Article
Constrained Planar Array Thinning Based on Discrete Particle Swarm Optimization with Hybrid Search Strategies

Wanhan Cai, Lixia Ji, Chenglin Guo, Ke Mei and Hao Zeng *

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
* Correspondence: haoz@cqu.edu.cn

Abstract: This article presents a novel optimization algorithm for large array thinning. The algorithm is based on Discrete Particle Swarm Optimization (DPSO) integrated with some different search strategies. It utilizes a global learning strategy to improve the diversity of populations at the early stage of optimization. A dispersive solution set and the gravitational search algorithm are used during particle velocity updating. Then, a local search strategy is enabled in the later stage of optimization. The particle position is adaptively adjusted by the mutation probability, and its motion state is monitored by two observation parameters. The peak side-lobe level (PSLL) performance, effectiveness and robustness of the improved PSO algorithm are verified by several representative examples.

Keywords: array thinning; search strategy; peak side-lobe level; particle swarm optimization

1. Introduction

Thinned arrays have been the focus of research in recent years for their lower cost, lower energy consumption and lighter weight compared with the conventional uniform arrays. The main purpose of array thinning is to obtain a lower peak side-lobe level (PSLL) on the condition that the antenna array satisfies gain demand. Planar array thinning can be achieved by adjusting the “ON” or “OFF” states of each element in a uniform array.

To suppress the PSLL, several optimization methods have been proposed. As suggested by Liu in [1], the thinning of a reconfigurable linear array can be reduced by matrix beam method with high efficiency. RF switching technology in the T/R module makes the application of adaptive sparse technology possible [2]. Rahardjo et al. [3] developed a new method for designing linear thinned arrays using spacing coefficients based on Taylor line source distribution, which is verified to be able to correctly match different beam pattern configurations without repeated global optimization. Recently, analytical thinning methods by convex optimization [4] and Bayesian compressive sensing [5] have been introduced, but these methods need to set a suitable reference pattern as a precondition. Keizer [6] proposed the iterative Fourier technique (IFT) for array thinning. He calculates the array factor and excitation of the uniform array by making use of the Fourier transforms. Moreover, the IFT method is also used to optimize large planar arrays with higher convergence speed [7].

Benefitting from excellent global search performance, some intelligent optimization algorithms, such as the real genetic algorithm (RGA) [8] and asymmetric mapping method with differential evolution (DE) [9], also play a good role in array thinning. For this problem, analyses are also carried out with the newly introduced Honey Badger Algorithm (HBA) and Chameleon Swarm Algorithm (CSA) algorithms in [10]. Vankayalapati introduced a new algorithm called multi-objective modified binary cat swarm optimization to deal with the optimization of multiple contradicting parameters [11].

First proposed by Eberhart and Kennedy in 1995 [12], particle swarm optimization (PSO) is a new optimization algorithm for solving continuous and discontinuous problems in multiple dimensions. With a variety of analytical and numerical tools available, PSO has been applied to different application domains such as antenna pattern synthesis, array thinning, and sensor networks. The PSO and its improved algorithms are employed
frequently for suit ing multiple objectives simultaneously. Random drift particle swarm optimization (RDPSO), which is used to solve the electric power economic scheduling problem, has been applied to the optimal design of thinned array in [13], achieved good results. To compare the performance of single-ring planar antenna arrays, Bera et al. [14] proposed a novel wavelet mutation-based novel particle swarm optimization (NPSOWM). By applying the binary particle swarm optimization (BPSO) algorithm to the total focusing method (TFM) [15] for thinned array design, the simulation results indicate that the proposed TFM can greatly increase computational efficiency and provide significantly higher image quality. The PSO study in [16] aims to generate multiple-focus patterns and a large scanning range for random arrays thinning, which is applied to the ultrasound treatment of brain tumors and neuromodulation. A search mode with multi-objective particle swarm optimization was proposed in [17] to solve the problem of optimal array distribution, which met the requirements of reducing the number of antenna elements and maintaining the PSLL simultaneously. In 2021, Guo et al. [18] studied the SLL reduction optimization of linear antenna arrays (LAA) with specific radiation characteristics and circular antenna arrays (CAA) with isotropic radiation characteristics. In the optimization process, an optimization method (GWO-PSO) combining gray wolf optimization and PSO will be used.

