Design of Yagi–Uda antennas using comprehensive learning particle swarm optimisation

S. Baskar, A. Alphones, P.N. Suganthan and J.J. Liang

Abstract: A method of using particle swarm optimisation (PSO) algorithms to optimise the element spacing and lengths of Yagi–Uda antennas is presented. SuperNEC, an object-oriented version of the numerical electromagnetic code (NEC-2) is used to evaluate the performance of various Yagi–Uda antenna designs. In order to show the capabilities of the PSO algorithm in Yagi–Uda antenna design, three different antenna design cases are optimised for various performance specifications. The three objectives considered are gain only, gain and input impedance only, and gain, input impedance and relative sidelobe level (rSLL). To alleviate the premature convergence problem of PSO, a novel learning strategy is employed. Each design problem is optimised using three variants of PSO algorithms, namely the modified PSO, fitness-distance ratio PSO (FDR-PSO), and comprehensive learning PSO (CLPSO). For the purpose of comparison and benchmarking, equally spaced arrays, genetic algorithm optimised antenna design, and computational intelligence optimised antenna design are considered. The results clearly show that the CLPSO is a robust and useful optimisation tool for designing Yagi antennas for the desired target specifications.

1 Introduction

Yagi–Uda antennas are widely used in the VHF/UHF bands and have major applications in television signal reception and direction finding [1]. Also, in recent years quasi-Yagi antennas on printed substrates at microwave and millimetre-wave frequencies are being attempted for low cost communication and radar applications [2–4]. In the design and synthesis of Yagi antennas, the goal is to find a radiating structure to satisfy a set of performance criteria such as gain, maximum sidelobe level (SLL), beam width, input impedance and physical size. The antenna design specifications are known to be highly multimodal functions of the physical dimensions of the antenna elements and their spacing.

Several researchers investigated the problem of optimising the performance of Yagi–Uda antennas. Ehrenspeck and Poehler [5] performed extensive laboratory experiments to accurately determine the spacing and lengths of the elements to provide maximum gain. Cheng *et al.* [6–8] applied several gradient-based techniques to optimise spacing and length of Yagi–Uda antennas for maximum gain. In general, the performance of gradient-based techniques is largely dependent on the choice of initial solution. It is difficult to provide good initial solutions for the gradient-based methods. This design problem becomes even more difficult when other performance measures such as impedance, rSLL and beam width are also considered in the optimisation process.

IEE Proceedings online no. 20045087

doi:10.1049/ip-map:20045087

Paper first received 22nd August 2004 and in final revised form 16th March 2005

The authors are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

E-mail: alphones@ieee.org

Population-based evolutionary algorithms are effective in solving multimodal function optimisation problems, as they explore multiple solutions simultaneously. Altshuler and Linden [9] applied genetic algorithms (GA) to design wire antennas. The optimised designs were verified experimentally to meet design specifications. Jones *et al.* [10] successfully applied binary coded genetic algorithms to the design of Yagi–Uda antennas to maximise the gain and to achieve other desired antenna characteristics. They formulated this design as an unconstrained single objective maximisation problem by combining different objectives using scaling factors. Recently, Venkatarayalu and Ray [11, 12] presented single and multi-objective formulations for the design of Yagi–Uda antennas using a computational intelligence (CI) method that can handle constraints and objectives separately via Pareto ranking thereby eliminating the need for scaling and aggregation. Even though their results are better than GA optimised antennas [10], the number of objective function evaluations performed was approximately 20 times higher.

Recently, particle swarm optimisation (PSO) was proposed as a simple tool for solving function optimisation problems. PSO is an evolutionary computation technique based on the social behaviour of bird flocks and fish schools. PSO has been shown to be effective in optimising complex multidimensional problems [13–15]. This technique has been applied successfully to solve electromagnetic design problems such as reconfigurable phase-differentiated array design [16], corrugated horn antenna [17] and phased-array synthesis [18].

To the best of our knowledge, the performance of PSO in design optimisation of Yagi–Uda antennas has not been investigated before. To improve the performance of the basic PSO, a novel learning strategy has been employed.

