Progress In Electromagnetics Research, Vol. 118, 205–221, 2011

SYNTHESIS OF UNEQUALLY SPACED ANTENNA ARRAYS BY USING INHERITANCE LEARNING PARTICLE SWARM OPTIMIZATION

D. Liu[1,2,*], Q.-Y. Feng[2], W.-B. Wang[3], and X. Yu[4]

[1]School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

[2]School of Information Science & Technology, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

[3]School of Electric Information, Xihua University, Chengdu, Sichuan 610039, China

[4]School of Information Science & Technology, Xiamen University, Xiamen, Fujian 361005, China

Abstract—In this paper, synthesis of unequally spaced linear antenna arrays based on an inheritance learning particle swarm optimization (ILPSO) is presented. In order to improve the optimization efficiency of the PSO algorithm, we propose an inheritance learning strategy that can be applied to different topology of different PSO algorithms. In ILPSO algorithm, each cycle contains several PSO optimization processes, and uniform initial particle positions, part of which inherited from the good results in pre-cycles, are adopted in post-cycles. ILPSO enhances the exploration ability of PSO algorithm significantly, and can escape from the trap of local optimum areas with greater probability. The results demonstrate good performance of the ILPSO in solving a set of eight 30-D benchmark functions when compared to nine other variants of the PSO. The novel proposed algorithm has been applied in 32-element position-only array synthesis with three different constraints, simulation results show that ILPSO obtains better synthesis results reliably and efficiently.

Received 5 May 2011, Accepted 21 June 2011, Scheduled 3 July 2011

* Corresponding author: Dong Liu (liudong@home.swjtu.edu.cn).

1. INTRODUCTION

Synthesis of linear antenna arrays is a classic optimization problem in electromagnetism [1], whose main objective is to find an appropriate excitation vector and a layout of the elements to generate the desired radiation pattern.

Various techniques have been developed to array synthesis. The classic techniques have many practical difficulties in the array design especially if there exist some restricted conditions [1, 2]. In recent years, the evolutionary algorithms (EAs) for array synthesis have been extensively studied [1, 3–17]. Several global optimization algorithms such as differential evolution (DE) [3–5], genetic algorithm (GA) [6–8], ant colony optimization (ACO) [9], and particle swarm optimization (PSO) [10–15] are used in antenna array pattern. However, these methods appear certain drawbacks to the possibility of premature convergence to a local optimum when achieving the global optimum.

The synthesis of unequally spaced array has attracted increasing attention [1, 4, 5, 13, 16, 17]. The main advantage of unequally spaced antenna array is that the array can work with uniform amplitude excitation. Compared to equally spaced array which uses non-uniform amplitude or phased excitation, unequally spaced array can reduce the system cost and difficulties in designing feeding network [4]. For antenna engineers it is a big challenge, which mainly originates from the nonlinear and non-convex dependency of the array factor to element positions and excitation phases, to synthesize antenna array with sidelobe level suppression and null control. For unequally spaced linear array synthesize, especially for position-only (PO), the array synthesis is much more difficult because only element positions can be adjusted.

In this paper, an inheritance **learning** particle swarm optimization (ILPSO) is applied to position-only synthesis of unequally spaced arrays. The simulation results show that ILPSO is able to get lower peak sidelobe levels (PSLLs) than (at least the same as) those reported in previous literatures [1, 4, 13].

The rest of this paper proceeds as follows. The PSO, ILPSO are briefed in Section 2 and Section 3 respectively. And Section 4 formulates synthesis of a general linear antenna array. Numerical results are illustrated in Section 5. Conclusion is drawn in Section 6.

2. PARTICLE SWARM OPTIMIZER

In this section, the classic PSOs are first introduced, followed by the balance analysis of exploration and the exploitation of classic PSO. Then the inefficiency reasons are discussed while the classic PSOs are used to optimize complex multimodal problems.

2.1. Classic PSO

PSO is an evolutionary algorithm proposed by Kennedy and Eberhart [18] and has been successfully applied to many scientific and engineering problems [19]. In original PSO, each individual possible solution is modeled as a particle that moves through the real number hyperspace of the optimization problem. The position of each particle is determined by its old position and current velocity [20].

