Communication

Successive Boolean Optimization of Planar Pixel Antennas

Shanpu Shen, Ying Sun, Sichao Song, Daniel P. Palomar, and Ross D. Murch

***Abstract*— We propose a novel heuristic method for optimizing planar pixel antennas which we refer to as successive exhaustive Boolean optimization (SEBO). The key step in SEBO is to cyclically optimize a multivariable binary optimization problem by exhaustively searching a subset of binary variables. An adaptive version of SEBO is also introduced. We provide both simulation and experimental results for optimized antennas covering single and dual bands for mobile phone applications. Comparisons with genetic algorithms, binary particle swarm optimization, and angle modulate differential evolution are also provided and these show that SEBO and adaptive SEBO have up to 50% less computational complexity and can satisfy the design objective more consistently. A major advantage of SEBO and adaptive SEBO is that only one parameter needs to be set up for the optimization, while for evolutionary optimization two or more parameters usually need to be set up.**

***Index Terms*— Binary optimization, pixel antenna, successive exhaustive Boolean optimization (SEBO).**

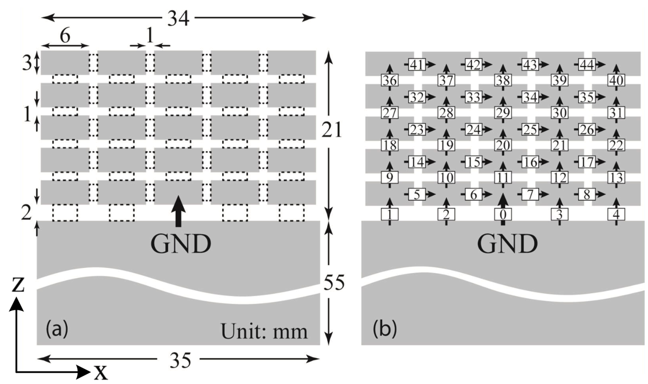

Fig. 1. (a) Geometry and dimensions of the illustrative pixel antenna and (b) modeling the pixel antenna as a multiport network in IMPM.

I. Introduction

The concept of pixel antennas is based on discretizing a radiation surface into small elements and connecting them through switches or hardwires [1], [2]. Given a mother structure, such as a grid of pixels, different antenna features, including operating frequency and radiation pattern, can be obtained by changing the connections between them. Determining the pixel structure that optimizes the design objectives can then be performed using evolutionary computation (EC)-based algorithms and they have been applied to many designs [3], [4]. Pixel antenna design is also a specific example of topology optimization [5]. General topology-based optimization also has good potential for optimizing antennas, however, further research is required to fully demonstrate its effectiveness for antenna design based on conductors [5].

In this communication, we propose a novel heuristic approach for optimizing planar pixel antennas which we refer to as *successive exhaustive Boolean optimization* (SEBO). The key step in SEBO is to cyclically optimize a multivariable binary objective by exhaustively searching a subset of binary variables. An adaptive version of SEBO is also introduced. We demonstrate the technique by providing both simulation and experimental results for optimized antennas operating at several frequency bands with dimensions of 34 mm × 21 mm on a ground plane of 35 mm × 55 mm. Comparisons with designs from EC-based algorithms, *genetic algorithm* (GA), *binary particle swarm optimization* (BPSO) [6], and *angle modulate differential evolution* (AMDE) [7], [8] are also provided which demonstrate that our approach is efficient with up to 50% less computational load while also satisfying the design objectives more consistently. The comparisons also illustrate that our approach is easy to apply as there is only one parameter to be set up while for GA, BPSO, and AMDE several combinations of multiple parameter values need to be tried to find their best operating conditions.

Manuscript received May 1, 2016; revised September 7, 2016; accepted November 2, 2016. Date of publication December 1, 2016; date of current version February 1, 2017. This work was supported by the Hong Kong Research Grants Council under Grant GRF 16207314 and Grant 617911. *(Corresponding author: Shanpu Shen.)*

The authors are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon 999077, Hong Kong (e-mail: sshenaa@ust.hk).

Color versions of one or more of the figures in this communication are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2016.2634399

II. Formulating the Problem

Our focus is on designing a planar pixel antenna consisting of a rectangular grid of $N \times N$ square pixels adjacent to a ground plane. Each pixel can connect to its neighboring pixels or the ground and there are $Q = (2N - 1)N - 1$ possible hardwires. An example for a 5×5 pixel antenna with $Q = 44$ possible hardwires denoted by dashed squares is shown in Fig. 1(a). The presence or absence of hardwires between pixels can be represented by binary digits $x_q \in \{0, 1\}$ $(q = 1, 2, \ldots, Q)$, which are grouped into a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_Q]^T$ to describe the hardwire configuration. There are in total 2^Q combinations of $\boldsymbol{x}$ and various combinations produce distinct antenna characteristics so that antennas with a wide range of possible performance characteristics can be created. The excitation in this example is fixed at the center which is marked as a black arrow in Fig. 1(a) and sits on top of an FR-4 epoxy PCB ($\epsilon_r = 4.4$) having a thickness of 1.6 mm.