2 Yagi–Uda antenna design problem

An N-element end wire radiator Yagi–Uda antenna consists of a linear array with one dipole, one reflector, and $N-2$

directors. The parasitic antenna array structure of a Yagi–Uda antenna with four elements is shown in Fig. 1. An N-element Yagi–Uda antenna has $2N-1$ variables that determine the various antenna characteristics, apart from the radius of the elements. The design variables of an N-element Yagi–Uda antenna are $X = [L_0, L_1, L_2, \ldots, L_{N-1}, S_0, S_1, S_2, \ldots, S_{N-2}]$, where $2L_i$ is the length of the ith element, and S_i is the spacing between the $(i+1)$th and the ith elements.

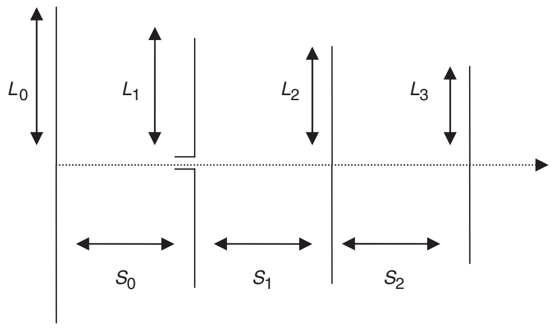

Fig. 1 *Four element Yagi–Uda antenna*

The objective is to develop an antenna to satisfy performance specifications such as gain, input impedance, sidelobe level, beam width, front-to-back ratio, and size. As is common in a Yagi–Uda antenna, the input impedance is relatively low (20–30 Ω), the optimisation of antenna parameters to meet input impedance is critical so that the balun to match with 50 Ω can be avoided. The following objective function has been shown to represent the desired antenna performance characteristics [10]:

$$\mathrm{O}(X) = a\,G(X) - b|50\text{-}\mathrm{Re}(Z(X))| - c|\mathrm{Im}(Z(X))| + d\ \max(\mathrm{rSLL}) \quad (1)$$

This function rewards an antenna design parameter vector X for having a high gain G and maximum rSLL and penalises the design if the real part of the input impedance Z does not equal 50 Ω or if the imaginary part of the input impedance does not equal zero. The constants a, b, c and d are weights to control the contribution from each term to the overall objective function. Jones found that making a 20–40 times greater than b and c and 10–20 times greater than d yielded a good balance between gain, impedance and rSLL [10].

3 Particle swarm optimisation

Particle swarm optimisation developed by Eberhart and Kennedy [13] is one of the evolutionary computation techniques. PSO like genetic algorithm (GA) is a population based optimisation algorithm. The swarm initially has a population of random solutions. Each potential solution, called a particle, is given a random velocity and is flown through the problem space. The particles have memory and each particle keeps track of its previous best position, called pbest and corresponding fitness. The swarm remembers another value called gbest, which is the best solution discovered by the swarm. Velocity and position of the particles are changed according to (2) and (3) respectively [14]

$$V_i(d) = \omega V_i(d) + c_1 * rand * (P_i(d) - X_i(d)) + c_2 * rand * (P_g(d) - X_i(d)) \quad (2)$$

$$X_i(d) = X_i(d) + V_i(d) \quad (3)$$

where $V_i(d)$ and $X_i(d)$ represent the velocity and position of the dth dimension of the ith particle respectively and $rand$ is a uniform random number in the range [0, 1]. $P_g(d)$ and $P_i(d)$ are the dth dimensional positions of the gbest and pbest respectively.

Even though the PSO algorithm is simple in concept, easy to implement and computationally efficient, its performance may not be satisfactory in solving complex multi-modal optimisation problems. Many researchers have worked on improving the performance of PSO by incorporating a number of modifications. Clerc [15] indicates that the use of a constriction factor may be useful to insure convergence of the particle swarm algorithm. Suganthan [19] proposed PSO with a neighbourhood operator where a local version is executed first followed by a global version. Peram *et al.* [20] developed a variant of PSO, fitness distance ratio PSO with near neighbourhood interactions. In the FDR-PSO algorithm, each particle learns from the experience of the neighbouring particles with better fitness than itself. The FDR-PSO algorithm selects only one other particle at a time when updating each velocity dimension.

Ratnaweera *et al.* [21] introduced the time-varying acceleration coefficients in addition to the time-varying inertia weight. Some researchers investigated hybridisation by combining PSO with other evolutionary operators in addition to local search techniques to improve the original PSO's performance [22]. Recently, novel learning methods were proposed by the authors [23] to alleviate premature convergence problem of the original PSO when solving complex multimodal function optimisation problems. Much improved performance of the new learning strategy is demonstrated on benchmark unconstrained optimisation problems [23]. The new learning strategy is discussed briefly in the following Section.