In general, in order to avoid the rapid loss of particle distribution in solution space, the existing PSO methods give a variety of evaluation strategies for the diversity of particle population distribution. However, these methods mainly focus on improving the algorithm efficiency and ignore the balance between global search and local search, resulting in the lack of ability to adjust the search focus dynamically in different search stages.

In this paper, a new novel optimization algorithm for large array thinning is proposed. The innovative part of this algorithm is the combined usage of different particle learning strategies with discrete particle swarm optimization, which enhances the ability of global search and effectiveness of the algorithm.

The rest of this paper is organized as follows: The planar array structure and optimization problem model are briefly outlined in Section 2. Section 3 introduces the DPSO algorithm and our improvement to it. Several simulation results and discussions are presented in Section 4. Finally, a brief conclusion is given in Section 5.

2. Optimization Model

Assume a large planar array with elements arranged in square grids with a spacing of \( d \) along \( M \) columns and \( N \) rows, as shown in Figure 1.
Matrices $A$ and $B$ are set as:

$$A = \begin{pmatrix}
a_{00} & a_{01} & \cdots & a_{0N-1} \\
a_{10} & a_{11} & \cdots & a_{1N-1} \\
\vdots & \vdots & \ddots & \vdots \\
a_{M-10} & a_{M-11} & \cdots & a_{M-1N-1}
\end{pmatrix}$$

(1)

$$B = \begin{pmatrix}
b_{00} & b_{01} & \cdots & b_{0N-1} \\
b_{10} & b_{11} & \cdots & b_{1N-1} \\
\vdots & \vdots & \ddots & \vdots \\
b_{M-10} & b_{M-11} & \cdots & b_{M-1N-1}
\end{pmatrix}$$

(2)

where $a_{mn}$ and $b_{mn}$ are the excitation of element $(m, n)$ and the “ON” or “OFF” state of the element $(m, n)$, respectively. So, $b_{mn}$ is 1 or 0. In Figure 1, $\theta$ and $\phi$ are the elevation and the azimuth angles in the spherical coordinate, respectively. $u$ and $v$ are direction cosines defined by $u = \sin \theta \cos \phi$ and $v = \sin \theta \sin \phi$. If $(u_0, v_0)$ is the desired main beam direction, the radiation beam pattern $F(u, v)$ can be expressed as:

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn}b_{mn}e^{jk\left[n\theta(u-u_0)+m\theta(v-v_0)\right]}$$

(3)

The fitness function considered in the present study is the PSLL of the radiation beam pattern, which is desired to be as low as possible. The PSLL of the planar array can be computed as:

$$\text{PSLL} = \max_{(u,v) \in S} \left[20\log \left| \frac{F(u,v;u_0,v_0)}{F(u_0,v_0)} \right| \right]$$

(4)

where $S$ denotes the angular region excluding the main beam. Considering the constraint of the array filling rate, the sum of all elements in matrix $B$ should be a definite constant. If the aperture remains the same, the four corner elements of the planar array must be “ON”. The model of optimization can be represented as:

$$\min_{B} \text{PSLL}$$

s.t. $0 \leq m \leq M - 1, 0 \leq n \leq N - 1$

$$b_{00} = 1, b_{M-10} = 1$$

$$b_{0N-1} = 1, b_{M-1N-1} = 1$$

$$\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} b_{mn} = K$$

(5)

where $K$ denotes the number of elements that turned “ON”.

### 3. Improved Algorithm

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

#### 3.1. Fundamental PSO Algorithm

The fundamental PSO algorithm assumes a swarm of particles in the solution space, and the positions of these particles indicate possible solutions to the variables defined for a specific optimization problem. The particles move in directions based on update equations impacted by their own local best positions and the global best position of the entire swarm.

Assume a swarm composed of $N_P$ particles is uniformly dispersed in a $D$-dimensional solution space, where the position $x_i$ and velocity $v_i$ of $i$th particle can be expressed as:

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$$

(6)
\[
v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})
\] (7)

The velocity update equation is given below:
\[
v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 r_1(t) [pBest_{ij}(t) - x_{ij}(t)] + c_2 r_2(t) [gBest_j(t) - x_{ij}(t)]
\] (8)

where \(i = 1, 2, \ldots, NP\), \(j = 1, 2, \ldots, D\). Obviously, \(NP\) represents the number of particles, and \(D\) represents the number of dimensions. \(c_1\) and \(c_2\) are the acceleration constants, and \(r_1\) and \(r_2\) are two random numbers within the range \([0,1]\). \(v_{ij}(t)\) and \(x_{ij}(t)\) are the velocity and position along the \(j^{th}\) dimension for the \(i^{th}\) particle at \(t^{th}\) iteration, respectively. \(pBest_{ij}(t)\) is the best position along the \(j^{th}\) dimension for the \(i^{th}\) particle at \(t^{th}\) iteration, respectively. \(gBest_j(t)\) is the best position found by the swarm along the \(j^{th}\) dimension at \(t^{th}\) iteration.