In LDWPSO algorithm, at $t+1$ iteration, the ith particle's position x_i and velocity v_i are updated as follows:

$$v_{ij}(t+1)=w(t)v_{ij}(t)+c_1r_{1j}(t)(p_{ij}-x_{ij}(t))+c_2r_{2j}(t)(p_{gj}-x_{ij}(t)) \quad (1)$$

$$w(t)=w_{\min}+(w_{\max}-w_{\min})\cdot t/\max_{-}gen \quad (2)$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \quad (3)$$

where i is the particle serial number, j is the dimension number, t is the time step, $\max_{-}gen$ is the total generation number, $[w_{\max}, w_{\min}]$ are range of inertia weight, $w(t)$ is current inertia weight, p_i is the personal best position of particle i, p_g is the global best position of the particle swarm, c_1, c_2 are acceleration coefficients, $r_{1j}(t)$, $r_{2j}(t)$ are random numbers usually uniformly distributed in [0, 1]. Usually, $w_{\max}$, $w_{\min}$ are set to 0.9 and 0.4 respectively.

The velocity updating equation has three major components which refer to the old velocity, the distances between particles, its best historic position and the global best particle position. The original PSO algorithm has some deficiencies and cannot ensure the algorithm convergence. Therefore, to guarantee the convergence and to prevent the "explosion" of the swarm [20], several considerations must be taken into account, including limiting the maximum velocity, selecting acceleration constants, the constriction factor, or the inertia constant [20, 21].

The introduction of inertia constant brings an important improvement to PSO algorithm, and the constriction factor can be considered as a variant of the inertia constant. Appropriate inertia constant can prevent "swarm explosion" and ensure the swarm convergence. The inertia constant controls the exploration of the search space, and can be either implemented as a fixed value or changing dynamically [22–25]. Commonly, larger inertia constant makes swarm converge slower with stronger exploration ability, and vice versa. PSO with a fixed inertia constant is usually used as standard PSO (SPSO). A linearly decreasing inertia weight particle swarm optimizer (LDWPSO) has produced good results in many applications [22]. The main disadvantage of this method is that once the inertia weight is decreased, the swarm loses its exploration ability to search new areas globally.

2.2. Balance of Searching Ability

For any swarm optimization algorithms, the balance between exploration and exploitation must be taken into account. The exploration ability of algorithm makes it possible to search the optimum area in whole parameter multi-space, and the exploitation ability is used to improve the precision of the solutions found in optimum area. Poli [26] pointed out that the mean and standard deviation of swarm in any converged PSO (with inertia constant) algorithm must tend to a fix finite value. So in any converged PSO with inertia constantthe swarm distributing space becomes smaller and smaller while optimization process goes on. And in this process, the ability of exploration decreases while the ability of exploitation increases. If the searching time of exploration and exploitation is better balanced, the performance of the algorithm will be satisfied. For PSO and most of its variants, the swarm will congregate quickly to the best position found by swarm when the optimization generation is large. In this condition, most searching computation is consumed on exploitation searching. Too much exploitation searching reduces the optimization efficiency, especially on complex optimization problems.

2.3. Randomicity of Classic PSO

Randomicity is an essential character of evolution optimization algorithm [26]. No matter how the PSO algorithm is improved, the optimization results still stays random. The success probability of optimization may be greater if the improved PSO algorithm has better performance, and vice versa. In engineering optimization application, the increment of problem complexity is much larger than the improvement of optimization algorithm performance. Therefore, for complex problems, if large optimization generation cannot ensure algorithms to find the optima result, we can increase the executions to improve the optimization efficiency.

3. INHERITANCE LEARNING PARTICLE SWARM OPTIMIZER

From what have been discussed above, we can come to a conclusion that the balance of searching ability and the optimization randomicity are both connected to algorithm performance. Exploration is more important for a PSO algorithm while optimizing complicated problems. Without enough exploration searches, exploitation can only improve the precision in local optimal solution area. Many researches try to improve the exploration ability of PSO in different ways [19], such as changing parameter strategies, using different topology structures, adopting more effective information share strategies, hybridizing operation of other evolution algorithms (such as GA and DE), and so on. Different strategies are adopted to make particles share information more efficiently so that swarm gains better exploration ability. Unfortunately, most variants of PSO still have no enough exploration search time while optimizing complex problems with very large optimization generations.

In order to get better balance searching ability and to take the randomicity in count, the suitable maximal generation of LDWPSO and Inheritance Learning (IL) strategy are both discussed. These strategies are adopted to improve the LDWPSO performance without any changes to algorithm.

3.1. Maximal Generation of Single PSO Process

Single PSO optimization is a whole optimizing process with the certain PSO algorithm working on the certain optimization problem. It is the basis of ILPSO and decides the balance searching ability of whole algorithm.

When optimizing an appointed optimization problem with large generations, most of generations in LDWPSO are consumed on exploitation searching. Most of searching computation is wasted on useless precision improvement of local optimum solution and have little contribution to global result. If we set a suitable maximal generation

Table 1. Success optimization results of different maximal generations and test times (Mg = maximal generation, Tt = test times).