4 Comprehensive learning PSO (CLPSO)

Though there are numerous PSO variants, premature convergence when solving a complex multimodal problem is still the main deficiency of most PSO-based algorithms. In the original PSO, each particle learns from its pbest and gbest simultaneously. Restricting the social learning aspect to only the gbest in original PSO appears to be somewhat an arbitrary decision. Furthermore, all particles in the swarm learn from the gbest even if the current gbest is far from the global optimum. In such situations, particles may be attracted easily and trapped into an inferior local optimum if the search environment is complex with numerous local solutions as in the Yagi–Uda antenna design problem.

As the fitness value of a particle is decided by all dimensions, a particle which has discovered the value corresponding to the global optimum in one dimension may have a low fitness value because of the poor solutions in other dimensions. This good genotype may be lost in this situation. In order to prevent this, novel learning strategies were introduced [23]. They differ in two main aspects compared to many present PSO variants:

(i) Instead of learning from two exemplars, namely pbest and gbest in every generation in the original PSO as in (2), each dimension of a particle learns from just one exemplar for a few iterations.

(ii) Instead of learning from the pbest and gbest for all dimensions, each dimension of a particle in general learns from a different pbest for different dimensions for a few iterations.

In CLPSO, the particle's own pbest and other particles' pbests are used as the particle's exemplars. Each particle learns from potentially all particles' pbests in the swarm. During the search process, we do not know which dimensions of each particle's pbest are good or bad. Therefore, each dimension of a particle has equal chance to be learnt by other particles. For each particle, some dimensions of other particles' pbests are chosen randomly according to a probability P_c as social exemplars to learn from, while other dimensions learn from its pbest, the cognitive exemplar. The flowchart of the CLPSO algorithm is given in Table 1.

The learning from different particle's pbests for different dimensions increases the particle's initial diversity and enables the swarm to overcome premature convergence problem. By inspecting the expressions in (2) and (3), it is clear that PSO performs based on a variable update and not a vector update of the positions of the particles, that is, each dimension is updated independently. Hence, learning each dimension of a particle from a different pbest exemplar is within the spirit of the original PSO.

5 Simulation results

In order to compare the performances of several variants of the PSO algorithm on Yagi–Uda antenna design optimisation, three different antennas with different objectives are tested. The obtained results are compared with published results from equally-spaced arrays, genetic algorithm optimised antennas [10], and computational intelligence optimised antennas [11, 12]. The design specifications for the comparisons are used from Jones [10]. First a six-element array is optimised solely for maximum gain and then for both gain and input impedance. In the second example, a 15-element array is optimised for maximum gain and appropriate impedance. The third example optimises a four-element array for gain, input impedance, and sidelobe levels.

The spacing between elements is allowed to vary between $0.10\,\lambda$ and $0.45\,\lambda$, and the length of each element is allowed to vary between $0.15\,\lambda$ and $0.35\,\lambda$. The best antenna parameters obtained out of ten different runs are presented for all test cases. In this study, SuperNEC, an object oriented version of the numerical electromagnetic code (NEC-2) has been used in the analysis of the antenna. MATLAB-based interface is used for editing structures and viewing simulation results [24].

For each test case, experiments are conducted using three PSO-based algorithms, namely modified PSO [14], FDR-PSO [20], and CLPSO [23]. Each antenna element is divided into seven segments for simulation purposes. For meaningful comparison of results with other methods, only the length and spacing of optimised antennas are taken and SuperNEC is employed to obtain the corresponding performance characteristics.

Table 1: CLPSO flow chart [23]

Initialise the swarm: Initialise positions and associated velocities between $\pm V_{max}$ of all particles in the population randomly in the D-dimensional search space, where $V_{max} = 0.25\,(X_{max} - X_{min})$. Evaluate the fitness values of all particles. Set the current position as pbest and the current particle with the best fitness value in the whole population as the gbest.