The value of a position along each dimension for each particle is limited to “0” and “1” in discrete algorithms. The position update equation along the \(j^{th}\) dimension for the \(i^{th}\) particle is given by
\[
s(v_{ij}) = \frac{1}{1 + e^{-v_{ij}}}
\] (9)
\[
x_{ij}(t+1) = \begin{cases} 1, & r < s(v_{ij}) \\ 0, & \text{others} \end{cases}
\] (10)

where \(s(v_{ij})\) denotes a function that maps the particle velocity to the probability of the particle position, and \(r\) represents a random number within the range \([0,1]\).

### 3.2. DPSO with Hybrid Search Strategies

The improved PSO algorithm proposed in this paper consists of three main strategies, which are the global learning strategy based on niche technique, dispersed solution sets, the local mutation search strategy and the motion state monitoring strategy. The execution of the corresponding strategies is adaptively adjusted in different search stages of the optimization.

#### 3.2.1. Global Learning Strategy

It is necessary to maintain as high a swarm diversity as possible in the early stage of optimization to avoid premature convergence. So, two strategies are utilized in the early stage, which are the niche technique and the gravitational search algorithm.

The niche technique proposed in [19] can form multiple stable subpopulations within a population. Each particle interacts only with its immediate neighbors through a ring topology. Obviously, the ring topology structure can promote individuals to search thoroughly in its local neighborhood before global optimization. Therefore, it is acceptable for discovering multiple optima. On the basis of the ring topology proposed by [19], a high-quality solution set \(P_G\) is proposed in this paper, as shown in Figure 2. The solution set \(P_G\) consists of two kinds. One is the optimal solutions of all particles, the other is the eliminated optimal solutions of some particles with excellent fitness function values. Preserving the possibility of interaction between each particle and its neighbors, particles can also learn from \(P_G\) directly. This structure will substitute for the role of the global best position \(gBest\) in (8).

Because the positions of elements in \(P_G\) may be uniformly distributed in the whole solution space, particles under the guidance of elements in \(P_G\) have a considerable number of potential motion directions, which improves the diversity of the population, and avoids premature convergence.
3.2. DPSO with Hybrid Search Strategies

The improved PSO algorithm, where a particle is impacted by the combined gravity of all the other particles in the solution space. We use a GSA-based acceleration to replace the personal best position $pBest_i$ in (8) to enhance the correlation between particles.

At the $i^{th}$ iteration, the gravitational attraction of particle $q$ on particle $i$ in the $k^{th}$ dimension can be defined as:

$$F_{iq}^k(t) = G(t) \frac{M_i(t) \times M_q(t)}{R_{iq} + \varepsilon} \left[ x_q^k(t) - x_i^k(t) \right]$$  \hspace{1cm} (11)

where $M_q(t)$ represents the inertial mass of the particle applying force, $M_i(t)$ represents the inertial mass of the particle subjected to the force, $R_{iq} = || x_i(t) - x_q(t) ||$, $R_{iq}$ denotes the Euclidean distance between particle $i$ and particle $q$, and $\varepsilon$ is a small constant to make sure the denominator is not zero. $G(t)$ is the gravitational constant whose value changes dynamically, which can be expressed as:

$$G(t) = G_0 e^{-\alpha \frac{t}{T}}$$  \hspace{1cm} (12)

where $\alpha$ represents the attenuation rate, $T$ is the max iteration times, and $G_0$ is the initial value.

At the $i^{th}$ iteration, the resultant force $F_i^k(t)$ caused by all the other particles will play a role to $i^{th}$ particle in the $k^{th}$ dimension, so:

$$F_i^k(t) = \sum_{q=1, q \neq i}^{NP} rand_q \cdot F_{iq}^k(t)$$  \hspace{1cm} (13)

where $rand_q$ is a random number within the range $[0,1]$.