To demonstrate SEBO, we focus on finding a hardwire configuration $\boldsymbol{x}$ to match the antenna impedance across a specified frequency band (however, the method is general and other parameters such as patterns can also be specified). The binary optimization problem is then expressed as

$$\min_{\boldsymbol{x}} \sum_{k=1}^{K} R(S_{11}(\boldsymbol{x}, f_k) - T) \quad (1)$$
$$\text{s.t. } \boldsymbol{x} \in \{0, 1\}^Q$$

where $S_{11}(\boldsymbol{x}, f_k)$ denotes the reflection coefficient (dB) at the kth (out of K) frequency sample in the required band with hardwire configuration $\boldsymbol{x}$, T is the required threshold such as −6 dB or −10 dB

0018-926X © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

and $R(r) = \max(0, r)$ denotes the ramp function. The objective function has a minimum value of zero if S_{11} is below the threshold T over the required band and increases to a positive number to penalize the solution when S_{11} gets larger than T.

III. Solving the Problem

For problem (1) we define the *fundamental operation* as computing the value $R(S_{11}(\boldsymbol{x}, f_k)-T)$. We denote the number of objective function evaluations as N_{eval} and the number of fundamental operations calls as N_{fo}. For SEBO, GA, BPSO, and AMDE we have $N_{\text{fo}} = KN_{\text{eval}}$ while for adaptive SEBO we have $N_{\text{fo}} = \sum_a K^{(a)} N_{\text{eval}}^{(a)}$ where $K^{(a)}$ and $N_{\text{eval}}^{(a)}$ denote the number of frequency samples and the number of objective function evaluations in the ath adaptation. To compare the computational complexity of all the algorithms in a unified framework, we consider N_{fo} instead of N_{eval}. It is therefore critical that the optimization algorithm is as efficient as possible to minimize N_{fo}.

Various popular algorithms for tackling binary optimization problems have been tried but were inefficient for solving problem (1). For example, the well-known *branch and bound* (BB) algorithm [9] reduces to exhaustive search for this problem (1) so that BB cannot solve it efficiently. In addition, we could use *semidefinite relaxation* (SDR) [10] to approximate the binary optimization problem (1) as a convex optimization, but SDR also does not perform well since relaxing any of the binary constraints would lead to the design criteria trivially satisfied. The result returned by SDR is completely determined by the initial point, and as a consequence, its performance is the same as random guessing. Therefore, we propose a novel approach denoted by SEBO and also a modified version denoted by adaptive SEBO where K is also adaptively selected to minimize the total computational complexity.

A. Successive Exhaustive Boolean Optimization

SEBO is a heuristic approach for solving the binary optimization problem and is related to the block coordinate descent algorithm [11], [12]. The objective function is optimized at each step using a subset of variables while fixing the value of the other variables.

We first initialize the binary decision vector $\boldsymbol{x}$ to be $\boldsymbol{x}_0$ (a way to find the initial point will be specified later on) and set the number of frequency samples K. After the initialization of $\boldsymbol{x}$, there are two fundamental steps in SEBO. The first step is to perform a cyclic search by partitioning $\boldsymbol{x}$ into $M = \lceil Q/J \rceil$ blocks (subsets) with each block having length $J = J_{\max}$. We set the iteration counter as $t = 1$ and block counter as $b = 1$. At the tth iteration, the bth block of $\boldsymbol{x}$ is treated as a variable and the remaining blocks of $\boldsymbol{x}$ are fixed to be the corresponding part of $\boldsymbol{x}_{t-1}$, hence there are in total $2^{J_{\max}}$ possible combinations for $\boldsymbol{x}_t$. Given that $J_{\max}$ is relatively small, it is possible to exhaustively search all the combinations within this subset. Then $\boldsymbol{x}_t$ is selected as the combination that minimizes the objective function. We upgrade the counter by $t = t + 1$ and $b = b \bmod M+1$ and repeat cyclically this process until convergence.

The second step is to go back and check whether there exists any other local minimum with a smaller objective value, which is related to the randomly initialized hill-climbing algorithm [13], [14]. At the beginning of the second step, J is set to be 1. We randomly pick up J bits of $\boldsymbol{x}$ and optimize them by exhaustive search. After this process is repeated for $\lceil Q/J \rceil$ times, we check whether $\boldsymbol{x}$ has achieved a smaller objective value. If the objective value is improved, the process is repeated for another $\lceil Q/J \rceil$ times, otherwise the block size J is increased by 1 and the procedure is repeated with the new parameters. If $J > J_{\max}$, the algorithm ends. The algorithm will also end if the stop condition is satisfied during the running process. The stop condition is where either the pre-set minimum value of the objective function is achieved or N_{fo} exceeds an allowable value, which we refer to as the maximum computational limit $N_{\text{fo}}^{\max}$. The pseudocode of SEBO is shown in Algorithm 1.

Algorithm 1 SEBO

1) Initialize $\boldsymbol{x} = \boldsymbol{x}_0$. Set the block size $J = J_{\max}$. Partition $\boldsymbol{x}$ into $M = \lceil Q/J \rceil$ blocks (subset). Set the iteration counter as $t = 1$ and block counter as $b = 1$.
2) Treat the bth block of $\boldsymbol{x}$ as variables and fix remaining blocks as the corresponding block of $\boldsymbol{x}_{t-1}$. Exhaustively search all $2^{J_{\max}}$ combinations and find the optimal combinations $\boldsymbol{x}_t$. Upgrade $\boldsymbol{x} = \boldsymbol{x}_t$.
3) Upgrade the counter by $t = t + 1$ and $b = b \bmod M + 1$.
4) Check the stop condition. End the algorithm if satisfied.
5) Check convergence. Go to Step 2 if not converged.
6) Set $J = 1$.
7) Randomly pick up J bits of $\boldsymbol{x}$ and optimize them by exhaustive search. Repeat this process for $\lceil Q/J \rceil$ times.
8) Check the stop condition. End the algorithm if satisfied.
9) Go to Step 7 if $\boldsymbol{x}$ is improved, otherwise $J = J + 1$.
10) End the algorithm if $J > J_{\max}$, otherwise go to Step 7.