For $k = 1$ to *max_iteration*

$$\omega(k) = \frac{(\omega_0 - 0.2)\times(\max\_gen - k)}{\max\_gen} + 0.2 \text{ and } \omega_0 = 0.9 \tag{4}$$

If Mod(k, 5) = 1 //assign dimensions every 5 generations

For $i = 1$ to ps, // ps is the population size

$rc = \text{randperm}(D)$; //random permutations of dimensions for learning

$b_i = \lceil P_c - rand(1, D) \rceil$ // $\lceil\ \rceil$ represents ceiling operator

$$f_i = \lceil rand(1, D) * (ps - 1) \rceil \tag{5}$$

EndFor i

EndIf

For $i = 1$ to ps //updating velocity, position of each particle

For $d = 1$ to D //updating V, X of each dimension

If $b_i(d) == 1$

$$V_i(d) = \omega_k * V_I(d) + rand() * (pbest_{fi(d)}(d) - X_i(d)) \tag{6a}$$

Else

$$V_i(d) = \omega_k * V_I(d) + rand() * (pbest_I(d) - X_i(d)) \tag{6b}$$

EndIf

$V_i(d) = \min(\lambda V_{max}(d), \max(-V_{max}(d), V_i(d)))$ //Limit the velocity

$$X_i(d) = X_i(d) + V_i(d) \tag{7}$$

EndFor d

If $X_i \in [X_{min}, X_{max}]$

Calculate the fitness value of X_i

Update pbest, gbest if needed and record gbest_*id*

EndIf

EndFor i

Stop if a stop criterion is satisfied

EndFor k

For the execution of different PSO algorithms, the following parameters are used. Inertia parameter ω is varied from 0.9 to 0.2 linearly with generations. V_{max} is set at 20% of the maximum range allowed for each dimension. The particles are allowed to fly without any physical restrictions. However, particles that roam outside solution space are not evaluated for fitness [17]. Acceleration constants c_1 and c_2 are set at unity. Normally, a low value of P_c is better to alleviate premature convergence problems [23]. Hence P_c is set to 0.1 in all simulations.

5.1 Six-element Yagi–Uda antenna design

In six-element antenna design, there are 11 design variables (6 elements for lengths and five elements for spacing). The population size is 10 and the number of function valuations, Feval, is 3500. The radius of each element is held constant at 0.003369 λ. The optimum results obtained using the GA as reported in [10] and PSO algorithms are shown in Table 2. The optimum gain obtained using the modified PSO algorithm is less than the gradient optimised and GA optimised antennas [10]. The maximum gain obtained by the other PSO variants is higher than the gain obtained by the GA and gradient optimised antennas.

The convergence characteristics of different PSO algorithms are shown in Fig. 2. The Figure clearly indicates the premature convergence problem of the modified PSO and FDR-PSO algorithms. The modified PSO and FDR-PSO algorithms are faster during the initial iterations. After 100 generations there was no further improvement in the solution quality as these algorithms converged to local optima. But in the CLPSO algorithm, we could observe the improvements in the objective value even after 250 generations. The main reason for this is that all the dimensions are not learning from just one exemplar, instead some of the particle's dimensions are learning from its own pbest and other dimensions from other particles' pbests for few generations, thereby retaining diversity among the particles. Hence, the CLPSO algorithm is not affected by premature convergence when solving this problem.

In general, in gain optimised antennas, the real part of impedance is very low and the imaginary part of impedance is very high. Hence, these antennas are difficult to match, as the desired value of impedance is 50 Ω for most of the applications. Next for the same six element antenna design the impedance term is also taken into account in the objective function. The a, b and c values in (1) are taken as 30, 1 and 1 respectively. The optimum results obtained using the GA [10] and PSO algorithms are given in Table 3.

The results obtained using different variants of PSO algorithms satisfy the 50 Ω impedance requirement compared to the GA optimised antenna. Except for the CLPSO, the optimum gain achieved by the other PSO algorithms for this case is slightly lower compared to the GA optimised antenna. In CLPSO, both gain and impedance are better than the GA optimised antenna. In general, the performance of CLPSO is better than the other PSO variants and GA optimised antenna [10].