According to Newton’s second law, the acceleration produced by this resultant force can be expressed as:

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)}$$  \hspace{1cm} (14)
The inertial mass of each particle is calculated from its fitness function value and the updating equation of inertial mass is given below:

$$
m_i(t) = \frac{\text{PSLL}_i(t) - \text{gWorst}(t)}{\text{gBest}(t) - \text{gWorst}(t)}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{q=1}^{NP} m_q(t)}$$

(15)

where PSLL$_i$(t) represents the fitness function value of particle $i$ at $t^{th}$ iteration, and gWorst(t) is the worst position found by the swarm at $t^{th}$ iteration.

Therefore, after applying the global learning strategy, the updating equation of particle velocity can be rewritten as:

$$v_{id}(t + 1) = \begin{cases} 
    w \cdot v_{id}(t) + c_1 r_1(t) \cdot a_{id}(t), & d \notin C \\
    w \cdot v_{id}(t) + c_1 r_1(t) [pGood_{id}(t) - x_{id}(t)], & d \in C
\end{cases}$$

(16)

where $d = 1, 2, \ldots, D$. D represents the number of dimensions, C denotes a set of $q$ dimensions randomly selected from all $D$ dimensions, $pGood_{id}(t)$ is the best position of $l$ candidate solution randomly selected from $P_G$ and neighborhood particles along the $d^{th}$ dimension for the $i^{th}$ particle at $t^{th}$ iteration, $a_{id}(t)$ is the acceleration along the $d^{th}$ dimension for the $i^{th}$ particle at $t^{th}$ iteration.

### 3.2.2. Local Search Strategy

An algorithm should have strong local search ability to improve the convergence efficiency in the later stage of optimization, especially when it is applied to the optimal design of large-scale array. The flowchart of the local search strategy is shown in Figure 3.

---

**Figure 3.** The flowchart of the local search strategy.
It can be considered that a particle $i$ can be regarded as moving from a certain position $x_i$ to another position $x'_i$ in the solution space when $x_i$ along some dimensions are mutated. Therefore, we propose a local search method that the neighborhood of the optimal position of a particle $i$ is searched by only changing partial elements of its position $x_i$ under the guidance of a Gaussian random variable. The position moving the equation along the $j^{th}$ dimension for the $i^{th}$ particle is given by:

$$x_{ij} = \begin{cases} -x_{ij}, & r_j < \left| X_{ij}(0,\sigma^2) \right| \\ x_{ij}, & r_j \geq \left| X_{ij}(0,\sigma^2) \right| \end{cases} \quad (17)$$

where $X_{ij}(0,\sigma^2)$ represents a Gaussian random variable with a mean value of 0 and standard deviation of $\sigma$, and $r_j$ is a random number within the range [0,1].

The mutated position $x'_i$ should be used to replace $x_i$ if it has a better fitness function value, and the standard deviation $\sigma$ should be amplified to increase the moving distance of particles. Otherwise, keep $x_i$ unchanged and reduce $\sigma$. The updating equation of standard deviation $\sigma$ can be expressed as:

$$\sigma(k+1) = \begin{cases} \sigma(k) \cdot \rho, & \text{PSLL}(x'_i) < \text{PSLL}(x_i) \\ \sigma(k) \cdot \mu, & \text{PSLL}(x'_i) \geq \text{PSLL}(x_i) \end{cases} \quad (18)$$

where $k$ represents the number of variations, $\rho$ is the expansion parameter and $\rho > 1$, $\mu$ is the contraction parameter and $0 < \mu < 1$. If the size of the standard deviation $\sigma$ is less than a preset threshold $\sigma_0$, that means there is no better solution for particle $i$ in the adjacent position. So, the update should be stopped.

### 3.2.3. Particle Movement Condition Monitoring

In order to avoid some particles falling into local optimum prematurely and to improve the performance of the optimization algorithm, a condition monitoring strategy is proposed to adjust the position of the particle according to the moving state of the particle.

There are two preconditions for a particle $i$ to trigger a mutation operation. The first one is that the optimal solution of particle $i$ has not been updated for $u$ iterations, which can be expressed as:

$$\text{Tpb}_i > u \quad (19)$$

where $\text{Tpb}_i$ represents the number of iterations of the optimal solution that have not been updated. The second precondition is that the moving distance of the position $x_i$ is less than the set value $\epsilon$ for successive $h$ iterations, which can be expressed as:

$$\text{Txb}_i > h \quad (20)$$

where $\text{Txb}_i$ represents the number of iterations in which the moving distance of the position $x_i$ is less than the set value $\epsilon$. The moving distance is defined as the number of different elements in all dimensions before and after the particle position changes.

