dual-beam array pattern: experiment 1 (top) after quantisation and experiment 2*

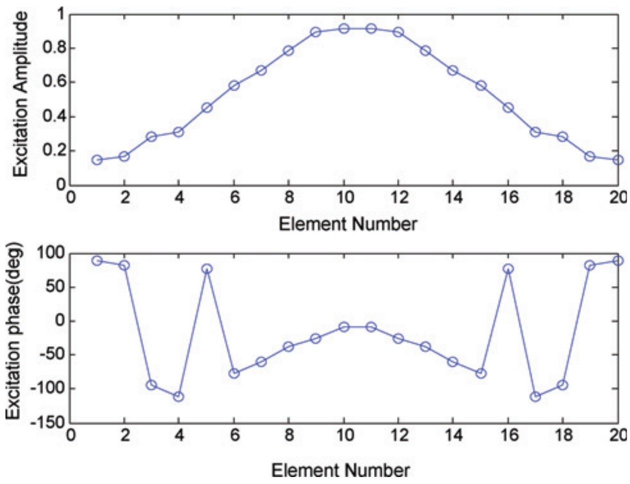

Fig. 7 *Amplitude and phase excitation (experiment 2)*

The best of optimal results for experiment 1 (after quantisation of the phase excitations), and experiment 2 are shown in Table 4. The table also illustrates the ADR of the optimised excitation amplitudes and fitness function value. The optimised excitation patterns and dual-beam patterns are described in Figs. 4 and 5, respectively. Fig. 5 shows the satisfaction of designed parameters simultaneously for both the pencil and sector beam.

Table 5 Effects of quantisation on different design specifications

		Pencil beam			Sector beam				Fitness
		HPBW	SLLBW	SLL	HPBW	SLLBW	SLL	Ripple	
DE	continuous phase excitation(experiment 1)	0.4	0	0	0	0	0	0	0.16
	after quantisation	0.4	0	0	1.8	1.2	0	0	4.84
	optimisation with discrete variable(2)	0.6	0	0	0	0	0	0	0.36
DE/ABC	continuous phase excitation(experiment1)	0.4	0	0	0	0	0	0	0.16
	after quantisation	0.4	0	0	1.8	0.2	1.26	0	5.03
	optimisation with discrete variable(2)	0.4	0	0	0	0	0	0	0.16

Table 6 Optimum results of experiment 1 and 2 with ADR

Element numbber	Experiment 1		Experiment 1 after quantisation		Experiment 2	
	Amplitude	Phase, deg.	Amplitude	Phase, deg.	Amplitude	Phase, deg.
1/20	0.211	−9.5	0.211	−8.6	0.218	−168.6
2/19	0.211	−16.8	0.211	−14.3	0.218	−162.9
3/18	0.221	−55.9	0.221	−54.3	0.226	−128.6
4/17	0.357	−59.8	0.357	−60.0	0.368	−145.7
5/16	0.439	−89.6	0.439	−88.6	0.452	117.1
6/15	0.577	105.6	0.577	111.4	0.585	−60.0
7/14	0.685	−77.3	0.685	−77.1	0.704	−77.1
8/13	0.794	88.1	0.794	94.3	0.803	−54.3
9/12	0.870	−41.9	0.870	−37.1	0.889	60.0
10/11	0.911	39.9	0.911	42.9	0.927	−105.7
ADR	4.31		4.31		4.25	
fitness value	0.05		7.19		0.09	

