Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 686730, 7 pages
http://dx.doi.org/10.1155/2014/686730

Research Article

Application of Hybrid Optimization Algorithm in the Synthesis of Linear Antenna Array

Ezgi Deniz Ülker,[1] Ali Haydar,[1] and Kamil Dimililer[2]

[1] *Department of Computer Engineering, Girne American University, Mersin 10, Turkey*
[2] *Department of Electrical and Electronics Engineering, Girne American University, Mersin 10, Turkey*

Correspondence should be addressed to Ezgi Deniz Ülker; ezgideniz@gau.edu.tr

Received 21 June 2014; Accepted 7 September 2014; Published 25 September 2014

Academic Editor: Ebrahim Momoniat

Copyright © 2014 Ezgi Deniz Ülker et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The use of hybrid algorithms for solving real-world optimization problems has become popular since their solution quality can be made better than the algorithms that form them by combining their desirable features. The newly proposed hybrid method which is called Hybrid Differential, Particle, and Harmony (HDPH) algorithm is different from the other hybrid forms since it uses all features of merged algorithms in order to perform efficiently for a wide variety of problems. In the proposed algorithm the control parameters are randomized which makes its implementation easy and provides a fast response. This paper describes the application of HDPH algorithm to linear antenna array synthesis. The results obtained with the HDPH algorithm are compared with three merged optimization techniques that are used in HDPH. The comparison shows that the performance of the proposed algorithm is comparatively better in both solution quality and robustness. The proposed hybrid algorithm HDPH can be an efficient candidate for real-time optimization problems since it yields reliable performance at all times when it gets executed.

1. Introduction

In recent years, many metaheuristic algorithms have been proposed to solve problems in various fields [1–10]. Some well known and preferred metaheuristic algorithms are Clonal Selection Algorithm (CSA), Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colony (ABC) algorithm, and Harmony Search (HS) Algorithm.

These algorithms have different operators for achieving exploration and exploitation attributes to find the global optimum point of a given problem. Hence, their performances depend on the selection and adjustment of control parameters. In many real-world optimization problems the success depends on the selection of appropriate optimization algorithm with well-tuned control parameters [11–13]. This is because the characteristics of the problem at hand may need more exploration than exploitation or vice versa in order to find a solution with high quality. Usually each optimization algorithm uses different properties to achieve the exploration and exploitation goals. In general the algorithm may yield a better performance for a specific problem, but when this algorithm is applied to other problems its performance can be considerably worse. In order to overcome these difficulties many approaches were proposed in literature often based on the combination of several techniques. These are called hybrid models and are quite efficient in optimization [14–17]. The optimization algorithms use control parameters to have a balance between exploration and exploitation attributes and the performance of an algorithm mainly depends on selection of control parameters properly. The optimization algorithms are aimed to be robust or in another way they should be insensitive to the control parameter variations [18–20]. Solution quality is also an important aspect for solving optimization problems. However, the solution quality of the algorithm is not directly related to its robustness [21]. An algorithm that is highly robust may have a low solution quality or the algorithm that has high solution quality may be sensitive to the control parameter variations [21]. In order to have both high solution quality and robustness features, a hybrid algorithm was proposed recently [22]. The new hybrid method is called HDPH which is Hybrid Differential,

Particle, and Harmony algorithm. The HDPH algorithm is based on the combination of well known algorithms such as DE and PSO algorithms, together with the HS algorithm.

In this paper, the benefits of HDPH algorithm on a real-world application, namely, the array antenna synthesis problem, both for solution quality and for robustness are shown. This problem can be thought as a nonlinear multidimensional optimization problem which is very difficult to optimize using traditional methods. Hence, in the literature, evolutionary algorithms and their variants have been extensively used for solving antenna array problems [23–30]. The algorithms are usually modified for obtaining better results for a specific antenna problem.

In this study, the main aim is to show the performance of HDPH algorithm on the given problem without making any modification on it. In order to ensure our observations found in [22] about the performance of HDPH algorithm on the robustness and the solution quality of the results, the performances of the HDPH algorithm and the algorithms that form it are compared on array antenna synthesis problem. The experimental results show that HDPH algorithm performed better than the selected algorithms DE, PSO, and HS in terms of robustness and solution quality.

The rest of the paper is organized as follows. In Section 2, the theory of the proposed hybrid algorithm HDPH is explained. Section 3 presents the selected application which is an array antenna synthesis used for comparison and performance evaluation for HDPH and the selected algorithms. Lastly, in Section 4 concluding remarks of the paper are given.