It can be assumed that particle $i$ has fallen into local optimum when both preconditions are met, and then an individual variation probability $\gamma_a$ is used to vary all dimensions of position $x_i$. The equation of variation can be represented as:

$$x_{id} = \begin{cases} -x_{id}, & r_d < \gamma_a \\ x_{id}, & r_d \geq \gamma_a \end{cases} \quad (21)$$

where $d = 1, 2, \ldots, D$, $D$ represents the number of dimensions, and $r_d$ is a random number within the range [0,1].
3.3. Steps of Algorithm

The new Algorithm 1 exploits the hybrid search strategies (HSS) described above to improve the performance of a fundamental DPSO. Therefore, it is called DPSO-HSS. The detailed steps of the improved algorithm are summarized as follows.

**Algorithm 1: DPSO with hybrid search strategies**

1. Generate the initial particle swarm that satisfying the conditions. Initialize $p_{Best}$, $g_{Worst}$, and $g_{Best}$. Initialize observation parameters $T_{pb}$, and $T_{xb}$. Initialize the solution set $P_{G}$.
2. Calculate the fitness function value of each particle and update $p_{Best}$, $g_{Worst}$, $g_{Best}$, and $T_{pb}$. Replenish the set $P_{G}$ with the good solutions that have been eliminated.
3. Update the velocity and position of the particle according to (16), (9), and (10). Determine whether the number of iterations $t$ is larger than $t_{L}$. If so, go to Step 4. Otherwise, go to Step 5.
4. Initiate the local search strategy.
5. Determine whether $T_{pb} > u$ and $T_{xb} > h$ are both valid. If so, update the position of the particle according to (21). Otherwise, go to Step 6.
6. Constrain the particle position according to the constraint condition and update the parameter $T_{xb}$.
7. Do boundary treatment for particle velocity.
8. Output $g_{Best}$. Determine whether the termination conditions are met. If so, end the optimization. Otherwise, $t = t + 1$, and return to Step 2.

4. Numerical Results and Analysis

In this section, several examples are presented to compare the performance, effectiveness and robustness of the DPSO-HSS algorithm and some contrast algorithms.

4.1. Simulation 1: Function Optimization Tests

The algorithms tested include the proposed DPSO-HSS algorithm, RDPSO algorithm in [13], and NPSOWM algorithm in [14].

The five typical functions used to compare the performance of each algorithm are shown in Table 1. The test is to minimize the function value within a specified range of dimensions and variables. In order to compare the effects of each algorithm under the same conditions, the test of each function is run several times and its statistical results are calculated for comparison.

| Name      | Range            | Dimension | Minimum | Population | Running time | Number of Iterations |
|-----------|------------------|-----------|---------|------------|--------------|----------------------|
| Ackley    | $-30 \leq x_i \leq 30$ | 10        | 0       | 150        | 50           | 500                  |
| Rastrigin | $-10 \leq x_i \leq 10$ | 10        | 0       | 200        | 50           | 1000                 |
| Sphere    | $-10 \leq x_i \leq 10$ | 30        | 0       | 100        | 100          | 500                  |
| Rosenbrook| $-30 \leq x_i \leq 30$ | 10        | 0       | 200        | 50           | 1000                 |
| Griewank  | $-30 \leq x_i \leq 30$ | 10        | 0       | 100        | 100          | 500                  |

The three statistical characteristics as evaluation criteria are:
1. The mean value of multiple simulation results;
2. The variance of multiple simulation results;
3. The minimum value of multiple simulations.