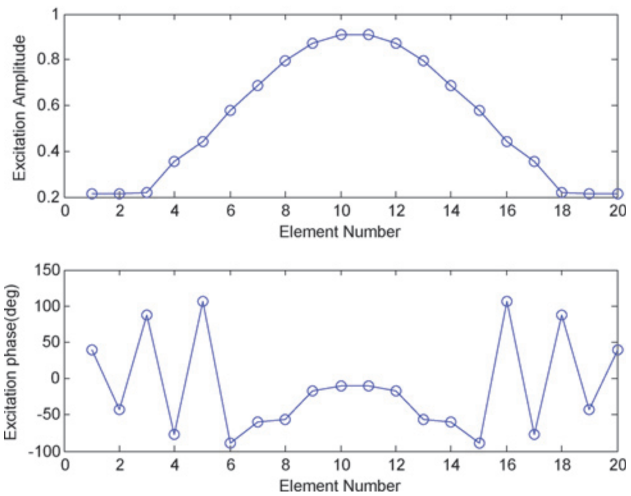

Fig. 8 *Amplitude and phase excitation (experiment 1) with coupling effect*

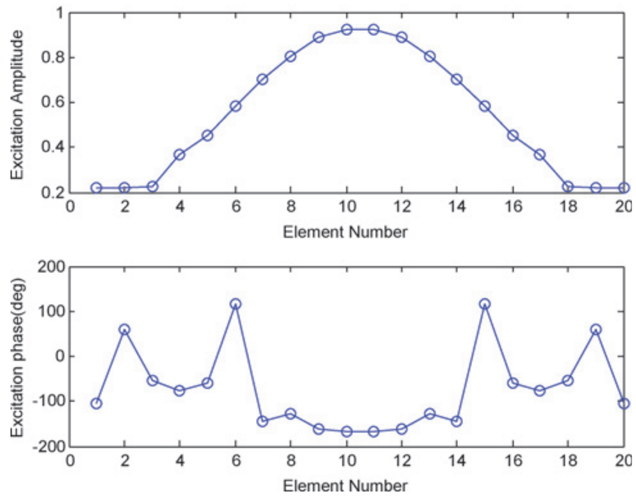

Fig. 10 *Amplitude and phase excitation (experiment 2) with coupling effect*

For the DE/ABC, the best dual-beam pattern is 0.16 for experiment 1. After quantising the optimum phase values, we can find the fitness value increases to 5.03 because the sector beam increases most of the fitness value in Table 4. From the quantisation of the optimum result obtained in experiment 1, it may not be optimum for the discrete case.

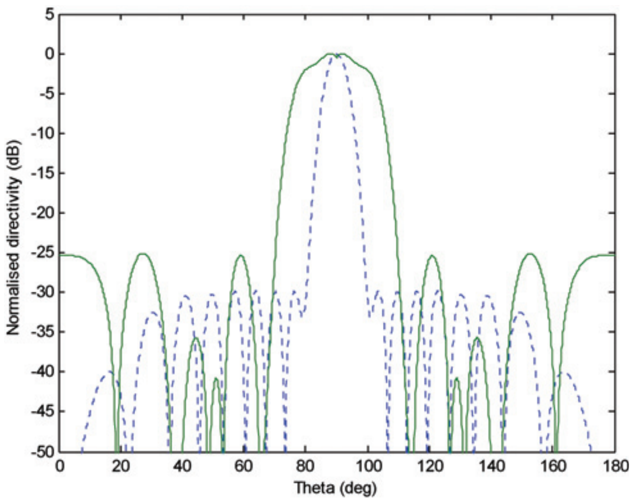

Fig. 9 *Dual-beam array pattern (experiment 1) with coupling effect*

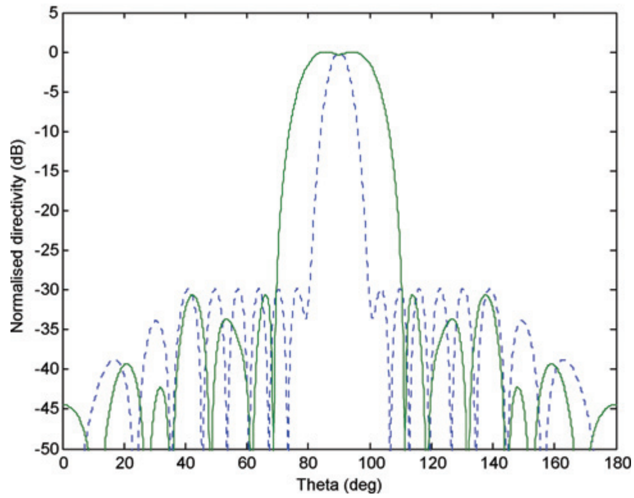

Fig. 11 *Dual-beam array pattern (experiment 2) with coupling effect*

doi: 10.1049/iet-map.2011.0611

Table 7 Comparison of G3-GA with DE

	Experiment 1 without ADR	Experiment 2 without ADR	Experiment 1 with ADR		Experiment 2 with ADR	
	Fitness	Fitness	ADR	Fitness	ADR	Fitness
G3-GA	0.16	0.619	4.4137	0.1028	5.8026	0.2630
ABC	0.16	0.16	4.36	0.04	4.76	0.22
DE	0.16	0.36	4.3470	0.04	4.7190	0.16
DE/ABC	0.16	0.16	4.31	0.05	4.25	0.09

Table 8 Comparison of DE/ABC with DE, CoDE, SaDE, jDE and JADE

	Experiment 1 without ADR	Experiment 2 without ADR	Experiment 1 with ADR		Experiment 2 with ADR	
	Fitness	Fitness	ADR	Fitness	ADR	Fitness
DE	0.16	0.36	4.3470	0.04	4.7190	0.16
CoDE	0.64	0.64	4.94	0.23	4.31	0.09
SaDE	0.16	0.36	4.63	0.28	4.38	0.12
jDE	0.36	0.36	4.40	0.10	4.33	0.26
JADE	0.36	0.17	4.35	0.09	4.47	0.16
DE/ABC	0.16	0.16	4.31	0.05	4.25	0.09

Therefore in the evolutionary process, discrete values represent the phase excitation that can eliminate the error arising because of quantisation.