2. HDPH Algorithm

In the literature, many optimization algorithms are proposed to generate new hybrid methods to improve the solution quality [14–17]. In order to generate a hybrid algorithm, different features of the several algorithms are combined. In some hybrid methods the main features of the algorithms are kept. However, in some hybrid methods, the main characteristics of the algorithms are improved and then merged. Some well known and powerful algorithms such as DE, PSO, and HS are preferred by many researchers in various fields [1, 2, 4]. Each algorithm has different characteristics in terms of robustness and solution quality. Performances of the algorithms DE and PSO are effective in solution quality. On the other hand, HS algorithm is a robust algorithm on different applications [22]. Therefore, a new hybrid method called HDPH is generated to generate a powerful algorithm in terms of high solution quality and robustness.

In this section, the steps of the novel hybrid algorithm, HDPH, which was recently proposed [22] and is composed of DE, PSO, and HS algorithms, are briefly explained. The algorithm HDPH does not change the properties of the selected algorithms. As it is observed from the previous research that the performances of selected algorithms depend on the selection of control parameter and fine tuning of control parameters is needed [11–13]. However, in order to avoid complexity of selecting control parameters for DE, PSO, and HS algorithms these parameters are chosen randomly from the ranges that are used for the selection of control parameters of the algorithms [22]. HDPH uses the main operators of the selected algorithms without changing their characteristics. The operators of the merged algorithms are applied on candidate solutions in a repeated sequence. Population of solutions obtained after the application of operators to one algorithm is used as a new population to the next algorithm. Figure 1 shows the flowchart of the algorithm. The steps of the HDPH algorithm are given as follows.

Step 1. Initialize the candidate population of solutions X_i where $i = \{1, 2, 3 \ldots, NP\}$ within given ranges and NP is the size of the population.

Step 2. Apply crossover, mutation, and selection operators of DE algorithm by using the following expressions:

$$Z_i = X_a + F\left(X_b - X_c\right), \tag{1}$$

$$U_{ij} = \begin{cases} Z_{ij}, & \text{if } r_j \le CR, \\ X_{ij}, & \text{otherwise}, \end{cases} \tag{2}$$

$$X_i = \begin{cases} U_i, & \text{if } f\left(U_i\right) < f\left(X_i\right), \\ X_i, & \text{otherwise}. \end{cases} \tag{3}$$

The mutant vector Z_i is calculated by using (1) for each member in the candidate population with X_a, X_b, and X_c as distinct members in this population. In the crossover operator Z_{ij} is crossed with X_{ij} to generate U_{ij} by using (2), where Z_{ij} and X_{ij} are the jth elements of the ith mutant vector Z_i and ith solution vector X_i, respectively, and r_j is uniformly distributed number for each jth element of Z_i. The main control parameters of DE are F and CR. They are used for mutation and crossover operations. In the selection process, new candidates for X_i are determined as either the vector U_i or its previous solution depending on the fitness values of U_i and X_i by using (3).

Step 3. Apply operators of PSO algorithm by using the following expressions:

$$V_i = wV_i + c_1\left(P_{\text{best},i} - X_i\right) + c_2\left(\text{global}_{\text{best}} - X_i\right), \tag{4}$$

$$X_i = X_i + V_i. \tag{5}$$

w, c_1, and c_2 are main control parameters applied on velocities V_i by using (4). Particles update their positions using (5). Value $\text{global}_{\text{best}}$ is the best known position reached up to that point and $P_{\text{best},i}$ is the best position reached by the ith particle in the swarm.

Step 4. Perform operators of HS algorithm by applying control parameters *hmcr*, *par*, and fw to update positions of candidates in search space.

