Front. Electr. Electron. Eng. China 2007, 2(4): 454–458
DOI 10.1007/s11460-007-0085-9

RESEARCH ARTICLE

JIN Ronghong, YUAN Zhihao, GENG Junping, FAN Yu, LI Jiajing

Pattern synthesis of antennas based on a modified particle swarm optimization algorithm

Abstract In order to overcome the drawbacks of standard particle swarm optimization (PSO) algorithm, such as prematurity and easily trapping in local optimum, a modified PSO algorithm is proposed, in which special techniques, as global best perturbation and inertia weight jump threshold are adopted. The convergence speed and accuracy of the algorithm are improved. The test by some benchmark problems shows that the proposed algorithm achieves relatively higher performance. Thereafter, the applications of the modified PSO in the radiation pattern synthesis of antenna arrays are presented.

Keywords particle swarm optimization (PSO) algorithm, premature convergence, array antennas, patterns synthesis

1 Introduction

Since the 1990s, optimization methods imitating natural biological behaviors have come into being. The particle swarm optimization (PSO) algorithm proposed by Eberhart and Kennedy in 1995 [1,2] was a simulation of behaviors of birds, fish and the human society. It obtains the optimum solutions through the coordination between individuals and sharing of information. As an evolution algorithm, PSO algorithm has many outstanding advantages, such as fast convergence, simple computation and easy implementation. However, the basic PSO algorithm suffers from problems such as prematurity, limited searching scope and trend to converge to local extremes and similar to other evolution algorithms.

In recent years, PSO algorithm has attached great attention. Various modified PSO algorithms have been proposed to overcome its weakness of prematurity and convergence to local extremes [3–8]. Inertia weights and constriction factors are used widely to improve the performance of the basic PSO algorithms at present [3,4]. Linear decreasing weight algorithm proposed by Shi [3] improved the convergence speed of PSO, but it is easily trapped by local minimums. Constriction factors [4] have been proved to ensure the convergence for the modified algorithms, but it converges too slowly. Fuzzy PSO algorithm and PSO algorithm with mutations have been proposed in Refs. [7] and [8] respectively, and the latter analyzed the impact of neighborhood and topologies in PSO algorithms, all of which improved the performance to some extent.

A modified PSO algorithm is proposed, in which global best perturbation and inertia weight jump threshold are adopted to improve the convergence speed. Compared with the modified algorithms in literatures mentioned above, it immensely improved the convergence speed and accuracy with simplicity and less computational burden.

The PSO algorithms were primarily employed to optimize continuous functions. As research proceeds, satisfactory results can be obtained in combined optimization problems. In the electromagnetic field, many significant results were also achieved. For example, the design of corrugation horn antennas was optimized [9] and array beam pattern was also synthesized by PSO algorithms [10].

Antenna arrays are capable of increasing the antenna gain and radiation capability, suppressing the side-lobe and forming nulls towards a specific direction. Various antenna pattern synthesis and optimization methods focus on controlling the beam and suppressing side-lobes by adjusting the excited amplitudes, phases and locations of elements. People pay much more attention to genetic algorithms because ordinarily analytical approaches or conjugate gradient methods are invalid to these problems [11–13]. As another smart evolution algorithm, a modified particle swarm algorithm was utilized for the synthesis of antenna array patterns in this paper. Compared with results by genetic algorithms, simulation results show that the proposed method is fast and stable with certain advantages.

Translated from *Chinese Journal of Radio Science*, 2006, 21(6): 873–878 [译自: 电波科学学报]

JIN Ronghong (✉), YUAN Zhihao, GENG Junping, FAN Yu, LI Jiajing
Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200030, China
E-mail: rhjin@sjtu.edu.cn

2 Basic particle swarm optimization algorithm

The PSO algorithm is an evolutional algorithm based on the group intelligence. Birds, fish and other species search food by themselves. Meanwhile, they get the exact location of food in a short period time by sharing information among the group. The PSO algorithm is proposed on the basis of such biological models.