The key idea behind SEBO is to optimize the binary variables block by block to reduce the search space, but it may lead to a solution that is not globally optimal. With a larger value of $J_{\max}$, the required number of fundamental operations calls to optimize one block gets larger, but the chance of getting trapped in a local minima gets smaller. Therefore $J_{\max}$ should be carefully chosen to achieve a good fitness result at an affordable computational cost.

B. Adaptive Successive Exhaustive Boolean Optimization

To reduce the computational complexity, we also propose an adaptive SEBO which increases the number of frequency samples K gradually until $S_{11}(f) < T$ for $f \in [f_1, f_2]$. In the 1st adaptation, we initialize $K^{(1)} = 2$ and the frequency samples set as $\mathbf{S}_f = \{f_1, f_2\}$, and use SEBO to find $\boldsymbol{x}^{(1)}$ so that S_{11} at frequency samples $\{f_1, f_2\}$ is no larger than T. The algorithm terminates if no such $\boldsymbol{x}^{(1)}$ can be found, otherwise we will check whether $S_{11}(f) < T$ for $f \in [f_1, f_2]$ is satisfied (the checking is performed by calculating S_{11} at frequency samples spaced as $(f_2 - f_1)/200$.). If satisfied, the design is finished. In the case that both $S_{11}(f_1) < T$ and $S_{11}(f_2) < T$, but there is a range of frequencies, denoted by $f \in \bigcup_{l=1}^{L} [f_{1,l}, f_{2,l}]$, whose S_{11} is larger than T, then, in the 2nd adaptation, the mid-points of $\bigcup_{l=1}^{L} [f_{1,l}, f_{2,l}]$, i.e. $(f_{1,l} + f_{2,l})/2$, are included into the frequency samples (so $K^{(2)} = 2 + L$). We then use SEBO with initial value $\boldsymbol{x}^{(1)}$ to find $\boldsymbol{x}^{(2)}$ so that S_{11} at the new set of frequency samples are no larger than T and again check whether $S_{11}(f) < T$ is satisfied at frequency samples spaced as $(f_2 - f_1)/200$. The adaptation is repeated and the algorithm terminates if the design requirement is satisfied or SEBO fails to find the solution in some adaptation. For the dual band design, the frequency samples are initialized to be $\{f_1, f_2, f_3, f_4\}$, where $[f_1, f_2]$ and $[f_3, f_4]$ are the lower and upper frequency band, respectively. Then we gradually add new frequency samples and repeat the adaptation of SEBO to achieve the design requirement. The pseudocode of adaptive SEBO is shown in Algorithm 2.

C. Internal Multiport Method

The objective function in problem (1) can be computed by any electromagnetic solver, but in this communication we compute the

Algorithm 2 Adaptive SEBO

1) Initialize $K^{(1)} = 2$ and frequency samples set as $\mathbf{S}_f = \{f_1, f_2\}$. Set the counter as $a = 1$ and initial value $\boldsymbol{x}^{(0)}$.
2) In the ath adaptation, use SEBO with initial value $\boldsymbol{x}^{(a-1)}$ to find $\boldsymbol{x}^{(a)}$ to make S_{11} at $\mathbf{S}_f$ no larger than T.
3) End the algorithm if SEBO cannot find such $\boldsymbol{x}^{(a)}$ within $N_{\text{fo}}^{\max}$ (end if $N_{\text{fo}} > N_{\text{fo}}^{\max}$). Otherwise check whether $S_{11}(f) < T$ for $f \in [f_1, f_2]$ is satisfied.
4) End the algorithm if satisfied. Otherwise find the set $\bigcup_{l=1}^{L} [f_{1,l}, f_{2,l}]$ whose S_{11} is larger than T.
5) Upgrade $K^{(a)} = K^{(a-1)} + L$, $a = a + 1$, and $\mathbf{S}_f$ by adding the mid-points $(f_{1,l} + f_{2,l})/2$. Go to Step 2.

objective function efficiently by a technique we refer to as *internal multiport method* (IMPM). IMPM makes use of a technique described in [15] which determined the effect of shorting strap switches on a patch antennas' input impedance and has been extended to pixel antennas [16]. IMPM is particularly well suited for use with pixel antennas as a standard commercial electromagnetic solver can be used in such a way that computational complexity is reduced significantly [16]. IMPM can also be adapted to calculate field patterns and other parameters from pixel antennas efficiently. More details can be found in [16] on its application to pixel antennas.

D. Evolutionary Computation-Based Algorithms

Evolutionary Computation (EC)-based algorithms, such as genetic algorithms (GA) [17]–[20], particle swarm optimization (PSO) [6], [21]–[23], and differential evolution (DE) [24]–[27] are popular for optimizing objective functions which are highly nonlinear, stiff, multiextremal, nondifferentiable, and computationally expensive to evaluate. Advantages of these methods include flexibility and adaptability to the task at hand, robust performance, and global search characteristics [28].