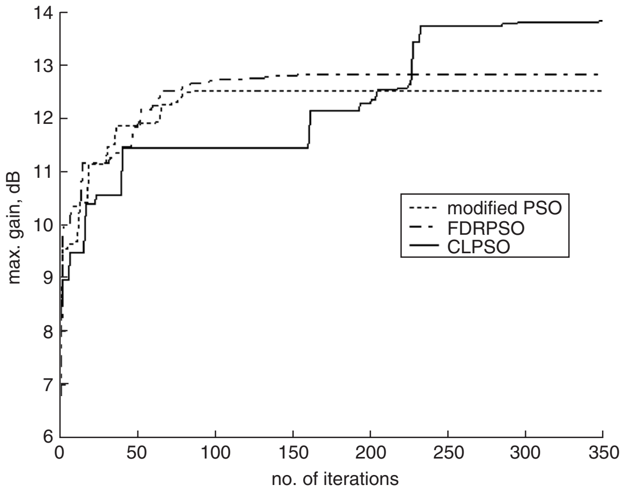

Fig. 2 *Convergence characteristics of various PSO algorithms*

5.2 Fifteen-element Yagi–Uda antenna design

To illustrate the capability of PSO in antenna design optimisation with a large number of variables, 15-element antenna design is considered. This antenna is optimised for gain and input impedance, keeping the radius of the element the same as before. Population size and Feval are set at 20 and 7500 respectively. The results obtained using the GA [10], CI [11, 12] and PSO algorithms are given in Table 4.

These results clearly indicate the superior performance of PSO algorithms compared to the GA. The maximum gain obtained by the CLPSO optimised antenna is 6.7% higher than the gain obtained in the GA optimised antenna. The 50 Ω requirement is also satisfied almost exactly by the PSO algorithms. Even though the gain obtained in the CI optimised antenna is slightly higher than in the CLPSO optimised antenna, the 50 Ω requirement is not satisfied. In addition, the computation time of the CI algorithm is very high, as the number of function evaluations is 160,000. As compared to the CI algorithm, CLPSO algorithm is able to yield good compromise solution with about 20 times less

Table 2: Results of gain optimised six element Yagi–Uda antenna designs

	Gradient optimised [10]		GA [10]		Modified PSO		FDR-PSO		CLPSO	
Element	*L*	*S*	*L*	*S*	*L*	*S*	*L*	*S*	*L*	*S*
1	0.238λ	–	0.252λ	–	0.247λ	–	0.240λ	–	0.242λ	–
2	0.226λ	0.25λ	0.301λ	0.101λ	0.265λ	0.101λ	0.262λ	0.241λ	0.234λ	0.183λ
3	0.218λ	0.289λ	0.221λ	0.321λ	0.223λ	0.246λ	0.219λ	0.273λ	0.221λ	0.228λ
4	0.215λ	0.406λ	0.219λ	0.274λ	0.215λ	0.389λ	0.216λ	0.313λ	0.212λ	0.415λ
5	0.217λ	0.323λ	0.210λ	0.428λ	0.213λ	0.391λ	0.211λ	0.429λ	0.210λ	0.405λ
6	0.215λ	0.422λ	0.211λ	0.435λ	0.150λ	0.167λ	0.213λ	0.389λ	0.214λ	0.384λ
Gain (dB)	12.80		13.55		12.81		13.77		13.84	
Z (Ω)	8.34+j22.60		5.678+j217.680		2.39+j101.23		12.23+j119.7		3.9+j24.19	

Table 3: Results of gain and impedance optimised six element Yagi–Uda antenna designs

Element	GA [10] L	GA [10] S	Modified PSO L	Modified PSO S	FDR-PSO L	FDR-PSO S	CLPSO L	CLPSO S
1	0.239λ	–	0.232λ	–	0.237λ	–	0.236λ	–
2	0.225λ	0.182λ	0.231λ	0.271λ	0.222λ	0.269λ	0.231λ	0.257λ
3	0.224λ	0.152λ	0.222λ	0.216λ	0.221λ	0.185λ	0.221λ	0.192λ
4	0.217λ	0.229λ	0.213λ	0.262λ	0.212λ	0.254λ	0.215λ	0.296λ
5	0.211λ	0.435λ	0.204λ	0.415λ	0.210λ	0.365λ	0.211λ	0.334λ
6	0.220λ	0.272λ	0.213λ	0.356λ	0.215λ	0.328λ	0.214λ	0.345λ
Gain (dB)	12.57		12.38		12.47		12.65	
Z (Ω)	52.16−j10.09		50+j0.0009		50−j0.0003		50.013−j0.013	

Table 4: Results of gain and impedance optimised fifteen-element Yagi–Uda antenna designs