It searches the best solution with a group of N particles in a D-dimension space. The iterations go on with reference to the best location so far of the single particle and that of the group. The location and speed vectors for each particle in a single round of iteration are defined as follows:

$$\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{id}), \boldsymbol{v}_i = (v_{i1}, v_{i2}, ..., v_{id}), 1 \leqslant i \leqslant N, 1 \leqslant d \leqslant N.$$

The formulas for the basic PSO algorithm are

$$\boldsymbol{v}_{id}^{k+1} = \omega \boldsymbol{v}_{id}^{k} + c_1 r_1^k \left(\boldsymbol{p}_{\text{best}}{}_{id}^{k} - \boldsymbol{x}_{id}^{k}\right) + c_2 r_2^k (\boldsymbol{g}_{\text{best}}{}_{d}^{k} - \boldsymbol{x}_{id}^{k}) \tag{1}$$

$$\boldsymbol{x}_{id}^{k+1} = \boldsymbol{x}_{id}^{k} + \boldsymbol{v}_{id}^{k+1} \tag{2}$$

The inertia weight ω, which was not included in initial formulas, has been found to possess a remarkable impact on the overall performance [3]. c_1 and c_2 are named learning factors or acceleration factors. $\boldsymbol{v}_{id}^{k}$ is the current speed vector of the ith particle while $\boldsymbol{x}_{id}^{k}$ is its location vector. $\boldsymbol{p}_{\text{best}_i}$ is the best location vector so far of the ith particle and $\boldsymbol{g}_{\text{best}}$ is the optimum location vector of the group. r_1 and r_2 are two random numbers within the interval of (0, 1). The upper index k stands for the kth iteration and the lower index d for the dth dimension.

An initial group, or a random solution, with random locations and speeds should be given at first. The particles fly in the space to search for the optimum solution through adjusting the speed vectors via learning by itself and the group. The optimum solutions for individuals and the group are updated after every iteration. The algorithm stops when the optimum solution is obtained or the maximum number of iteration has been met. Then the optimum solution for the group is regarded as the optimum solution.

The basic PSO algorithm often converges to a local extreme, with the presentation that particles are gathering more and more closely while the global optimum particle remains the same during a long period, which indicates that PSO algorithm is stagnated. Thus, a method to jump out of the local extreme is demanded on this condition, so that the dynamic of the group can be maintained and the whole space can be searched sufficiently.

3 Modified PSO algorithm

3.1 Introduction to the algorithm

To overcome the problems of prematurity and easy trapping into a local extreme, a modified PSO algorithm is proposed. The basic idea is to introduce a fractional perturbation to the global best particle to alter its location while updating other particles with the origin method in the iteration. This algorithm demonstrates its great performance in the later tests.

The iteration formulas for the global best particle are

$$\boldsymbol{v}_{id}^{k+1} = 0 \tag{3}$$

$$\boldsymbol{x}_{id}^{k+1} = \boldsymbol{x}_{id}^{k} + \boldsymbol{x}_{id}^{k} P_p (r - 0.5) \tag{4}$$

The speed of the global best on each dimension is set to 0 while the location vector is iterated with perturbation around its origin location. $\boldsymbol{x}_{id}^{k}$ is the location of the ith particle on the dth dimension in the kth iteration. P_{p} is the perturbation parameter and r is random number within the interval of (0, 1).

Equations (1) and (2) are still valid for un-global best particles. However, according to Eqs. (1) and (2), if particle j is the best particle in the kth iteration, then

$$\boldsymbol{x}_{j}^{k} = \boldsymbol{p}_{\text{best}}{}_{j}^{k} = \boldsymbol{g}_{\text{best}}{}^{k} \tag{5}$$

The inertia weigh decreases as the iteration continues, which leads to

$$\boldsymbol{v}_{id}^{k+1} \approx 0,\ \boldsymbol{x}_{jd}^{k+1} = \boldsymbol{x}_{jd}^{k+1}.$$

This particle remains in the later iterations and other particles converge to it. Then it is hard to jump out of the local extreme to converge to the global best if this particle is unfortunately a local extreme.

With Eqs. (3) and (4), the stopped particle is activated. Because the perturbation is performed around the current global best, there is a higher possibility to find a global best compared with searching randomly in the whole space.

The inertia weigh jump threshold is adopted in iterations for the un-global best particles. The weight decreases linearly before the threshold is met, and is reset when the threshold is met, thus ensuring the dynamic of the group.

The weight jump threshold is set the same as the maximum iteration rounds in the basic LDW-PSO algorithm. The convergence performance is not satisfactory due to the uncertainty of the maximum number of iterations. ω varies too slowly with a large maximum number of iterations while the iteration stops with a small number, leading to a failure in searching.

In this paper, the threshold for weight jump and the maximum number of iterations are two separated variables, which results in an increase in the controllability and flexibility of the algorithm. Therefore, the maximum number of iterations can be defined with a large number to ensure the convergence, and a small number is assigned to the jump threshold to accelerate the convergence speed.