The deviation between the desired and the computed design specification of the optimised results in experiment 1 and experiment 2 are shown in Table 5. Fig. 6 shows experiment 1 (after quantisation) and experiment 2 for the dual-beam patterns. The difference between experiment 1 and experiment 2 is clearly shown. The best amplitude and phase excitations with discrete values are shown in Fig. 7.

7.4 Optimisation with ADR

In this section, we will use the DE/ABC for the reconfigurable antenna-array design with the coupling effects. The objective function (4) is as the fitness. Table 6 illustrates the experiment results of experiment 1 and experiment 2. The table also generates the ADR and the fitness values. The best fitness is less than the previous in this case. Moreover, in experiment 1, the ADR is reduced from 5.99 to 4.31. In experiment 2, the ADR is reduced from 6.23 to 4.25. Hence, we can reduce the coupling effects by minimising the dynamic range ratio. Figs. 8 and 9 show the excitation pattern and dual-beam pattern obtained in experiment 1. Figs. 10 and 11 show the same thing as in Figs. 8 and 9 (the excitation pattern and dual-beam pattern obtained in experiment 2).

7.5 Comparison of DE/ABC with DE, ABC and G3-GA [14]

In order to study the effect of DE/ABC, we carried out a scalability study to compare the algorithm with the generalised generation gap GA, artificial bee colony and DE. The experiment is conducted for the determination of amplitude and phase excitation patterns for the dual beam optimisation with quantisation. The best fitness is reported in Table 7. As can be seen in Table 7, we can find that the DE/ABC can obtain better solution for experiment 1 and experimental 2. Especially, for the dual beam optimisation with quantisation, DE/ABC can perform better than G3-GA, ABC and DE.

7.6 Comparison of DE/ABC with DE, CoDE, SaDE, jDE and JADE

In order to evaluate the effectiveness and efficiency of DE/ABC, we compare its performance with DE, CoDE [22], SaDE [23], jDE [24] and JADE [25]. Brest *et al.* [24] proposed a self adaptive parameter setting in DE in order to avoid the manual parameter setting of *F* and *CR*. The parameter control technique is based on the self adaption of two parameters associated with the evolutionary process. Qin and Suganthan [23] propose a self-adaptive DE algorithm (SaDE), in which both the trail vector generation strategies and their associated control parameter values are gradually self-adaptive by learning from their previous experiences in generating promising solutions. In the JADE propose by Zhang and Sanderson [25], a normal distribution and a Cauchy distribution are utilised to generate *F* and *CR* for each target vector, respectively. JADE extracts information from the recent successful *F* and *CR* and uses such information for generating new *F* and *CR*. Wang [22] proposes a novel method, called composite DE (CoDE), which has been proposed in this paper. This method uses three trial vector generation strategies and three control parameter settings. It randomly combines them with the generated trial vectors. Each method was run 30 times on each test function. Table 8 summarises the experimental results. As can be seen in Table 8, DE/ABC significantly outperforms DE, CoDE, SaDE, jDE and JADE for experiment 1 and experimental 2. By minimising the dynamic ratio, we can find that the DE/ABC can provide 4.25 (ARD) and 0.09 (fitness) better than those of the other algorithms.

8 Conclusions

The application of hybrid DE with artificial bee colony for the reconfigurable antenna array with quantised phase shifter is discussed in this paper. The effectiveness of the proposed algorithm is demonstrated in the design of a reconfigurable array antenna without and with the quantised phase excitations. The effect of the quantisation in the continuous formulation of the phased excitation is presented. In order to reduce the effect of mutual coupling between the antenna-array elements, the dynamic range ratio is minimised. The experimental results clearly indicate superior performance of the proposed algorithm in comparison with some recent optimisation algorithms. We hope that this paper sparks a new venue of research in the problem of solving reconfigurable antenna array.

9 Acknowledgments

This research is fully supported by Opening Fund of Top Key Discipline of computer software and theory in Zhejiang Provincial colleges at Zhejiang Normal University under Grant Nos. ZSDZZZZXK37, the Fundamental Research Funds for the Central Universities Nos. 11CXPY010 and the Natural Science Foundation of Jilin Province Nos. 20110104.