In this communication, we also use EC-based algorithms, GA, BPSO, and AMDE, to solve the binary optimization problem (1) for comparison with SEBO and adaptive SEBO. GA can be used to solve both binary or continuous variable optimization problem, but the standard PSO and DE are designed to solve continuous variable optimization problem. Therefore, we use the modified binary versions of PSO and DE, BPSO, and AMDE, to solve this problem (1).

E. Algorithm Comparison

Since SEBO, adaptive SEBO, and EC-based algorithms are stochastic, we run these algorithms for D times and compare their average performance for solving the problem (1). All the five algorithms end as soon as $N_{\text{fo}} > N_{\text{fo}}^{\max}$ or they meet the requirement that $S_{11}(f) < T$ for $f \in [f_1, f_2]$. In the ith ($i = 1, 2, \ldots, D$) optimization, we denote the number of fundamental operations calls as N_{fo}^{i} and we can find the bandwidth coverage, denoted as B^i, by computing the ratio of bandwidth achieved by optimization and the required bandwidth. We can also count the number of full bandwidth coverage ($B^i = 100\%$) and denote it as N_{full}. We mainly consider the *average number of fundamental operations calls* $N_{\text{avg}} = (\sum_i N_{\text{fo}}^{i})/D$, the *average bandwidth coverage* $B_{\text{avg}} = (\sum_i B^i)/D$, and the *frequency of full bandwidth coverage* $F_{\text{full}} = N_{\text{full}}/D$ to compare the algorithms. These quantities are evaluated in two examples in the next section for algorithm comparison.

IV. Design Examples

We demonstrate SEBO and adaptive SEBO with two examples for wireless communication applications. The first is matching across a single band and the second across dual bands. We set the required threshold $T = -6$ dB. We focus on optimizing the 5×5 pixel antenna and its multiport structure shown in Fig. 1(b) is first modeled in a commercial method of moment (MoM) solver HyperLynx 3-D electromagnetic (IE3D) [29] to generate the 45×45 impedance matrix which is used to compute $S_{11}(f)$ in IMPM. As a comparison, we also use GA, BPSO, and AMDE to solve the binary optimization problem (1). We use the built-in GA function in the Global Optimization Toolbox in MATLAB while we implement SEBO, adaptive SEBO, BPSO, and AMDE in MATLAB ourselves. The initial point for SEBO and adaptive SEBO is selected by using a random number generator. In GA, BPSO, and AMDE, the population size and generation size are denoted as P and G, respectively, and we have $N_{\text{fo}}^{\max} = P(G+1)K$. In the following examples, we will evaluate the performance of GA, BPSO, and AMDE with different combinations of P and G within $N_{\text{fo}}^{\max}$. Furthermore, we also evaluate the performance of BPSO and AMDE with different combinations of algorithm parameters. For BPSO, its algorithm parameters are *inertial weight* ω and *Hooke's constants* c_1 and c_2, which control the update of the velocity of particles $\mathbf{V}_t$ by

$$\mathbf{V}_t = \omega \mathbf{V}_{t-1} + c_1 \eta_1 (\mathbf{P}_{t-1} - \mathbf{X}_{t-1}) + c_2 \eta_2 (\mathbf{G}_{t-1} - \mathbf{X}_{t-1}) \quad (2)$$

where $\mathbf{X}_t$ denotes the position of particles, $\mathbf{P}_{t-1}$ denotes the *personal best*, $\mathbf{G}_{t-1}$ denotes the *global best*, and η_1 and η_2 have a uniform distribution in (0, 1) [6]. For AMDE, its algorithm parameters are *crossover rate* p_r and *scaling factor* γ, which control the generation of new offspring O_n by

$$O_{n,j} = \begin{cases} C_{n_3,j} + \gamma (C_{n_1,j} - C_{n_2,j}) & \text{if } \eta \le p_r \text{ or } j = i \\ C_{n,j} & \text{otherwise} \end{cases} \quad (3)$$

where C_{n_1}, C_{n_2}, and C_{n_3} are randomly selected three parent individuals with $n_1 \neq n_2 \neq n_3 \neq n$, η has a uniform distribution in (0, 1), and i is randomly selected from the set $\{1, 2, \ldots, I\}$ with I being the dimension of individuals [7], [8]. For SEBO and adaptive SEBO, we evaluate the performance with different values of $J_{\max}$ to study its effect on performance and computational load. We run all algorithms with all combinations for $D = 100$ times to find B_{avg}, N_{avg}, and F_{full}.

A. Single Frequency Band Antennas

In this example, we design a pixel antenna for operation in the GSM-850 band (824–894 MHz) using SEBO, adaptive SEBO, GA, BPSO, and AMDE. We set $N_{\text{fo}}^{\max} = 3 \times 10^5$. We also set $K = 6$ for SEBO, GA, BPSO, and AMDE and the results are shown in Table I. For GA, we evaluate its performance with 8 combinations of P and G and we can find that its performance fluctuates with different values of P and G and the most effective operating point (maximum B_{avg} and F_{full}) is when $P = 2500$, $G = 19$. For BPSO, we set the range of particle velocity $[V_{\min}, V_{\max}] = [-4, 4]$ and adopt the time-varying inertial weight ω, which linearly varies from 1 to 0.4 with the iteration, to accelerate convergence. Therefore, we evaluate the performance of BPSO with 648 combinations of P, G, c_1, and c_2 (the values of c_1 and c_2 are selected from the set $\{0.4, 0.6, \ldots, 2\}$ and for each combination of P and G we only show the result with the best c_1 and c_2 in Table I). We can see that for different values of P and G, the optimal combination of c_1 and c_2 and the corresponding performance are also different. Without exhaustively trying all the 648 combinations, it is difficult to find the optimal algorithm parameter settings which are $P = 125$, $G = 399, c_1 = 1.8, c_2 = 1.8$ in this example. For AMDE, we evaluate its performance with 720 combinations of P, G, p_r,