Success Condition	Success Result of Each Parameter Set (Mg/Tt)					
	100/ 1000	200/ 500	400/ 250	600/ 166	800/ 125	1000/ 100
< 600	71	**75**	64	59	57	49
< 500	36	**38**	36	**38**	29	30
< 400	15	17	14	**18**	12	15
< 300	6	7	6	5	4	6
< 200	1	**3**	2	1	1	1

for LDWPSO on an appointed problem, the algorithm will end when swarm congregates to a small area with enough solution precision. Then, the computation can be used as much as possible on exploration search.

For example, we use 10-D Schwefel function as the benchmark, the w range of LDWPSO is [0.9, 0.4], the particle population is 10. When search computation is set to $10 + E5$, there are different combinations with different maximal generations and test times. The optimization results with different combinations are given in Table 1.

For 10-D Schwefel function, the global optimal is 0. Table 1 shows that the algorithm has the best efficiency when maximal generation is set to 200. So suitable maximal generation can make LDWPSO have

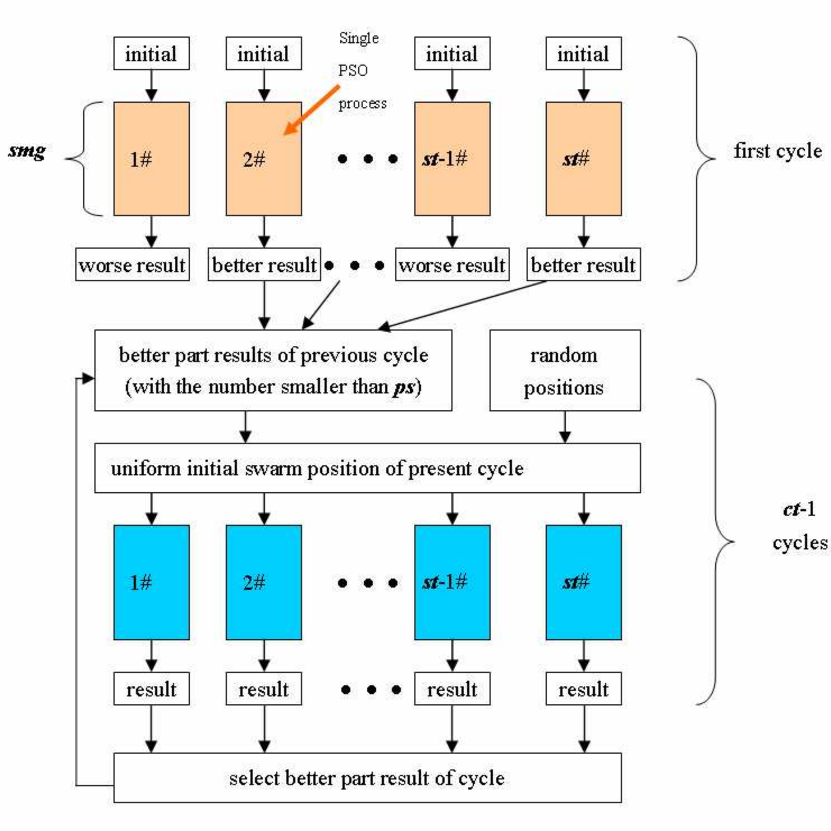

Figure 1. Algorithm frame of ILPSO. smg = single maximal generation. st = PSO process in one cycle. ct = cycle times.

a better balance of searching ability and can improve the algorithm computation efficiency.

3.2. Inheritance Learning PSO

Though we can improve LDWPSO performance and success probability by suitable use of LDWPSO, the precision problem still exists. In Table 1, most results are higher than 500. The suitable maximal generation strategy improves the balance searching ability and the success probability of LDWPSO, but the solution precision is still not satisfying. In order to keep a balance of searching ability and improve the results precision, the inheritance learning PSO (ILPSO) is presented. The ILPSO can serve as a frame or a container for single PSO optimization processes, and its details are given in Figure 1.

ILPSO works with process cycles, each of which is consist of a number of single PSO optimizing processes. Each single PSO optimizing process uses the same parameter set, such as w, c_1, c_2, population size and maximal generation. The maximal generation is suitable for better balance searching ability.

In an ILPSO cycle, a number of single PSO process are executed independently. This strategy helps ILPSO to escape from the trap of local optimum area with greater probability. Processes in one cycle (except the first cycle) using uniform initial particle position which inherited from previous cycle. ILPSO works with tremendous balance searching abilities and swarm diversities, which improves algorithm performance greatly in a simple way, especially on complex multidimensional problems with a very large maximal generation.

3.3. Simulation of ILPSO Algorithm

To verify its effectiveness, ILPSO has been applied to classical benchmark functions. All simulations are conducted in a Windows 7 Professional OS environment using 12-core processors with Intel Xeon (R), 3.33 GHz, 72 GB RAM and the codes are implemented in Matlab 7.10.