10 References

1 Bucci, O.M., D'Elia, G.K., Mazzarella, G., Panariello, G.: 'Antenna pattern synthesis: a new general approach', *Proc IEEE*, 1994, **82**, pp. 358–371
2 Bucci, O.M., Mazzarella, G., Panariello, G.: 'Reconfigurable arrays by phase-only control', *IEEE Trans. Antennas Propag.*, 1991, **39**, pp. 919–925
3 Durr, M., Trastoy, A., Ares, F.: 'Multiple-pattern linear antenna arrays with single prefixed distributions: modified Woodward–Lawson synthesis', *Electron. Lett.*, 2000, **26**, pp. 1345–1346
4 Rahmat-Samii, Y., Michielssen, E.: 'Electromagnetic optimization by genetic algorithms' (Wiley, New York, 1999)
5 Alphones, A., Passoupathi, V.: 'Null steering in phased arrays by positional perturbations: a genetic algorithm approach'. IEEE Conf. Phased Arrays, 1996, pp. 203–207
6 Baskar, S., Alphones, A., Suganthan, P.N.: 'Design of reconfigurable antenna array using improved multi-agent genetic algorithms'. Asia-Pacific Microwave Conf., New Delhi, 2004, http://www.ntu.edu.sg/home/EPNSugan/
7 Gies, D., Rahmat-Samii, Y.: 'Particle swarm optimization for reconfigurable phase differentiated array design', *Microw. Opt. Technol. Lett.*, 2003, **38**, pp. 168–175
8 Boeringer, D.W., Werner, D.H.: 'Particle swarm optimization versus genetic algorithm for a phased array synthesis', *IEEE Trans. Antennas Propag.*, 2004, **52**, pp. 771–789
9 Guney, K., Akdagli, A.: 'Null steering of linear antenna arrays using a modified Tabu search algorithm', *Prog. Electromagn. Res. (PIER)*, 2001, **33**, pp. 167–182
10 Storn, R., Price, K.: 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *J. Glob. Optim.*, 1997, **11**, pp. 341–359
11 Karaboga, D., Basturk, B.: 'On the performance of artificial bee colony (ABC) algorithm', *Appl. Soft Comput.*, 2008, **8**, pp. 687–697
12 Karaboga, D., Akay, B.: 'A comparative study of artificial bee colony algorithm', *Appl. Math. Comput.*, 2009, **214**, pp. 108–132
13 Noman, N., Iba, H.: 'Accelerating differential evolution using an adaptive local search', *IEEE Trans. Evol. Comput.*, 2008, **12**, (1), pp. 107–125
14 Baskar, S., Alphones, A., Suganthan, P.N.: 'Genetic-algorithm-based design of a reconfigurable antenna array with discrete phase shifters', *Microw. Opt. Technol. Lett.*, 2005, **45**, (6), pp. 461–465
15 Dobias, F., Gunther, J.: 'Reconfigurable array antennas with phase-only control of quantized phase shifters'. IEEE APS Int Symp., Newport Beach, CA, 1995, pp. 35–39
16 Zhao, S.Z., Suganthan, P.N., Pan, Q.-K., Tasgetiren, M.F.: 'Dynamic multi-swarm particle swarm optimizer with harmony search', *Expert Syst. Appl.*, 2011, **38**, (4), pp. 3735–3742
17 Choi, D.H.: 'Cooperative mutation based evolutionary programming for continuous function optimization', *Oper. Res. Lett.*, 2002, **30**, (3), pp. 195–201
18 Zhang, C.S., Ning, J.X., Lu, S., Ouyang, D.T., Ding, T.N.: 'A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization', *Oper. Res. Lett.*, 2009, **37**, pp. 117–122
19 Shi, X.H., Liang, Y.C.: 'An improved GA and a novel PSO-GA-based hybrid algorithm', *Inf. Process. Lett.*, 2005, **93**, pp. 255–261
20 Rodriguez, J.A., Ares, F., Moreno, E.: 'Linear array pattern synthesis optimizing array element excitations using the simulated annealing technique', *Microw. Opt. Technol. Lett.*, 1999, **23**, pp. 224–226
21 Hansen, R.C.: 'Phased array antennas' (Wiley-Interscience, New York, 1998)
22 Wang, Y., Cai, Z., Zhang, Q.: 'Differential evolution with composite trial vector generation strategies and control parameters', *IEEE Trans. Evol. Comput.*, 2011, **15**, (1), pp. 55–66
23 Qin, A.K., Huang, V.L., Suganthan, P.N.: 'Differential evolution algorithm with strategy adaptation for global numerical optimization', *IEEE Trans. Evol. Comput.*, 2009, **13**, (2), pp. 398–417
24 Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: 'Self adapting control parameters in differential evolution: a comparative study on numerical benchmark problems', *IEEE Trans. Evol. Comput.*, 2006, **10**, (6), pp. 647–657
25 Zhang, J.Q., Sanderson, A.C.: 'JADE: adaptive differential evolution with optional external archive', *IEEE Trans. Evol. Comput.*, 2009, **13**, (5), pp. 945–958

doi: 10.1049/iet-map.2011.0611