TABLE I
COMPARISON RESULT FOR GSM-850 BAND

	B_{avg}	N_{avg}	F_{full}
SEBO ($J_{\max} = 6$)	98.83%	137360	73%
SEBO ($J_{\max} = 7$)	98.51%	149838	72%
SEBO ($J_{\max} = 8$)	98.99%	118854	80%
SEBO (average over all combinations)	98.78%	135351	75%
Adaptive SEBO ($J_{\max} = 6$)	99.87%	61278	95%
Adaptive SEBO ($J_{\max} = 7$)	99.47%	99899	83%
Adaptive SEBO ($J_{\max} = 8$)	99.84%	55376	97%
Adaptive SEBO (average over all combinations)	99.73%	72184	92%
GA ($P = 5000,\ G = 9$)	97.24%	274800	39%
GA ($P = 2500,\ G = 19$)	98.06%	228000	59%
GA ($P = 1000,\ G = 49$)	97.61%	192600	53%
GA ($P = 500,\ G = 99$)	95.93%	216570	37%
GA ($P = 250,\ G = 199$)	94.29%	215160	32%
GA ($P = 125,\ G = 399$)	91.74%	248490	22%
GA ($P = 100,\ G = 499$)	89.56%	275556	10%
GA ($P = 50,\ G = 999$)	90.64%	269406	15%
GA (average over all combinations)	94.38%	240073	33%
BPSO ($P = 5000, G = 9, c_1=0.4, c_2=1.4$)	91.76%	288000	13%
BPSO ($P = 2500, G = 19, c_1=1.2, c_2=1.4$)	92.33%	272500	20%
BPSO ($P = 1000, G = 49, c_1=1.6, c_2=2$)	96.14%	262400	30%
BPSO ($P = 500, G = 99, c_1=1, c_2=1.8$)	96.62%	217500	37%
BPSO ($P = 250, G = 199, c_1=1.8, c_2=1.8$)	97.43%	184950	50%
BPSO ($P = 125, G = 399, c_1=1.8, c_2=1.8$)	98.43%	161275	60%
BPSO ($P = 100, G = 499, c_1=1.8, c_2=2$)	98.24%	171580	53%
BPSO ($P = 50, G = 999, c_1=0.4, c_2=2$)	97.16%	176130	53%
BPSO (average over all combinations)	93.60%	265710	16%
AMDE ($P = 5000, G = 9, p_r=0.1, \gamma=1.8$)	88.09%	295200	2%
AMDE ($P = 2500, G = 19, p_r=0.6, \gamma=1.8$)	88.76%	295000	3%
AMDE ($P = 1000, G = 49, p_r=0.3, \gamma=1.6$)	91.21%	295500	5%
AMDE ($P = 500, G = 99, p_r=0.5, \gamma=0.8$)	91.36%	296250	5%
AMDE ($P = 250, G = 199, p_r=0.6, \gamma=1$)	91.86%	296100	5%
AMDE ($P = 125, G = 399, p_r=0.6, \gamma=1.2$)	91.36%	279040	10%
AMDE ($P = 100, G = 499, p_r=0.6, \gamma=0.4$)	92.43%	290940	5%
AMDE ($P = 50, G = 999, p_r=0.3, \gamma=0.2$)	92.50%	284175	10%
AMDE (average over all combinations)	88.56%	299490	0.4%

and γ (the values of p_r is selected from the set $\{0.1, 0.2, \ldots, 0.9\}$, the values of γ is selected from the set $\{0.2, 0.4, \ldots, 2\}$, and for each combination of P and G we again only show the result with the best p_r and γ in Table I). Similar to BPSO, the optimal combination of p_r and γ and the corresponding performance are changed with the different values of P and G, and the optimal algorithms parameters setting among all the 720 combinations are $P = 50, G = 999$, $p_r = 0.3, \gamma = 0.2$. For SEBO with all $J_{\max}$ considered, B_{avg} and F_{full} are better than those of GA, BPSO, and AMDE and N_{avg} is less than that of GA, BPSO, and AMDE. Therefore for the single band design problem, we find that SEBO can achieve a better result than GA, BPSO, and AMDE with lower computational complexity. Furthermore, for adaptive SEBO, B_{avg} and F_{full} can be improved and the N_{avg} can be further decreased. Therefore adaptive SEBO achieves the best result in the sense that on average it obtains better coverage of the required bandwidth at 65% less computational cost than GA, BPSO, and AMDE. Furthermore, for SEBO and adaptive SEBO there is only one parameter $J_{\max}$, but for GA, BPSO, and AMDE, there are several algorithm parameters such as P, G, c_1, c_2, p_r, and γ. In reality, it is difficult to find the best settings for these parameters without exhaustively trying all the combinations. We also compute the average B_{avg}, N_{avg}, and F_{full} over all combinations of the parameters listed above for each algorithms and list them as an additional row after the results for each algorithm in Table I. We can see that our proposed SEBO and adaptive have much better average performance than EC-based algorithms in this comparison. Hence, we find that SEBO and adaptive SEBO are straightforward to apply