3.3.1. Benchmark Functions

In this section, eight benchmark functions are employed, including unimodal and multimodal functions in [27, 28]. The test functions, search ranges and optimal goals are listed in Table 2.

Table 2. Benchmark functions.

Function Index	Function Name	$f(x^*)$	Initialization Range	Search Range	Dimension
f_1	Sphere	0	$[-100, 100]^D$	$[-100, 100]^D$	30
f_2	Rosenbrock	0	$[-2.048, 2.048]^D$	$[-2.048, 2.048]^D$	30
f_3	Ackley	0	$[-32.768, 32.768]^D$	$[-32.768, 32.768]^D$	30
f_4	Griewank	0	$[-600, 600]^D$	$[-600, 600]^D$	30
f_5	Weierstrass	0	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	30
f_6	Rastrigin	0	$[-5.12, 5.12]^D$	$[-5.12, 5.12]^D$	30
f_7	Noncontinuous Rastrigin	0	$[-5.12, 5.12]^D$	$[-5.12, 5.12]^D$	30
f_8	Schwefel	0	$[-500, 500]^D$	$[-500, 500]^D$	30

3.3.2. Parameter Settings

Most PSO algorithms can get better results for simple problems or lower dimensions problems, so we test ILPSO algorithm on the eight benchmark functions with 30 dimensions. The global optima x^*, the corresponding fitness value $f(x^*)$, the search ranges $[X_{\min}, X_{\max}]$, and the initialization range of each benchmark function are listed in Table 3.

When solving 30-D benchmark functions, LDWPSO is employed as the single PSO process of ILPSO. The population size is set to 40 and the total generations is set to 200 000. The inertia weight range is [0.4, 0.9], the single PSO maximal generation of ILPSO is set to 1000. There are 10 single PSO processes in one cycle. Each experiment runs 30 times. The mean values and standard deviation of the results are presented in Table 3 and compared with the results in [28].

3.3.3. Discussion

The ILPSO achieves the global optima on most complex multimodal problems. By analyzing the results of the ILPSO on 30-D problems, the conclusion is that the ILPSO does not perform the best for

Table 3. Results for 30D Benchmark functions (40 particles, 200000 generation).

PSOs \ Func	Sphere ($f1$)	Rosenbrock ($f2$)	Ackley ($f3$)	Griewank ($f4$)
PSO-w [28]	9.78e - 030 ± 2.50e - 029	2.933 + 001 ± 2.51e + 001	3.94e - 014 ± 1.12e + 000	8.13e - 003 ± 7.16e - 003
PSO-cf [28]	5.88e - 100 ± 5.40e - 100	1.11e + 001 ± 1.81e + 000	1.12e + 000 ± 8.65e - 001	2.06e - 002 ± 1.90e - 002
PSO-w-local [28]	5.35e - 100 ± 4.41e - 013	2.39e + 001 ± 3.07e + 000	9.10e - 008 ± 8.11e - 008	5.91e - 003 ± 6.69e - 003
PSO-cf-local [28]	7.70e - 030 ± 2.50e - 029	1.71e + 001 ± 9.16e - 001	5.33e - 015 ± 1.87e - 015	5.91e - 003 ± 8.70e - 003
UPSO [28]	4.17e - 087 ± 3.15e - 087	1.51e + 001 ± 8.14e - 001	1.22e - 015 ± 3.16e - 015	1.66e - 003 ± 3.07e - 003
FDR [28]	4.88e - 102 ± 1.53e - 101	5.39e + 000 ± 1.67e + 000	2.84e - 014 ± 4.10e - 015	1.01e - 002 ± 1.23e - 002
FIPS [28]	2.69e - 012 ± 6.84e - 013	2.45e + 001 ± 2.19e - 001	4.81e - 007 ± 9.17e - 008	1.16e - 006 ± 1.87e - 006
CPSO-H [28]	**1.16e - 113 ± 2.92e - 113**	7.08e + 000 ± 8.01e + 000	4.93e - 014 ± 1.10e - 014	3.63e - 002 ± 3.60e - 002
CLPSO [28]	4.46e - 014± 1.73e - 014	2.10e + 001 ± 2.98e + 000	**0 ± 0**	3.14e - 010 ± 4.46e - 010
ICLPSO	1.31e - 081± 2.15e - 081	**1.33e - 001 ± 7.28e - 001**	8.941e - 015 ± 2.46e - 015*	**0 ± 0**