ω complies with the following equation before the iteration times meet the threshold

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \frac{I_{iter}}{\omega_{run}} \tag{6}$$

where ω_{max} and ω_{min} are the maximum and minimum inertia weight respectively, which confine the interval of the weight. I_{iter} is the time of iterations and ω_{run} is the jump threshold. When I_{iter} meets the threshold, the following formulas are applied

$$I_{iter} = 0,\ \omega_{max} = \omega_{max}\omega_p \tag{7}$$

where ω_p is the coefficient of the initial value for the weight, which is usually set to 1. For some multi-dimensional complicated problems, it might be better to set a larger ω_p.

The steps for the modified PSO algorithm are given as below.

1) Set random speeds and locations for the particles.

2) Compute the fitness function for all the particles. The current location of a particle is $\boldsymbol{p}_{best}$ and the location of the best particle is $\boldsymbol{g}_{best}$.

3) If the convergence criteria are met, forward to 6), otherwise, execute the next step.

4) Update the speed and location for each particle according to the two different scenarios.

a) If the particle is not the best particle, update its speed and location with Eqs. (1) and (2), where ω is computed with Eqs. (6) and (7).

b) If the particle is the best particle, update its speed and location with Eqs. (3) and (4). Compute the fitness function with the updated particle. If the fitness equals to $\boldsymbol{p}_{best}$, reset $\boldsymbol{p}_{best}$. And if the fitness equals to $\boldsymbol{g}_{best}$, reset $\boldsymbol{g}_{best}$.

5) Move to 3).

6) Output $\boldsymbol{g}_{best}$. The algorithm terminates.

Perturbation, the amplitude of which is controlled by the perturbation coefficient, is used to change the location of the best particle to increase the possibility of finding a global best. Thus the algorithm is named as Perturbation PSO (PPSO). Furthermore, the jump weight increases the convergence speed while ensuring the convergence.

3.2 Algorithm performance analysis

The performance of the proposed algorithm is illustrated by a series of typical testing functions as Sphere, Rosenbrock, Rastrigin, Griewank and Schaffer F6, the specific forms and intervals of which for parameters are set according to Ref. [6]. The goal is to seek the minimum of these functions.

The experiment is designed as follows.

A group of 30 particles, with a maximum of 3 000 iterations, running for 20 times and the results are averaged. All test functions are of 30 dimensions except Schaffer F6 which is of two dimensions. E_{rr} is the requirement for convergence accuracy.

The results are shown in Table 1. The first column is the algorithm convergence rate under the requirements of the convergence accuracy with the maximum number of iterations, and the second is the average iteration times to meet the convergence accuracy. Set D is the result obtained with a group of fairly good compact factors in Ref. [6] and PPSO1–4 are results by various combinations of parameters according to Table 2.

It is even more important to select the proper parameters since they oppose to an obvious impact on the optimization performance and new parameters have been introduced in the modified algorithm. In Table 2, V_{max}/X_{max} stands for the maximum speed, ω_{run} the weight jump threshold, ω_{max} and ω_{min} the weight variation scope, P_p the perturbation coefficient, c_1 and c_2 the learning factor, and ω_p the initial weight coefficient.

Table 2 shows several typical parameter combinations. PPSO1 can be regarded as an inappropriate combination and PPSO2 is a fine setting. PPSO3 and PPSO4 are algorithms without weight jump. It can be seen from Table 1 that compared to PSO with set D, PPSO2 leads to a better performance. And convergence is accelerated on the basis of global convergence.

As having anticipated before, algorithms without weight jump cannot guarantee the convergence with a fast convergence rate. Although PPSO3 ensures the convergence, it converges much slower than the modified algorithm. PPSO4 converges fast but it does not guarantee the convergence.

4 Synthesis of the array patterns

The array pattern can approach the desired shape by adjusting the exciting current amplitude and phase shift of each element in a uniform linear array with N isotropic elements.