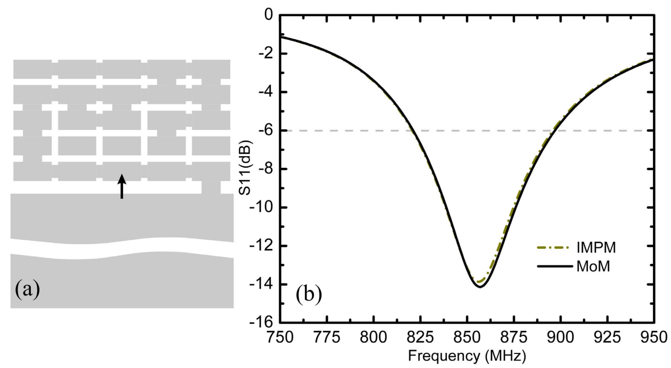

Fig. 2. (a) Optimal hardwire configuration and (b) reflection coefficient of the proposed antenna matched across GSM-850 band.

because they have only one parameter to set and they are also more computationally efficient.

In this single band example, we also provide the optimum hardwire configuration for the 5×5 pixel antenna, which is obtained by adaptive SEBO with $J_{\max} = 8$, and the corresponding S_{11} in Fig. 2. It can be observed that a meander like IFA structure has been formed. The curves of S_{11} for both IMPM and MoM are also provided and it can be seen that they are similar in Fig. 2, demonstrating the effectiveness of IMPM. Simulated S_{11} has a -6 dB bandwidth from 822 to 895 MHz, satisfying the design requirement.

B. Dual Band Antennas

Similar to [16], we also design a dual-band antenna matched to GSM-900 and DCS + PCS + UMTS bands (890–960 MHz and 1710–2170 MHz). We use SEBO, adaptive SEBO, GA, BPSO, and AMDE to solve the dual band problem and set $N_{\text{fo}}^{\max} = 3 \times 10^5$. For SEBO, GA, BPSO, and AMDE, we set $K = 12$ (six samples for each band). Similar to the previous example, we use various combinations of P and G in GA, various combination of P, G, c_1, and c_2 in BPSO, various combinations of P, G, p_r, and γ in AMDE, and various $J_{\max}$ in SEBO and adaptive SEBO. The results are shown in Table II and for all combinations of algorithm parameters shown, F_{full} is larger than 60%, N_{avg} is much less than that of single band design, and B_{avg} is larger than 97%, which indicates that this particular dual band design problem is easier than the single band design. This can be explained by noting that the lowest frequency of the dual band design is higher than that of the single band design. Similar to the single band design example, we find that the performance of GA, BPSO, and AMDE are changed with the different combinations of P, G, c_1, c_2, p_r, and γ and the optimal algorithm parameters (c_1, c_2, p_r, and γ) for BPSO and AMDE are also changed with P and G. This further demonstrates that it is difficult to find the optimal algorithm parameter settings without exhaustively trying numerous combinations. From Table II, we can find that SEBO with three different values of $J_{\max}$ has less N_{avg} than all the combinations of AMDE and all the combinations of GA except the optimal combination $P = 250$, $G = 99$, but more N_{avg} than half of the combinations of BPSO. Therefore, for this dual band design problem, SEBO is more efficient than GA and AMDE but less efficient than BPSO. For adaptive SEBO, we see that all the B_{avg} and F_{full} are 100% and that N_{avg} for the three different values of $J_{\max}$ are less than those of GA, AMDE, and SEBO (at least 50% less computation), and similar to those of BPSO when it has optimal combinations of parameters. However, the optimal BPSO result is achieved by trying 648 parameter combinations while the optimal adaptive SEBO result is only achieved by trying three