PSOs \ Func	Weierstrass ($f5$)	Rastrigin ($f6$)	Noncontinuous Rastrigin ($f7$)	Schwefel ($f8$)
PSO-w [28]	1.30e - 004 ± 3.30e - 004	2.90e + 001 ± 7.70e + 000	2..97e + 001 ± 1.39e + 001	1.10e + 003 ± 2.56e + 002
PSO-cf [28]	4.10e + 000 ± 2.20e + 000	5.62e + 001 ± 9.76e + 000	2.85e + 001 ± 1.14e + 001	3.78e + 003 ± 6.02e + 002
PSO-w-local [28]	4.94e - 003 ± 1.40e - 002	2.72e + 001 ± 7.58e + 000	2.08e + 001 ± 4.94e + 000	1.53e + 003 ± 3.00e + 002
PSO-cf-local [28]	1.16e - 001 ± 2.79e - 001	4.53e + 001 ± 1.17e + 001	1.54e + 001 ± 1.67e + 001	3.78e + 003 ± 5.37e + 002
UPSO [28]	9.60e+ 000 ± 3.78e + 000	6.59e + 001 ± 1.22e + 001	6.34e + 001 ± 1.24e + 001	4.84e + 003 ± 4.76e + 002
FDR [28]	7.49e - 003 ± 1.14e - 002	2.84e + 001± 8.71e + 000	1.44e + 001 ± 6.28e + 000	3.61e + 003 ± 3.06e + 002
FIPS [28]	1.54e - 001 ± 1.48e - 001	7.30e + 001 ± 1.24e + 001	6.08e + 001 ± 8.35e + 000	2.05e + 003 ± 9.58e + 002
CPSO-H [28]	7.82e - 015 ± 8.50e - 015	**0 ± 0**	1.00e - 010 ± 3.16e - 001	1.08e + 003 ± 2.59e + 002
CLPSO [28]	3.45e - 007 ± 1.94e - 007	4.85e - 010 ± 3.63e - 010	4.36e - 010 ± 2.44e - 010	**1.27e - 012 ± 8.79e - 013**
ICLPSO	**0 ± 0**	**0 ± 0**	**0 ± 0**	1.12e + 003 ± 2.10e + 002

* The result precision of Ackley functions is limited by the computer platforms, when $x^* = [0, 0, \ldots, 0]^{30}$, the $f(x^*) = 8.88\text{e} - 016$

sphere function. As mentioned above, the ILPSO uses an inheritance learning strategy to get better exploration searching ability, and the exploitation ability is decreased with same maximal generation. More generation should be needed to improve the result precision of sphere function. This is the cost for ILPSO to obtain better performance on complex multimodal problems. Therefore, it is difficult for one PSO variant to get the best performance on all classes of problems. The simple unimodal problems can be easily optimized by many algorithms, and the main improvements for PSO are focused on improving the PSO's performance on complex multimodal problems.

From the results above we can conclude that ILPSO improves the optimization efficiency of LDWPSO greatly and ILPSO can get better results than most mentioned PSOs in [28] on most benchmark functions. This property is due to the ILPSO novel learning strategy. Each PSO process learns from the best results of previous cycle, and uses a suitable maximal generation to ensure the balance searching ability, so the success probability is improved greatly. If the number of single PSO process in one cycle becomes larger, the ILPSO can get better performance on all benchmark functions.

4. ARRAY FACTOR AND SYNTHESIS OBJECTIVE

4.1. Array Factor

The array factor of a linear antenna array at angle θ can be expressed as (4):

$$\mathrm{AF}(\theta)=\sum_{i=-N}^{N} I_i e^{j\left(\frac{2\pi}{\lambda}x_i\sin(\theta)+\phi_i\right)} \tag{4}$$

where I_i, φ_i and λ are the excitation amplitude, phase of the element located at x_i and the wavelength respectively. For a position-only array synthesis with $2N$-element linear array symmetrically located along the x-axis (with no element located at zero), (4) becomes:

$$\mathrm{AF}(\theta)=2\sum_{i=1}^{N} e^{j\left(\frac{2\pi}{\lambda}x_i\sin\theta\right)} \tag{5}$$

4.2. Synthesis Objective

The goal of pattern synthesis optimization is to find the optimum element positions, so the sidelobe suppression, beamwidth and null control can meet with the setting target.

Usually, the above constraints are combined with one cost function by computing their proportion sum. Then the optimization problem is defined by the minimization of the objective function.

In this paper, the peak sidelobe level (PSLL) of the antenna array is defined as

$$\mathrm{PSLL}(x)=\max_{\forall\theta\in S}\frac{\mathrm{AF}(\theta)}{|\mathrm{AF}(\theta_0)|} \tag{6}$$

where S is the space spanned by angle excluding the mainlobe with the center at θ_0. In this paper, without loss of generality, θ_0 is set to zero.