The population size $NP$, the iteration times $T$, the dimension $D$, the learning factors $c_1$ and $c_2$, and the inertia weight $w$ of the three algorithms are the same. The simulation results are shown in Table 2.
Table 2. Statistical properties of function tests using different algorithms.

| Function    | Mean   | Variance | Minimum               |
|-------------|--------|----------|-----------------------|
| Ackley      | 0.0096 | 2.8385   | 2.7257                |
| Rastrigin   | 7.5486 | 11.8201  | 32.1085               |
| Sphere      | 1.2493 | 2.1050   | 5.0407                |
| Rosenbrock  | 3.8523 | 2.3869   | 39.3544               |
| Griewank    | 0.8081 | 0.0764   | 0.3513                |

Among the five test functions, the proposed algorithm outperforms the other two algorithms in the mean and variance of Ackley, Rastrigin, and Sphere. In Rosenbrock’s test, DPSO-HSS has a better variance. In Griewank’s test, the test results of DPSO-HSS and RDPSO are close to the same. Considering the test results of the above five functions comprehensively, the proposed DPSO-HSS algorithm performs well in both mean and variance compared with the RDPSO and NPSOWM algorithm. The low mean indicates the excellent global search ability and convergence effect of the algorithm, whereas the low variance indicates that the stability of the algorithm results in multiple runs.

The time complexity of each test function is $O(D)$, where $D$ is the number of dimensions. By substituting this result into the optimization of the DPSO-HSS algorithm, the time complexity of DPSO-HSS can be calculated as follows

$$T_{DPSO-HSS} = O\left[tp^2q + tpD\right]$$

where $D$ represents the number of dimensions, $p$ is the swarm population, $q$ denotes the number of dimensions randomly selected from all $D$ dimensions in (16), and $t$ is the number of iterations. It can be seen that, compared with the general PSO algorithm, the improved algorithm has higher time complexity.

4.2. Simulation 2: Application in Planar Array Thinning with Constraints

The algorithms involved in the simulation include the proposed DPSO-HSS algorithm, RDPSO algorithm in [13], NPSOWM algorithm in [14], and MOPSO-CO algorithm in [4].

Consider a planar array consisting of $20 \times 20$ elements with equal spacing $d = 0.5\lambda$ as an initial layout. $u$ and $v$ are both within the range $[-1, 1]$ with a scanning step of 0.01. The main beam direction is $(\theta, \phi) = (0^\circ, 90^\circ)$, that is, $(u, v) = (0, 0)$. The filling rate is 50%, so there are 200 elements in the “ON” state. The population size $NP = 100$, iteration times $T = 500$, and dimensions $D = 400$ are the same for all algorithms used.

The convergence curve of PSLL using four algorithms for array thinning is shown in Figure 4. At the early stage of optimization, the DPSO-HSS algorithm does not have more impressive search efficiency than other algorithms, but it maintains good global searching ability with the guiding of comprehensive learning strategy and aims at maintaining the high diversity of the population and the correlation between the particles. At the later stage of optimization, the curve of the proposed algorithm shows the clear change process with
local search strategy before reaching the optimal PSLL results. The trend of DPSO-HSS corresponds to the configured search strategy and meets the ideal expectation.

The time complexity of each test function is $O(D)$, where $D$ is the number of dimensions randomly selected from all variables. By substituting this result into the optimization of the DPSO-HSS algorithm, the time complexity of DPSO-HSS can be calculated as follows:

$$\text{Time Complexity} = O(D)$$

The algorithms involved in the simulation include the proposed DPSO-HSS algorithm, RDPSO algorithm in [13], NPSOWM algorithm in [14], and MOPSO-CO algorithm.

Figure 4. The convergence curve of PSLL using four kinds of algorithms in thinning the $10\lambda$ diameter array.

The elements distribution and beam pattern diagram of the optimized array of DPSO-HSS algorithm are shown in Figures 5 and 6, respectively. In Figure 5, the black triangles represent the array elements in the “ON” state, and the number of these elements is 200, which meets the filling rate of 50%. The four angular array elements in the array are all in the “ON” state because the array aperture must be unchanged. On the other hand, the direction of the main beam is $(u, v) = (0, 0)$ with the normalized amplitude of 0 dB, and the PSLL is $-18.32$ dB. These results mean the result of the array thinning design is correct.

Figure 5. The array elements’ distribution of an optimized array using the DPSO-HSS algorithm.
We can see the PSLL performance in details in Figure 7, illustrating the novelty of the proposed DPSO-HSS algorithm, which plots the $u$-cut of the beam pattern diagram of a thinned array optimized by four kinds of algorithms. The main lobe width of the beam pattern of each algorithm is identical. The PSLL of the DPSO-HSS algorithm is $-18.32$ dB, whereas those of RDPSO, NPSOWM, and MOPSO-CO are $-17.15$ dB, $-17.42$ dB, and $-17.66$ dB, respectively. Compared with the other three algorithms, the PSLL of DPSO-HSS is decreased by 6.82%, 5.17%, and 3.74%, respectively.