TABLE II
COMPARISON RESULT FOR GSM-900 AND DCS + PCS + UMTS BANDS

	B_{avg}	N_{avg}	F_{full}
SEBO ($J_{\max} = 6$)	98.09%	37584	94%
SEBO ($J_{\max} = 7$)	99.80%	44076	97%
SEBO ($J_{\max} = 8$)	98.28%	45264	92%
SEBO (average over all combinations)	98.72%	42308	94%
Adaptive SEBO ($J_{\max} = 6$)	100%	16875	100%
Adaptive SEBO ($J_{\max} = 7$)	100%	24879	100%
Adaptive SEBO ($J_{\max} = 8$)	100%	28710	100%
Adaptive SEBO (average over all combinations)	100%	23488	100%
GA ($P = 2500,\ G = 9$)	99.81%	106800	98%
GA ($P = 1000,\ G = 24$)	99.41%	62400	98%
GA ($P = 500,\ G = 49$)	99.52%	46020	98%
GA ($P = 250,\ G = 99$)	99.45%	35640	95%
GA ($P = 125,\ G = 199$)	98.90%	74856	87%
GA ($P = 100,\ G = 249$)	99.80%	53196	93%
GA ($P = 50,\ G = 499$)	97.33%	86832	79%
GA ($P = 25,\ G = 999$)	97.40%	137208	70%
GA (average over all combinations)	98.95%	75369	90%
BPSO ($P = 2500, G = 9, c_1$=0.4,c_2=1.8)	99.58%	144600	93%
BPSO ($P = 1000, G = 24, c_1$=1,c_2=2)	99.97%	83640	99%
BPSO ($P = 500, G = 49, c_1$=1,c_2=2)	99.82%	58140	96%
BPSO ($P = 250, G = 99, c_1$=1.6,c_2=1.8)	100%	39660	100%
BPSO ($P = 125, G = 199, c_1$=1,c_2=1.8)	99.82%	25200	95%
BPSO ($P = 100, G = 249, c_1$=0.6,c_2=2)	99.90%	22152	97%
BPSO ($P = 50, G = 499, c_1$=1.4,c_2=2)	100%	16572	100%
BPSO ($P = 25, G = 999, c_1$=1.8,c_2=2)	100%	12798	100%
BPSO (average over all combinations)	99.32%	72986	95%
AMDE ($P = 2500, G = 9, p_r$ =0.8,γ=0.6)	97.51%	238000	60%
AMDE ($P = 1000, G = 24, p_r$ =0.3,γ=0.6)	97.47%	261200	67%
AMDE ($P = 500, G = 49, p_r$ =0.7,γ=0.2)	98.02%	218600	77%
AMDE ($P = 250, G = 99, p_r$ =0.7,γ=0.2)	98.31%	201300	73%
AMDE ($P = 125, G = 199, p_r$ =0.1,γ=0.2)	99.11%	193400	87%
AMDE ($P = 100, G = 249, p_r$ =0.9,γ=0.6)	99.03%	167240	83%
AMDE ($P = 50, G = 499, p_r$ =0.2,γ=0.2)	99.47%	187340	87%
AMDE ($P = 25, G = 999, p_r$ =0.5,γ=0.2)	99.81%	143310	96%
AMDE (average over all combinations)	94.85%	224850	49%

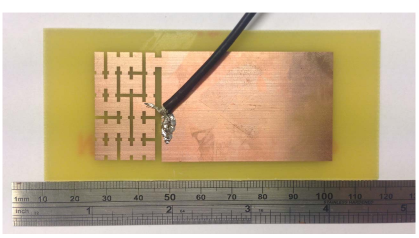

Fig. 3. Photo of the prototype.

different values of $J_{\max}$. We also compute the average B_{avg}, N_{avg}, and F_{full} over all combinations of the parameters listed above for each algorithm and list them as additional rows in Table II. Again, the better average performance illustrates that SEBO and adaptive SEBO are easy to apply and once the parameter settings are taken into account is more computationally efficient.

For the optimum solution found by adaptive SEBO with $J_{\max} = 6$, a prototype was also fabricated to verify the design and the prototype is shown in Fig. 3. The corresponding hardwire configuration for the 5×5 pixel antenna and the S_{11} from MoM and measurement are shown in Fig. 4. The S_{11} from IMPM is omitted because it is similar to the S_{11} obtained by MoM. The S_{11} measured by a vector network analyzer has -6 dB bandwidth from 900 to 1030 MHz and from 1700 to more than 2290 MHz. The measured S_{11} shifts to higher frequency compared with simulation because of the imperfect in-house fabrication and dielectric models used in the simulation which can be improved by further fine tuning. We also used the Satimo Starlab system to measure the radiation patterns of the

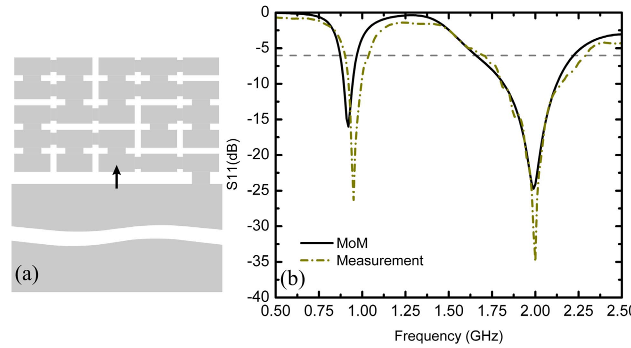

Fig. 4. (a) Optimal hardwire configuration and (b) reflection coefficient of the proposed antenna matched across GSM-900 and DCS + PCS + UMTS.

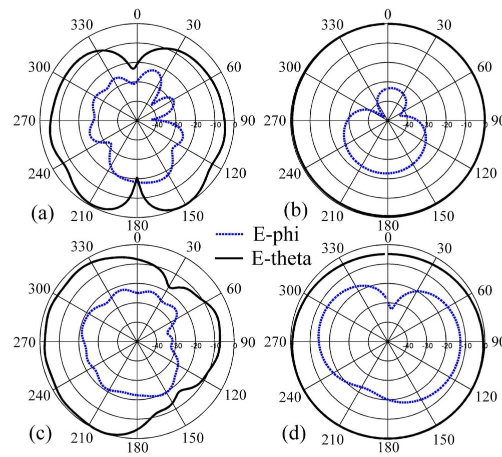

Fig. 5. Measured radiation patterns (a) xz plane at 1 GHz, (b) xy plane at 1 GHz, (c) xz plane at 2 GHz, and (d) xy plane at 2 GHz.

prototype at 1 and 2 GHz, which are shown in Fig. 5 for reference (coordinates are shown in Fig. 1).