Thus, the objective function to be minimized can be written as:

$$\begin{aligned} f(x) \;=\; & \alpha_1\cdot\mathrm{PSLL}(x)+\alpha_2\cdot\max\{0,(\mathrm{BW}-\mathrm{BW}_d) \\ & +\alpha_3\cdot\max\{0,\max_{\forall\theta_k\in\Psi}\{\mathrm{AF}^x_{\mathrm{dB}}(\theta_k)\}-C_{\mathrm{dB}}\}\} \end{aligned} \tag{7}$$

where BW is the calculated beamwidth, BW_d is the desired beamwidth, C_{dB} is the desired null level in dB, and Ψ is the aggregate of desired null angles. α_1, α_2, α_3 are proportion of each part, and are usually set to 0.8, 0.1, 0.1 respectively.

5. NUMERICAL RESULTS

In this section, ILPSO is applied to synthesize a 32-element symmetric linear array with three different constraints. The synthesis of this array has been studied before by other researchers [1, 4, 13]. For each test, the independent run times is 20, the population size is set to 60 and the total FE is set to 100 000. The inertia weight range is [0.4, 0.9], the single PSO maxima generation of ILPSO is set to 200, the PSO process number in one cycle is set to 50. Since the array synthesis is symmetrical, only the 16 element positions in one origin side are given in the all following simulation results.

5.1. Synthesis of Unequally Spaced Linear Array with PSLL Constraint

The first example is desired to synthesize a 32-element array factor with low PSLL by optimizing only elements positions. As the example in [4], the adjacent elements' distance is between 0.5λ and λ. Since PSLL is the only optimize goal, proportions of (9) are set to 1.0, 0, 0 respectively.

The best PSLL is -22.65 dB, same as the result in [4]. But the total FE in this paper is 100 000, much less than 500 000 in [4]. The corresponding element positions and radiation pattern are shown in Table 4 and Figure 2.

5.2. Synthesis of Unequally Spaced Linear Array with PSLL, BW and NULL Constraints

The second example is that of a 32-element array with a desired null at the direction of 9 [1, 13]. The desired null level is at -60 dB. The desired beamwidth is set to $7.1^\circ \pm 14\%$.

Table 4. Element positions of the first example (normalized with respect to $\lambda/2$).

Element Number	**1**	**2**	**3**	**4**
Position	0.5	1.5	2.5	3.5
Element Number	**9**	**10**	**11**	**12**
Position	9.1382	10.1929	11.275	12.9395
Element Number	**5**	**6**	**7**	**8**
Position	4.5	5.5	6.58	7.58
Element Number	**13**	**14**	**15**	**16**
Position	14.781	16.473	18.3671	20.2034

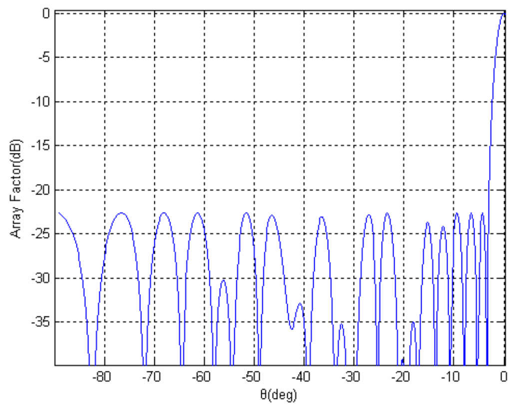

Figure 2. Best radiation pattern of the first example.

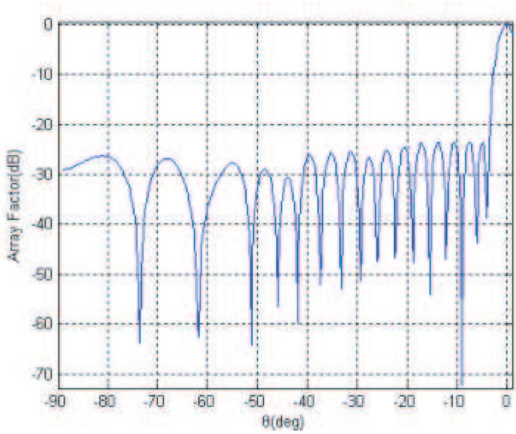

Figure 3. Best radiation pattern of the second example.

Unlike most researches, the minimum of adjacent elements' distance is smaller than 0.5λ in [1, 13], so the range of adjacent elements' distance may be set between 0.45λ and λ, which makes synthesis easier. However, smaller adjacent elements' distance may cause mutual coupling and grating lobes. When adjacent elements' distance is smaller than 0.5λ, formula (5) will not be appropriate anymore. Just for comparison, we keep the same search range as [1, 13].