With probability of *hmcr*, the element of candidate solution is selected from the population. With probability of 1-*hmcr*, the element of candidate solution is generated randomly. HS can have nonupdated candidate elements in the population with probability of 1-*par*. The jth element of

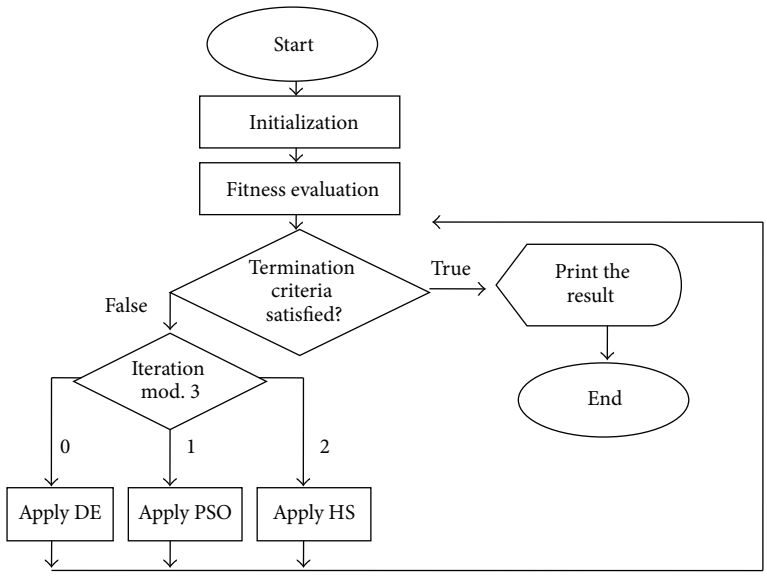


Figure 1: Basic processes of HDPH algorithm in form of flowchart.

the candidate i, X_{ij} can be updated with probability of *par* by applying the following expression:

$$X_{ij} = X_{ij} + \text{rand}(\,)\, fw, \tag{6}$$

where rand( ) is a random number in the range $\in (-1, 1)$.

Step 5. Repeat Steps 2, 3, and 4 until the chosen stopping criterion is met.

The algorithm is performed in a loop until the termination criterion is satisfied. Elitism is included in HDPH by keeping the best solution at the end of each iteration.

3. Problem Definition and Experimental Results

Synthesis of linear arrays is one of the application problems that are used for optimization. We can consider the array of $2N$ isotropic antennas which are placed at the same distance systematically in x-axis as shown in Figure 2.

Array factor (AF) functions which are used to constitute objective functions in the synthesizing of linear antenna arrays are nonlinear, nonseparable, and multimodal. Complexity of the problem increases with the number of array elements. Therefore, metaheuristic algorithms have become popular in synthesizing linear arrays recently [23–30]. The array factor of the linear antenna array (with $2N$ antenna elements) is described by the following expression:

$$\text{AF}\left(\vec{x}, \vec{I}, \vec{\phi}, \theta\right) = \sum_{i=-N}^{N} I_i \exp\left(2\pi \frac{x_i}{\lambda} \sin(\theta) + \phi_i\right). \tag{7}$$

Antenna elements are positioned on a line symmetrically with respect to the origin where λ is the wavelength and three

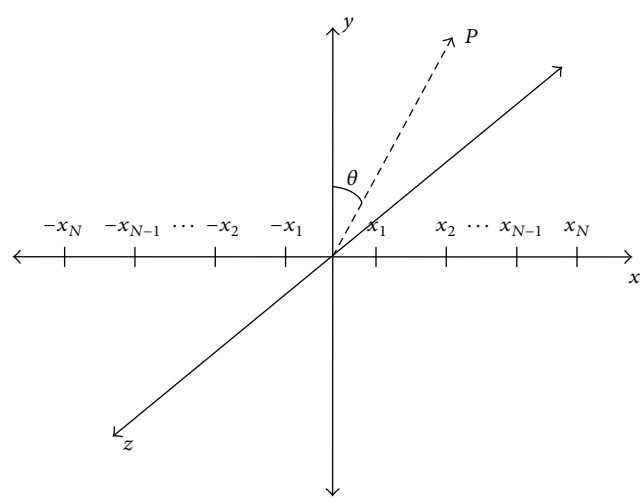


Figure 2: Geometry of the $2N$ elements symmetric linear antenna array elements placed along the x-axis.

vectors $\vec{x}$, $\vec{I}$, and $\vec{\phi}$ represent positions, excitation currents, and excitation phases of the antenna elements, respectively [26]. Angle θ represents the angular separation from the perpendicular plane (y-z plane in Figure 2). If vectors $\vec{x}$, $\vec{I}$, and $\vec{\phi}$ are fixed, then the array factor is a function of angle θ only at a given frequency.

We considered the antenna elements as placed in a symmetric manner with respect to origin and the excitation current amplitudes are all uniform and have unity value. Phase of each element is subject to change so that element phases are the same at symmetric positions.