Element	Method GA [10] L (in terms of λ)	S	CI [11, 12] L (in terms of λ)	S	PSO L (in terms of λ)	S	FDR-PSO L (in terms of λ)	S	CLPSO L (in terms of λ)	S
1	0.236	–	0.235	–	0.234	–	0.236	–	0.239	–
2	0.230	0.249	0.227	0.196	0.224	0.310	0.231	0.251	0.226	0.168
3	0.221	0.155	0.224	0.238	0.223	0.125	0.219	0.196	0.222	0.171
4	0.205	0.185	0.215	0.142	0.213	0.204	0.216	0.296	0.216	0.260
5	0.216	0.191	0.204	0.231	0.199	0.396	0.211	0.326	0.210	0.311
6	0.210	0.252	0.212	0.447	0.201	0.252	0.204	0.306	0.201	0.216
7	0.210	0.442	0.206	0.395	0.196	0.203	0.184	0.247	0.210	0.262
8	0.189	0.431	0.203	0.371	0.207	0.242	0.201	0.197	0.205	0.378
9	0.191	0.362	0.201	0.441	0.207	0.262	0.206	0.180	0.197	0.336
10	0.200	0.205	0.202	0.433	0.204	0.437	0.195	0.368	0.204	0.376
11	0.204	0.268	0.206	0.445	0.191	0.403	0.196	0.277	0.199	0.324
12	0.215	0.414	0.196	0.365	0.211	0.347	0.198	0.355	0.190	0.406
13	0.174	0.197	0.189	0.359	0.185	0.297	0.208	0.448	0.197	0.210
14	0.199	0.130	0.203	0.429	0.201	0.442	0.203	0.431	0.203	0.328
15	0.204	0.362	0.196	0.390	0.208	0.361	0.208	0.373	0.202	0.369
Gain (dB)	15.38		16.66		15.84		16.14		16.40	
Z(Ω)	49.62−j2.943		45.42−j5.74		50+j0		50.00−j0.001		50.09+j0.15	

computational burden. In order to test the consistency of the PSO algorithms best, mean, worst and standard deviation (SD) of results obtained in 10 simulation runs are presented in Table 5.

From Table 5, it is evident that the modified PSO and FDR-PSO algorithms are not consistent in yielding the optimal results. The standard deviation (SD) of impedance in modified PSO and FDR-PSO algorithms are very high compared to CLPSO. Therefore it is clear that the new learning algorithm significantly improved the consistency of the PSO algorithm.

5.3 Four-element Yagi–Uda antenna design

This is a four-element Yagi–Uda array optimised for high gain, low sidelobes, and required input impedance. Sidelobe means any lobe including the back lobe. The constants were set as $a = 30$, $b = 1$, $c = 1$ and $d = 2$. The PSO, GA and CI algorithm optimised results are presented in Table 6. Feval is set at 3500 for all PSO-based algorithms.

From Table 6, it is clear that results obtained using the modified PSO and the FDR-PSO algorithms are not good in terms of gain as compared to the results yielded by the GA [10] and the CI [11, 12] algorithms. As the input impedance matching is critically evaluated for better power transfer to the antenna, the overall performance of CLPSO is better as compared to that of the GA and CI algorithms. Even though the gain of the CLPSO optimised antenna is slightly lower than the optimised gain of the GA and CI optimised antennas, the 50 Ω impedance requirement is almost exactly satisfied by the CLPSO optimised antenna. Further, the computation time of the PSO algorithms is approximately twenty times less compared to the CI algorithm. The H-plane radiation patterns of the CLPSO and GA optimised four element Yagi–Uda antenna are shown in Fig. 3. It is seen from the results that sidelobe

Table 5: Performance of PSO algorithms

	PSO	FDR-PSO	CLPSO
Best (gain, dB)	15.8400	16.14	16.40
(Z, Ω)	50+j0	50.002−j0.0007	50.089+j0.15
Mean (gain, dB)	15.2740	15.81	16.244
(Z, Ω)	45.31+j 0.025	44.79−j0.0003	49.608−j0.099
Worst (gain, dB)	14.7200	15.620	16.130
(Z, Ω)	26.58+j0.125	23.98−j0.001	49.471−j0.1401
SD (gain, dB)	0.4013	0.4013	0.1478
(Z, Ω)	10.47+j0.055	10.47+j0.056	0.3082+j0.6176