### 4.3 Simulation 3: Beam Pattern of The Optimized Array with Different Main Beam Positions

In the above example, we set the main beam to a normal direction. Reference [21] proved that the main beam direction configuration has no impact to PSLL in unequally spaced linear antenna array thinning. In fact, this conclusion could also be obtained for planar array thinning. The $v$-cut of the radiation beam pattern diagram is shown in Figure 8 with different main beam directions, i.e., $(\theta, \varphi) = (0^\circ, 90^\circ)$, $(\theta, \varphi) = (30^\circ, 45^\circ)$, and $(\theta, \varphi) = (60^\circ, -30^\circ)$. The PSLLs are $-18.32$ dB, $-18.30$ dB, and $-18.31$ dB, respectively. The difference is less than 0.1 dB, which is consistent with the conclusion in [21].
As shown in Table 3, the 3 dB bandwidth of the main lobe is mainly affected by the aperture, with an increase in aperture leading to a slight decrease in bandwidth. However, the effect on the 3 dB beamwidth in the v direction is not as significant. The PSLL values also show a decrease as the aperture increases, with the optimal aperture for minimizing PSLL being different for each algorithm. The results indicate that the filling rate also plays a role in the performance of the array, with a higher filling rate leading to a slight increase in PSLL and a decrease in 3 dB beamwidth.

The results of PSLL under different filling rates show that the filling rate does have a certain influence on the optimization effect of thinned array and the optimization effect will be reduced if the number of open elements is too many or too few. The PSLL is improved when the filling rate is a constant of 60% and the aperture is changed from 5λ to 10λ, which indicates that the aperture also has a certain influence on the performance of array thinning. As shown in Table 3, the 3 dB bandwidth of the main lobe is mainly affected by the aperture, and the larger the array aperture is, the smaller the 3 dB bandwidth is.

5. Conclusions

A novel optimization algorithm for large array thinning based on DPSO with hybrid search strategies is proposed to improve the performance of large planar array thinning. The proposed algorithm, named DPSO-HSS, utilizes a global learning strategy to improve...
the diversity of populations at the early stage of optimization. A dispersive solution set and the gravitational search algorithm are used during particle velocity updating. Then, a local mutation strategy is employed in the later stage of optimization so that the local convergence is enhanced by continuous search around the best position of the particle. Several of the above-mentioned representative examples of large planar arrays thinning are provided to demonstrate the effectiveness and robustness of the DPSO-HSS algorithm. The brief comparison results are shown in the Table 4.

Table 4. Comparison of the proposed algorithm and other algorithms in key performance metrics.

| Metrics  | DPSO-HSS | MOPSO-CO [4] | RDPSO [13] | NPSOWM [14] |
|----------|----------|--------------|------------|-------------|
| Efficiency | Normal    | Normal       | Good       | Good        |
| Stability  | Good      | Normal       | Good       | Normal      |
| PSLL       | Good      | Normal       | Normal     | Normal      |