Here the weighting factors $\alpha_1 = 0.8$, $\alpha_2 = 0.1$, $\alpha_3 = 0.1$.

Figure 3 presents the array pattern optimized by ILPSO. The array geometry for this case is given in Table 5.

In Figure 3, the best pattern found by ILPSO has PSLL $= -23.5$ dB, while the one in [1] was -23.17 dB and in [13] was -22.75 dB, comparing to -63.16 dB in [1] and -60 dB in [13]. The null level at 9° found by ILPSO is -73 dB.

5.3. Synthesis of Unequally Spaced Linear Array with PSLL and NULL Constraints

The final example is that of a 32-element array with nulls in three specified directions, the desired null directions are at 30°, 32.5°, and 35° respectively, but the range of adjacent elements' distance is set between 0.5λ and λ, same as the first example. This example is a complex multi-object optimization, and is more difficult than the above two examples.

Table 5. Element positions of the second example (normalized with respect to $\lambda/2$).

Element Number	**1**	**2**	**3**	**4**
Position	0.45	1.2695	2.1421	2.9724
Element Number	**9**	**10**	**11**	**12**
Position	7.6649	8.8106	9.9363	11.1742
Element Number	**5**	**6**	**7**	**8**
Position	3.8845	4.7906	5.6967	6.7281
Element Number	**13**	**14**	**15**	**16**
Position	12.6143	14.3151	16.2368	17.6184

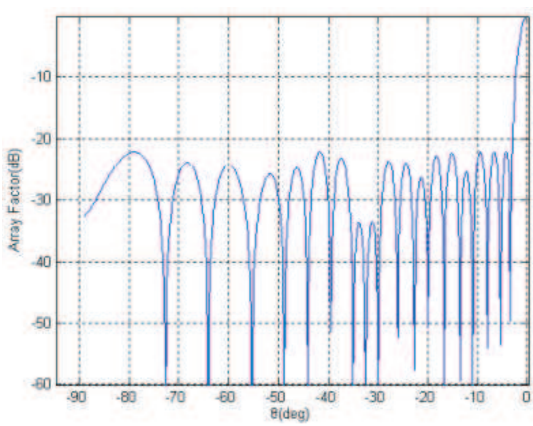

Figure 4. Best radiation pattern of the third example.

Table 6. Element positions of the third example (normalized with respect to $\lambda/2$).

Element Number	**1**	**2**	**3**	**4**
Position	0.5	1.5	2.5	3.5
Element Number	**9**	**10**	**11**	**12**
Position	8.9174	10.1844	11.4062	12.9946
Element Number	**5**	**6**	**7**	**8**
Position	4.5	5.5	6.5213	7.5504
Element Number	**13**	**14**	**15**	**16**
Position	14.4273	16.4008	18.3917	19.8073

The desired null level is at -60 dB. The desired beamwidth is set to 8°. The best result is shown in Figure 4, with a -22.1 dB PSLL, 7.6° beamwidth and -64 dB null level. The corresponding element positions and radiation pattern are shown in Table 6.

6. CONCLUSION

This paper illustrated the use of PSO algorithm in the pattern synthesis of antenna arrays. And a novel PSO algorithm named ILPSO is proposed. ILPSO can provide better global searching ability and can easily escape from the local optimum. In each cycle, for a better balance of searching ability, single PSO process works with a suitable maximal generation. The inheritance learning between cycles can strengthen the swarm diversity and keep the exploration ability.

ILPSO finds the global optima in most of eight 30-D benchmark functions. It works more effectively and adaptively on different problems when compared to nine other PSO variants [28]. ILPSO can significantly improve the PSO's performance and find the global optima on most benchmark functions whether they are rotated or not.

In addition, numerical examples of position-only synthesis of unequally spaced linear array have been studied and presented. When applying to the maximum sidelobe level suppression, ILPSO acquires at least equivalent performance to [4], but with smaller optimization generation. For the multi-objective synthesis with beamwidth, null point and sidelobe level suppression, ILPSO obtains better performance while compared to other proposals reported in [1, 13]. The results of ILPSO indicate its potential ability in the antenna designs for a wide class of electromagnetic applications.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 10876029), the National Natural Science Foundation of China (No. 60990320, No. 60990323) and the National 863 Project of China under Grant (No. 2009AA01Z230).