From the antenna theories, the far field pattern of a uniform linear array is

$$F(\theta) = \sum_{n=1}^{N} I_n \mathrm{e}^{\mathrm{j}(nkd\cos\theta + \phi_n)} \tag{8}$$

Table 1 Test results under specified convergence accuracy

Test function	Sphere, $E_{rr} < 0.01$	Rosenbrock, $E_{rr} < 100$	Rastrigin, $E_{rr} < 100$	Griewank, $E_{rr} < 0.1$	Schaffer F6, $E_{rr} < 0.000\,01$
Set D	100%, 226.5	100%, 199.5	100%, 131	100%, 147	85%, 272
PPSO1	100%, 324.6	100%, 283.2	100%, 109	100%, 247	100%, 147.1
PPSO2	100%, 139.5	100%, 180.9	100%, 71.0	100%, 133.1	100%, 104.9
PPSO3	100%, 265.8	100%, 258.0	100%, 91.8	100%, 273.5	100%, 123.5
PPSO4	80%, 137.1	45%, 135.9	100%, 86.6	75%, 129.7	55%, 124.8

Table 2 Combination of the parameters

	V_{max}/X_{max}	ω_{run}	ω_{max}	ω_{min}	p_p	c_1,c_2	ω_p
PPSO1	1	100	0.9	0.4	1.3	2	1
PPSO2	1	100	0.9	0.1	1.3	1.5	1
PPSO3	1	max_run = 3 000	0.9	0.1	1.3	1.5	1
PPSO4	1	max_run = 150	0.9	0.1	1.3	1.5	1

where N is the number of elements, I_n is the current amplitude and ϕ_n is the current phase.

4.1 Sample 1

The normalized desired far field pattern is

$$F_d(\theta)=\begin{cases}1, & \pi/4\leqslant\theta\leqslant 3\pi/4,\\ 0, & \text{others}\end{cases} \tag{9}$$

The PPSO2 algorithm in the last section is used to optimize the current phase of the element between 0 and π. The group consists of 40 particles and iteration runs for 80 times.

The fitness function is defined as follows:

$$f=\alpha\sum_{m=1}^{M}\left|f(\theta_m)-f_d(\theta_m)\right|+\beta f_{m\ \mathrm{std}} \tag{10}$$

where θ_m is the sample points, α and β are weights. The first term $\sum_{m=1}^{M}\left|f(\theta_m)-f_d(\theta_m)\right|$ is the sum of the absolute error in the sampling points and the second term $f_{m\ \mathrm{std}}$ is the standard deviation of the errors. In this sample, $\alpha=\beta=0.5$, and the sampling points are (1°, 2°,..., 180°). The results are shown in Fig. 1, compared with GA from Ref. [11].

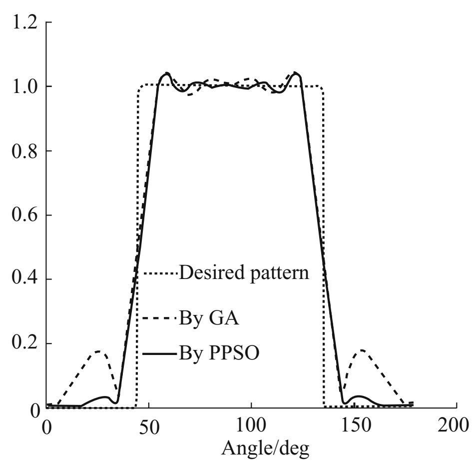

Fig. 1 Ten element array pattern

It can be seen that compared with the curve obtained from GA, the pattern by PSO is more flat and closer to the target curve, with better main-lobe ripple and lower side-lobe.

4.2 Sample 2

Suppose a uniform linear array consisting of 20 elements, with $\lambda/2$ between adjacent elements, the current phase of which is 0 and the amplitude is symmetrically distributed. The pattern is synthesized by optimizing the current amplitude.

The goal is to form a beam with side-lobes lower than −15 dB and nulls of −95 dB at 30°, 40°, 50°, 60° and 70° respectively.

Setting for the PPSO algorithm is the same as those of PPSO2 in the last section. The group of 100 particles runs for a maximum of 100 times with current amplitude varying between 0 and 1. The fitness function is defined according to Ref. [13] as follows:

$$f=\alpha\left|\mathrm{MSLL}-\mathrm{SLVL}\right|+\beta\left|\mathrm{NULL\_PAT}-\mathrm{NLVL}\right| \\ +\gamma\mathrm{NULL\_STD} \tag{11}$$

where MSLL is the highest side-lobe, SLVL is the designed side-lobe level, NULL_PAT is the average NULL, NLVL is the designed NULL and NULL_STD is the standard deviation for nulls.

Set $\alpha=1$, $\beta=0.2$, $\gamma=1.1$.