Author Contributions: W.C., L.J., C.G., K.M. and H.Z. conceived and designed the algorithm and experiments together; W.C., L.J. and H.Z. performed the simulations and experiments and analyzed the results. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Equipment Pre-Research Foundation of China during the “14th Five-Year Plan”, grant number 629010204.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Y.; Liu, Q.H.; Nie, Z. Reducing the Number of Elements in Multiple-Pattern Linear Arrays by the Extended Matrix Pencil Methods. *IEEE Trans. Antennas Propag.* 2014, 62, 652–660. [CrossRef]
2. Haupt, R.L. Adaptively Thinned Arrays. *IEEE Trans. Antennas Propag.* 2015, 63, 1626–1632. [CrossRef]
3. Rahardjo, E.T.; Sandi, E.; Zulkifli, F.Y. Design of Linear Sparse Array Based on the Taylor Line Source Distribution Element Spacing. In Proceedings of the 2017 IEEE Asia Pacific Microwave Conference (APMC), Kuala Lumpur, Malaysia, 13–16 November 2017.
4. Cao, A.; Li, H.; Ma, S.; Jing, T.; Zhou, J. Sparse circular array pattern optimization based on MOPSO and convex optimization. In Proceedings of the 2015 Asia-Pacific Microwave Conference (APMC), Nanjing, China, 6–9 December 2015.
5. Liu, X.; Hu, F.; Wei, X.; Yu, D.; Gang, G. Cuboid sparse array synthesis for sensor selection by convex optimization with constrained beam pattern based on WBAN. In Proceedings of the International Conference on Wireless Communications & Signal Processing, Nanjing, China, 15–17 December 2015.
6. Keizer, W. Linear Array Thinning Using Iterative FFT Techniques. *IEEE Trans. Antennas Propag.* 2008, 56, 2757–2760. [CrossRef]
7. Gu, L.; Zhao, Y.W.; Zhang, Z.P.; Wu, L.F.; Hu, J. Adaptive Learning of Probability Density Taper for Large Planar Array Thinning. *IEEE Trans. Antennas Propag.* 2020, 69, 155–163. [CrossRef]
8. Cheng, Y.F.; Shao, W.; Zhang, S.J.; Li, Y.P. An Improved Multi-Objective Genetic Algorithm for Large Planar Array Thinning. *IEEE Trans. Magn.* 2016, 52, 1–4. [CrossRef]
9. Dai, D.; Yao, M.; Ma, H.; Wei, J.; Zhang, F. An Asymmetric Mapping Method for the Synthesis of Sparse Planar Arrays. *IEEE Antennas Wirel. Propag. Lett.* 2018, 17, 70–73. [CrossRef]
10. Durmus, A. Novel Metaheuristic Optimization Algorithms for Sidelobe Suppression of Linear Antenna Array. In Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 21–23 October 2021; pp. 291–294.
11. Guo, J.; Xue, P.; Zhang, C. Optimal Design of Linear and Circular Antenna Arrays Using Hybrid GWO-PSO Algorithm. In Proceedings of the 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 138–141.
12. Vankayalapati, S.; Pappula, L.; Kumar, K.; Panda, P.K. Thinned Planar Antenna Array Synthesis: A Multiobjective Improved Binary Cat Swarm Optimization Approach. In Proceedings of the 2021 7th International Conference on Signal Processing and Communication (ICSC), Noida, India, 25–27 November 2021; pp. 129–132.
13. Brahma, D.; Deb, A. Optimal Design of Antenna Array using Tuned Random Drift Particle Swarm Optimization Algorithm. In *IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*; IEEE: Vancouver, BC, Canada, 2020.

14. Bera, R.; Mandal, D.; Ghoshal, S.P.; Kar, R. Wavelet Mutation based Novel Particle Swarm Optimization technique for comparison of the performance of single ring planar antenna arrays. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2016.

15. Zhang, H.; Zheng, J.; Bai, B.; Zhou, Y. Optimal Design of Sparse Array for Ultrasonic Total Focusing Method by Binary Particle Swarm Optimization. *IEEE Access* 2020, 8, 111945–111953. [CrossRef]

16. Zhang, Q.; Mao, J.; Zhang, Y.; Lu, M.; Li, R.; Liu, X.; Liu, Y.; Yang, R.; Wang, X.; Geng, Y.; et al. Multiple-Focus Patterns of Sparse Random Array Using Particle Swarm Optimization for Ultrasound Surgery. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* 2022, 69, 565–579. [CrossRef] [PubMed]

17. Li, H.; He, F.; Chen, Y.; Luo, J. Multi-objective Self-organizing Optimization for Constrained Sparse Array Synthesis. *Swarm Evol. Comput.* 2020, 58, 100743. [CrossRef]

18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.

19. Li, X. Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology. *IEEE Trans. Evol. Comput.* 2010, 14, 150–169. [CrossRef]

20. Doraghinejad, M.; Nezamabadi-pour, H.; Sadeghian, A.H.; Maghfoori, M. A hybrid algorithm based on gravitational search algorithm for unimodal optimization. In Proceedings of the 2012 2nd International eConference on Computer and Knowledge Engineering (ICCCE), Mashhad, Iran, 18–19 October 2012; pp. 129–132.

21. Miranda, A.V.; Ashwin, P.; Sharan, P.; Gangwar, V.S.; Singh, A.K.; Singh, S.P. An efficient synthesis of unequally spaced antenna array with electronic scan capability utilizing particle swarm optimization. In Proceedings of the 2017 IEEE MTT-S International Microwave and RF Conference (IMaRC), Ahmedabad, India, 11–13 December 2017; pp. 255–258.