REFERENCES

1. Wang, W.-B., Q. Feng, and D. Liu, "Application of chaotic particle swarm optimization algorithm to pattern synthesis of antenna arrays," *Progress In Electromagnetics Research*, Vol. 115, 173–189, 2011.
2. Qu, Y., G. Liao, S.-Q. Zhu, and X.-Y. Liu, "Pattern synthesis of planar antenna array via convex optimization for airborne forward looking radar," *Progress In Electromagnetics Research*, Vol. 84, 1–10, 2008.
3. Dib, N. I., S. K. Goudos, and H. Muhsen, "Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays," *Progress In Electromagnetics Research*, Vol. 102, 159–180, 2010.
4. Lin, C., A.-Y. Qing, and Q.-Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Transactions on Antennas and Propagation*, Vol. 58, 2553–2561, 2010.
5. Kurup, D. G., M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 51, 2210–2217, 2003.
6. Vaitheeswaran, S. M., "Dual beam synthesis using element position perturbations and the G3-GA algorithm," *Progress In Electromagnetics Research*, Vol. 87, 43–61, 2008.
7. Zhang, S., S.-X. Gong, Y. Guan, P.-F. Zhang, and Q. Gong, "A novel IGA-EDSPSO hybrid algorithm for the synthesis of sparse arrays," *Progress In Electromagnetics Research*, Vol. 89, 121–134, 2009.
8. Chen, K., Z. He, and C. Han, "A modified real GA for the sparse linear array synthesis with multiple constraints," *IEEE Transactions on Antennas and Propagation*, Vol. 54, No. 7, 2169–2173, 2006.
9. Rocca, P., et al., "Ant colony optimisation for tree-searching-

based synthesis of monopulse array antenna," *Electronics Letters*, Vol. 44, 783–785, 2008.

10. Perez Lopez, J. R. and J. Basterrechea, "Hybrid particle swarm-based algorithms and their application to linear array synthesis," *Progress In Electromagnetics Research*, Vol. 90, 63–74, 2009.
11. Lanza Diego, M., J. R. Perez Lopez, and J. Basterrechea, "Synthesis of planar arrays using a modified particle swarm optimization algorithm by introducing a selection operator and elitism," *Progress In Electromagnetics Research*, Vol. 93, 145–160, 2009.
12. Boeringer, D. W. and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, Vol. 52, 771–779, 2004.
13. Goudos, S. K., et al., "Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control," *IEEE Antennas and Wireless Propagation Letters*, Vol. 9, 125–129, 2010.
14. Selleri, S., et al., "Differentiated meta-PSO methods for array optimization," *IEEE Transactions on Antennas and Propagation*, Vol. 56, 67–75, 2008.
15. Boeringer, D. W. and D. H. Werner, "Efficiency-constrained particle swarm optimization of a modified Bernstein polynomial for conformal array excitation amplitude synthesis," *IEEE Transactions on Antennas and Propagation*, Vol. 53, 2662–2673, 2005.
16. Kumar, B. P. and G. R. Branner, "Generalized analytical technique for the synthesis of unequally spaced arrays with linear, planar, cylindrical or spherical geometry," *IEEE Transactions on Antennas and Propagation*, Vol. 53, No. 2, 621–634, 2005.
17. Kazemi, S., H. R. Hassani, G. R. Dadashzadeh, and F. G. Gharakhili, "Performance improvement in amplitude synthesis of unequally spaced array using least mean square method," *Progress In Electromagnetics Research B*, Vol. 1, 135–145, 2008.
18. Kennedy, J. and R. Eberhart, "Particle swarm optimization," *IEEE Int. Conf. Neural Netw.*, Piscataway, NJ, 1995.
19. Valle, Y., et al., "Particle swarm optimization: Basic concepts,variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, Vol. 12, 171–195, 2008.
20. Clerc, M. and J. Kennedy, "The particle swarm-explosion,

stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, Vol. 6, 58–73, 2002.

21. Bergh, F. and A. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 8, 225–239, 2004.
22. Shi, Y. and R. C. Eberhart, "A modified particle swarm optimizer," *Proc. IEEE Congress Evol. Comput.*, 69–73, 1998.
23. Abido, A., "Optimal power flow using particle swarm optimization," *Int. J. Elect. Power Energy Syst.*, Vol. 24, 563–571, 2002.
24. Eberhart, R. and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *Proc. IEEE Congress Evol. Comput.*, Vol. 1, 84–88, Jul. 2000.
25. Esmin, A., G. Torres, and A. Zambroni, "A hybrid particle swarm optimizationapplied to loss power minimization," *IEEE Trans. Power Syst.*, Vol. 20, 859–866, 2005.
26. Poli, R., "Mean and variance of the sampling distribution of particle swarm optimizers during stagnation," *IEEE Transactions on Evolutionary Computation*, Vol. 13, 712–721, 2009.
27. Liang, J. J., P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," *Proc. Swarm Intell. Symp.*, Jun. 2005. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan.
28. Liang, J. J., et al., "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, Vol. 10, 281–295, Jun. 2006.