The optimized current amplitudes are 0.266 0, 0.756 4, 0.428 1, 0.679 8, 0.959 7, 0.877 8, 0.682 6, 0.873 0, 0.966 5, and 1.000 0 respectively, which leads to the array pattern in Fig. 2.

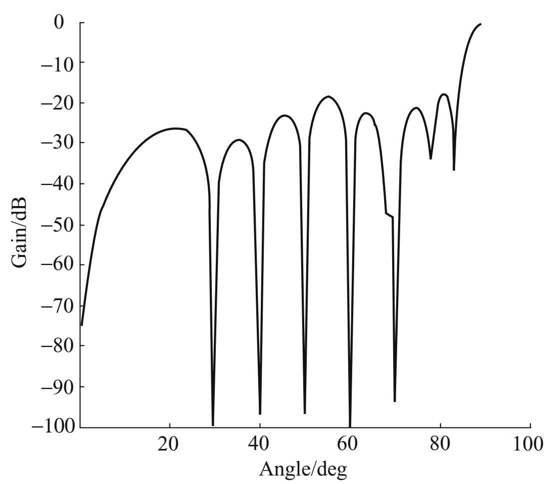

Fig. 2 Normalized array pattern of the linear array

It can be seen that the array pattern meets the design goal, with the average null of −95 dB and the highest side-lobe of −17.85 dB.

To reach the same goal, GA needs a larger group, more iterations and longer computation time. The GA in Ref. [13] using a group of 120 individuals got three nulls of −80 dB after 60 generations. However, only 50 particles and 50 iterations are necessary to get the same results with PPSO algorithm in this paper.

5 Conclusion

The PSO algorithm, as a novel optimization algorithm, provides a new solution for multidimensional, nonlinear, and multi-extreme complex problems. The basic PSO algorithm, although simple and good, suffers from flaws such as prematurity etc. The modified PSO algorithm proposed in this paper reduces the possibility of trapping in a local extreme and overcomes prematurity by perturbation of the global best. And the inertia weight jump threshold is used to accelerate the convergence speed. Through testing by the classical testing functions, the advantages in optimization accuracy and speed have been demonstrated. To illustrate the practicability and efficiency of the PPSO algorithm, the proposed method is used to synthesize array patterns and a satisfactory performance is obtained, demonstrating its capacity of dealing with complicatedly multidimensional problems.

References

1. Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, IEEE Service Center, 1995, 4: 1 942–1 948
2. Eberhart R C, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan. IEEE Press, 1995, 39–43
3. Shi Y, Eberhart R C. A modified particle swarm optimizer. In: Proceedings of IEEE Congress on Evolutionary Computation. N.J, Piscataway: IEEE Press, 1998, 69–73
4. Eberhart R C, Shi Y. Comparing inertia weights and constriction factors. In: Proceedings of IEEE Congress on Evolutionary Computation, La Jolla, CA, USA. IEEE Press, 2000, 48–88
5. Eberhart R C, Shi Y. Fuzzy adaptive particle swarm optimization. In: Proceedings of IEEE Congress on Evolution Computation, Seoul, South Korea. IEEE Press, 2001, 101–106
6. Carlisle A, Dozier G. An off-the-shelf PSO. In: Proceedings of 2001 Workshop on Particle Swarm Optimization. Indianapolis: Purdue School of Engineering and Technology, 2001, 1−6
7. Jancic S A, M Grundy I. Particle swarm optimization with mutation. The 2003 Congress on Evolutionary Computation, 2003, (2): 1 425–1 430
8. Kennedy J, Mendes R. Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms. In: Proceedings of 2003 IEEE International Workshop on Soft Computing in Industrial Applications, New York, USA. IEEE Press, 2003, 45–50
9. Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics. IEEE Transactions on Antennas and Propagation, 2004, 52(2): 397–407
10. Gies D, Rahmat-Samii Y. Reconfigurable array design using parallel particle swarm optimization. IEEE International Symposium on Antennas and Propagation Society, 2003, (1): 177–180
11. Ren S H, Wu Z Z. Application of genetic algorithms in the synthetic design for the radiation pattern of an array antenna. Chinese Journal of Radio Science, 1996, 11(4): 62–67 (in Chinese)
12. Ma Y H. Synthesis of the array antennas using genetic algorithm. Chinese Journal of Radio Science, 2001, 16(2): 172–176 (in Chinese)
13. Fan Y, Jin R H. Pattern synthesis of antennas based on a novel genetic algorithm. Chinese Journal of Radio Science, 2004, 19(2): 182–186