Table 6: Results of gain, impedance and SLL optimised four element designs

Element	Equal [10]		GA [10]		CI [11, 12]		PSO		FDR-PSO		CLPSO	
	L	*S*	*L*	*S*	*L*	*S*	*L*	*S*	*L*	*S*	*L*	*S*
	(in terms of λ)		(in terms of λ)		(in terms of λ)		(in terms of λ)		(in terms of λ)		(in terms of λ)	
1	0.243	–	0.245	–	0.238	–	0.202	–	0.249	–	0.238	–
2	0.2295	0.150	0.236	0.283	0.237	0.288	0.227	0.238	0.230	0.228	0.233	0.311
3	0.2265	0.150	0.221	0.179	0.220	0.200	0.231	0.322	0.206	0.235	0.217	0.205
4	0.2265	0.150	0.212	0.279	0.212	0.265	0.240	0.280	0.192	0.327	0.206	0.279
Gain (dB)	9.63		9.73		9.83		8.48		8.57		9.44	
Z(Ω)	37.96+j12.04		40.88−j3.46		46.19+j8.12		49.877+j.0115		50+j.0007		49.56+j.11	
rSLL (dB)	−8.35		−14.33		−15.1		−15.00		−18.02		−15.02	

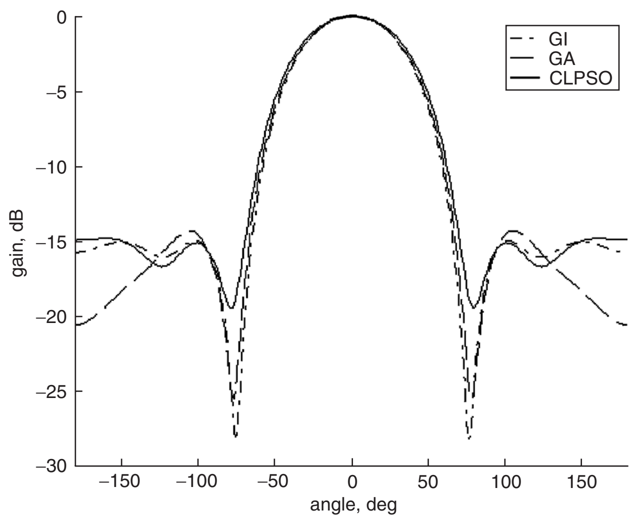

Fig. 3 *Radiation patterns of optimised four-element Yagi–Uda antenna*

levels are below 15 dB and the pattern is as expected in terms of beam width.

6 Conclusions

In this paper, PSO-based optimisation techniques have been developed for Yagi–Uda antenna design with various performance specifications. Four-, six-, and fifteen-element Yagi antenna design problems have been investigated. To alleviate the problem of premature convergence commonly encountered in the basic PSO, a novel self-learning strategy is applied. In comprehensive learning PSO (CLPSO), each particle learns from potentially all particles' pbests in the swarm. The results are compared to published results obtained by using genetic and computational intelligence optimised methods. The performance of the PSO with new learning strategy is better than the modified PSO and the FDR-PSO algorithms. For almost the same performance, the computation time required by the CLPSO is approximately 20 times less than the CI-based method. From the experimental results, we can conclude that the CLPSO algorithm is the best, in terms of overall solution quality and computation time, for solving this class of design problem. The results also show that PSO is a robust and useful optimisation technique for designing Yagi–Uda antennas according to the desired target specifications [Note 1].

7 Acknowledgment

The authors would like to thank the Department of Science and Technology, Government of India for the financial support under BOYSCAST Fellowship Scheme (SR/BY/E-08/2003) and Nanyang Technological University for providing necessary facilities to carry out this research. The authors also thank Poynting Innovations (Pty) Ltd, for providing the SuperNEC software for our simulations.