V. CONCLUSION

We have proposed a new heuristic method for optimizing the design of pixel antennas. Two versions of the method, namely, SEBO and adaptive SEBO, are described and the performance of the proposed algorithms are evaluated by two examples in which we perform impedance matching across single and dual bands. In the single band example, the results indicate that SEBO and adaptive SEBO achieve the best performance compared with GA, BPSO, and AMDE and reduces the computational load by more than 65% while achieving broader bandwidth coverage. In the dual band example, the results indicate that adaptive SEBO achieves better performance than SEBO, GA, and AMDE with less than 50% computation and is nearly as efficient as BPSO when BPSO has the optimal settings for its parameters. A major advantage of SEBO and adaptive SEBO is that only one parameter needs to be set up for the optimization while for evolutionary optimization two or more parameters usually need to be set up. Thus SEBO and adaptive SEBO are not only computationally efficient they are also easy to apply to pixel antenna design.

REFERENCES

[1] B. A. Cetiner, H. Jafarkhani, J.-Y. Qian, H. J. Yoo, A. Grau, and F. D. Flaviis, "Multifunctional reconfigurable MEMS integrated antennas for adaptive MIMO systems," *IEEE Commun. Mag.*, vol. 42, no. 12, pp. 62–70, Dec. 2004.

[2] C. Y. Chiu, J. Li, S. Song, and R. D. Murch, "Frequency-reconfigurable pixel slot antenna," *IEEE Trans. Antennas Propag.*, vol. 60, no. 10, pp. 4921–4924, Oct. 2012.

[3] L. N. Pringle *et al.*, "A reconfigurable aperture antenna based on switched links between electrically small metallic patches," *IEEE Trans. Antennas Propag.*, vol. 52, no. 6, pp. 1434–1445, Jun. 2004.

[4] Z. Li, D. Rodrigo, L. Jofre, and B. A. Cetiner, "A new class of antenna array with a reconfigurable element factor," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 1947–1955, Apr. 2013.

[5] T. Nomura, M. Ohkado, P. Schmalenberg, J. Lee, O. Ahmed, and M. Bakr, "Topology optimization method for microstrips using boundary condition representation and adjoint analysis," in *Proc. IEEE Eur. Microw. Conf. (EuMC)*, Oct. 2013, pp. 632–635.

[6] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 556–567, Mar. 2007.

[7] G. Pampará, A. P. Engelbrecht, and N. Franken, "Binary differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2006, pp. 1873–1879.

[8] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[9] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960. [Online]. Available: http://www.jstor.org/stable/1910129

[10] X. Fan, J. Song, D. P. Palomar, and O. C. Au, "Universal binary semidefinite relaxation for ML signal detection," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4565–4576, Nov. 2013.

[11] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[12] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Math. Program.*, vol. 144, no. 1, pp. 1–38, 2014.

[13] A. Fallahi, M. Mishrikey, C. Hafner, and R. Vahldieck, "Efficient procedures for the optimization of frequency selective surfaces," *IEEE Trans. Antennas Propag.*, vol. 56, no. 5, pp. 1340–1349, May 2008.

[14] C. Hafner, C. Xudong, J. Smajic, and R. Vahldieck, "Efficient procedures for the optimization of defects in photonic crystal structures," *J. Opt. Soc. Amer. A*, vol. 24, no. 4, pp. 1177–1188, 2007.

[15] J. L. A. Quijano and G. Vecchi, "Optimization of an innovative type of compact frequency-reconfigurable antenna," *IEEE Trans. Antennas Propag.*, vol. 57, no. 1, pp. 9–18, Jan. 2009.

[16] S. Song and R. D. Murch, "An efficient approach for optimizing frequency reconfigurable pixel antennas using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 62, no. 2, pp. 609–620, Feb. 2014.

[17] J. M. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 39, no. 4, pp. 7–21, Aug. 1997.

[18] E. A. Jones and W. T. Joines, "Design of Yagi–Uda antennas using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 45, no. 9, pp. 1386–1392, Sep. 1997.

[19] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 42, no. 7, pp. 993–999, Jul. 1994.

[20] F. J. Villegas, T. Cwik, Y. Rahmat-Samii, and M. Manteghi, "A parallel electromagnetic genetic-algorithm optimization (EGO) application for patch antenna design," *IEEE Trans. Antennas Propag.*, vol. 52, no. 9, pp. 2424–2435, Sep. 2004.

[21] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp. 397–407, Feb. 2004.

[22] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2674–2679, Aug. 2005.

[23] F. Afshinmanesh, A. Marandi, and M. Shahabadi, "Design of a single-feed dual-band dual-polarized printed microstrip antenna using a Boolean particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 56, no. 7, pp. 1845–1852, Jul. 2008.

[24] P. Rocca, G. Oliveri, and A. Massa, "Differential evolution as applied to electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 53, no. 1, pp. 38–49, Feb. 2011.

[25] D. G. Kurup, M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2210–2217, Sep. 2003.

[26] S. Yang, Y. B. Gan, and A. Qing, "Sideband suppression in time-modulated linear arrays by the differential evolution algorithm," *IEEE Antennas Wireless Propag. Lett.*, vol. 1, no. 1, pp. 173–175, 2002.

[27] S. Caorsi, A. Massa, M. Pastorino, and A. Randazzo, "Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method," *IEEE Trans. Antennas Propag.*, vol. 53, no. 1, pp. 372–376, Jan. 2005.

[28] T. Back, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, Apr. 1997.

[29] Mentor Graphics. (2010). *Hyperlynx 3D EM, Ver. 15.00*. [Online]. Available: http://www.mentor.com/