The peak sidelobe level (PSLL) of the antenna array is defined by the following expression:

$$\mathrm{PSLL}\left(\vec{x},\vec{I},\vec{\phi}\right)=\max_{\forall\theta\in S}\left\{\frac{\mathrm{AF}\left(\vec{x},\vec{I},\vec{\phi},\theta\right)}{\mathrm{AF}\left(\vec{x},\vec{I},\vec{\phi},0\right)}\right\}, \tag{8}$$

where S is the space spanned by the angle excluding the main lobe with the center at $\theta = 0$. In this problem, the objective of optimization is to minimize the peak sidelobe level (PSLL) value in order to find the antenna configuration which satisfies the maximum suppression of the sidelobes. In our case, the aim is to find the best positions ($\vec{x}$) and phases ($\vec{\phi}$) that minimize PSLL.

In this study the performance of the hybrid algorithm HDPH is compared with three algorithms DE, PSO, and HS in synthesizing a well-known linear antenna array problem for both the solution quality and robustness with random initialization of the population. For the selected algorithms the control parameters are chosen from the sets similarly used in literature [7, 8, 10, 21]. Population size is set to 100 and 150000 objective function calculations are performed which was the stopping criterion for all algorithms. Each combination of control parameters is used in the synthesis of linear array antenna problem. Each algorithm is run for 30 times for different combination of control parameters and only the parameter combination that yielded the best solution is taken into consideration. However, for the proposed HDPH algorithm the control parameters are randomized in the ranges used for other three algorithms. The results are obtained only by running it 30 times. The numerical results of only position phase synthesizing of 32-element linear array are presented here. The angular resolution is set at 0.2° which was suggested by Lin et al. [26], and desired first null beamwidth is set to 6.3°. The minimum and the maximum distances between any adjacent elements are set to 0.5λ and λ, respectively.

The best values, standard deviation, and average PSLL values that are obtained for the linear antenna synthesis problem are tabulated in Table 1. These values are obtained by running the programs of DE, PSO, and HS algorithms and the hybridization of them, and HDPH algorithm, which are all written in C++ language. As it can be seen from Table 1, HDPH found smaller values for PSLL compared with DE, PSO, and HS. Also, when standard deviation values are considered, again HDPH had smaller values compared to the other three algorithms. These results indicate that the performance of the HDPH algorithm both for solution quality and robustness to random initialization is much better than the algorithms that form it for this specific example.

In Table 2, the best solution sets which are obtained for position-phase synthesizing of 32-element array by HDPH, DE, PSO, and HS algorithms are tabulated. From these results, it is observed that the best antenna element positions found by different methods are quite similar, but the phases are found different than each other. This indicates that the objective function has many local optimum points that differ by their phases in the solution space.

The synthesized array patterns that were obtained for the best solution sets of HDPH, DE, PSO, and HS algorithms are shown in Figure 3. From this figure, it is observed that the beamwidths and sidelobe levels are different for each algorithm. This is because of each algorithm having a slightly different optimum value. Our observations show that the main differences between obtained results by different algorithms happen either in the first three sidelobes or at the back sidelobe. Therefore, we zoomed to these two regions in Figures 4 and 5.

Table 1: The best, average, and standard deviation values of PSLL on HDPH, DE, PSO, and HS algorithms.

Value	HDPH	DE	PSO	HS
Best	−23.6288	−23.0274	−21.914	−23.2408
Average	−23.31	−22.58	−18.02	−22.41
Standard deviation	0.0824	0.3142	0.948	0.275

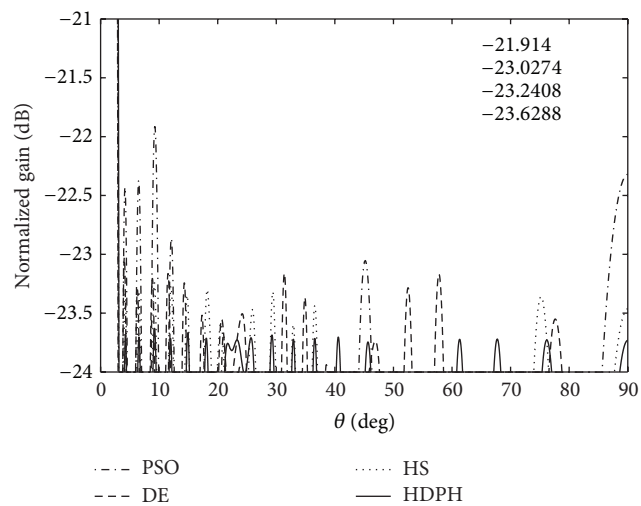


Figure 3: The best array patterns of position-phase synthesis of 32-element linear array obtained by using DE, PSO, HS, and HDPH algorithms.