Note 1. The codes are available from http://www.ntu.edu.sg/home/EPNSugan

8 References

1 Stutzman, W.L., and Thiele, G.A.: 'Antenna theory and design' (Wiley, 1997, 2nd edn.)
2 Qian, Y.X., Deal, W.R., Kaneda, N., and Itoh, T.: 'Microstrip-fed quasi-Yagi antenna with broadband characteristics', *Electron. Lett.*, 1998, **34**, (23), pp. 2194–2196
3 Sor, J., Qian, Y.X., and Itoh, T.: 'Coplanar waveguide fed quasi-Yagi antenna', *Electron. Lett.*, 2000, **36**, (1), pp. 1–2
4 Grajek, P.R., Schoenlinner, B., and Rebeiz, G.M.: 'A 24-GHz high-gain Yagi–Uda antenna array', *IEEE Trans. Antennas Propag.*, 2004, **52**, (5), pp. 1257–1261
5 Ehrenspeck, H.W., and Poehler, H.: 'A new method for obtaining maximum gain from Yagi antennas', *IRE Trans. Antennas Propag.*, 1959, **7**, pp. 379–386
6 Cheng, D.K., and Cheng, C.A.: 'Optimum element spacing for Yagi–Uda arrays', *IEEE Trans. Antennas Propag.*, 1973, **21**, pp. 615–623
7 Cheng, D.K., and Cheng, C.A.: 'Optimum element length for Yagi–Uda arrays', *IEEE Trans. Antennas Propag.*, 1975, **23**, pp. 8–15
8 Cheng, D.K.: 'Gain optimization for Yagi–Uda arrays', *IEEE Antennas Propag. Mag.*, 1991, **33**, pp. 42–45
9 Altshuler, E.E., and Linden, D.S.: 'Wire-antenna designs using genetic algorithms', *IEEE Antennas Propag. Mag.*, 1997, **39**, (2), pp. 33–43
10 Jones, E.A., and Joines, W.T.: 'Design of Yagi–Uda antennas using genetic algorithms', *IEEE Trans. Antennas Propag.*, 1997, **45**, (9), pp. 1386–1392
11 Venkatarayalu, N.V., and Ray, T.: 'Optimum design of Yagi–Uda antennas using computational intelligence', *IEEE Trans. Antennas Propag.*, 2004, **52**, (7), pp. 1811–1818
12 Venkatarayalu, N.V., and Ray, T.: 'Single and multi-objective design of Yagi–Uda antennas using computational intelligence'. Proc. Cong. Evolutionary Computation, Canberra, Australia, 2003, pp. 1237–1242
13 Eberhart, R.C., and Kennedy, J.: 'A new optimizer using particle swarm theory'. Proc. 6th Int. Symp. Micro machine and Human Science, Nagoya, Japan, 1995, pp. 39–43
14 Shi, Y., and Eberhart, R.C.: 'A modified particle swarm optimizer'. Proc. Cong. Evolutionary Computation, Anchorage, USA, 1998, pp. 69–73
15 Clerc, M.: 'The swarm and the queen: towards a deterministic and adaptive particle swarm optimization'. Proc. Cong. Evolutionary Computation, Washington DC, USA, 1999, pp. 1951–1957
16 Gies, D., and Rahmat-Samii, Y.: 'Particle swarm optimization for reconfigurable phase differentiated array design', *Microw. Opt. Technol. Lett.*, 2003, **38**, (3), pp. 168–175
17 Robinson, J., and Rahmat-Samii, Y.: 'Particle swarm optimization in electromagnetics', *IEEE Trans. Antennas Propag.*, 2004, **52**, (2), pp. 397–407
18 Boeringer, D.W., and Werner, D.H.: 'Particle swarm optimization versus genetic algorithms for a phased array synthesis', *IEEE Trans. Antennas Propag.*, 2004, **52**, (3), pp. 771–779
19 Suganthan, P.N.: 'Particle swarm optimizer with neighbourhood operator'. Proc. Cong. Evolutionary Computation, Washington DC, 1999, pp. 1958–1962
20 Peram, T.K., Veeramachaneni, K., and Mohan, C.K.: 'Fitness-distance-ratio based particle swarm optimization'. Proc. IEEE Swarm Intelligence Symp., Indiana, USA, 2003, pp. 174–181
21 Ratnaweera, S., Halgamuge, S.K., and Watson, H.: 'Self-organizing hierarchical particle swarm optimizer with time varying accelerating coefficients', *IEEE Trans. Evol. Comput.*, 2004, **8**, (3), pp. 240–255
22 Krink, T., and Lovbjerg, M.: 'The life cycle model: combining particle swarm optimization, genetic algorithms and hill climbers'. Proc. Parallel Problem Solving from Nature VII, Granada, Spain, 2002, pp. 621–630
23 Liang, J.J, Qin, A.K., Suganthan, P.N., and Baskar, S.: 'Evaluation of comprehensive learning particle swarm optimizer', *Lect. Notes Comput. Sci.*, 2004, **3316**, pp. 230–235
24 Poynting-SuperNEC Downloads, http://www.poynting.co.za/software/supernec.shtml