In Figure 4, it can be clearly seen that sidelobe values are suppressed more for HDPH. DE and HS produced similar performances in terms of giving values that produce suppressed sidelobes. PSO produces values which are comparatively worse than the others in terms of resulting in higher sidelobes. So in short, the effect of hybridization on this problem is clearly demonstrated by the suppression of the sidelobes which is much higher for HDPH than for the algorithms DE, PSO and HS. In Figure 5, it is observed that back sidelobe produced by PSO is the highest one among the others and DE yields the lowest back sidelobe. HS and HDPH yield almost the same back sidelobe levels. However, in general analysis of Figure 3 it is clear that suppression of the configuration obtained by the HDPH is the highest among the others.

4. Conclusion

The proposed hybrid algorithm HDPH is applied to a linear antenna array synthesis problem. The results show that this algorithm outperforms the combined algorithms in terms of solution quality and robustness and yields a very good

Table 2: The best solution sets obtained with the HDPH and the other algorithms.

i	Phase-position 32 elements							
	HDPH		DE		PSO		HS	
	Position χ_i/λ	Phase (radian)	Position χ_i/λ	Phase (radian)	Position χ_i/λ	Phase (radian)	Position χ_i/λ	Phase (radian)
1	0.2502	0.6802	0.2569	0.0166	0.2566	6.0927	0.25	1.4355
2	0.7502	0.6632	0.7821	6.2599	0.7644	6.1197	0.75	1.54
3	1.2503	0.6373	1.2912	6.28	1.2795	5.993	1.25	1.44
4	1.7504	0.6541	1.8587	0.0127	1.7829	6.0911	1.75	1.4694
5	2.2511	0.6161	2.3553	0	2.2936	6.0053	2.25	1.48
6	2.7616	0.6303	2.9331	6.28	2.8242	5.9135	2.82	1.3673
7	3.3125	0.6909	3.4884	6.28	3.375	6.2422	3.3712	1.5101
8	3.8735	0.6495	4.1007	6.28	3.9542	6.1168	3.9512	1.5157
9	4.5241	0.6704	4.7589	0.0008	4.6	6.0234	4.5412	1.5133
10	5.0909	0.6784	5.3332	6.28	5.1457	6.095	5.0919	1.5055
11	5.7519	0.6229	6.0569	6.28	5.8165	6.0303	5.8081	1.3939
12	6.5689	0.7794	6.9017	0.1496	6.6542	6.28	6.6281	1.2872
13	7.4283	0.6691	7.8284	6.2618	7.5208	6.135	7.5183	1.5038
14	8.2934	0.1645	8.7129	5.7984	8.398	5.5635	8.3283	1.9569
15	9.2419	1.2184	9.6456	0.6109	9.2947	6.2552	9.3283	0.9861
16	10.1211	0.678	10.5306	0.0526	10.1481	6.0318	10.1583	1.5572

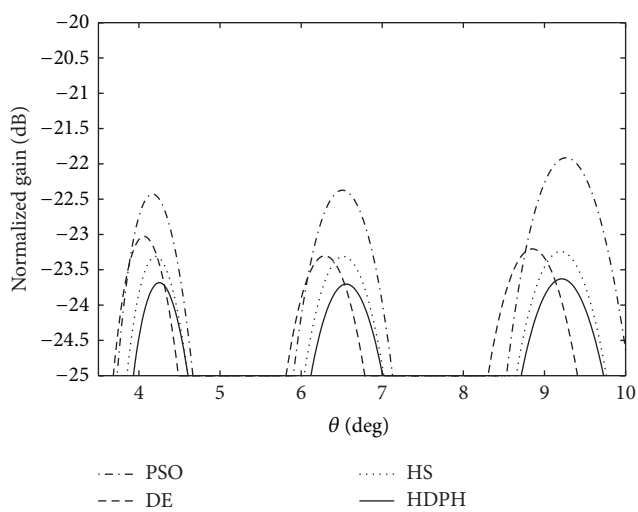


Figure 4: The first three sidelobes of position-phase synthesis of 32-element linear array obtained by using DE, PSO, HS, and HDPH algorithms.

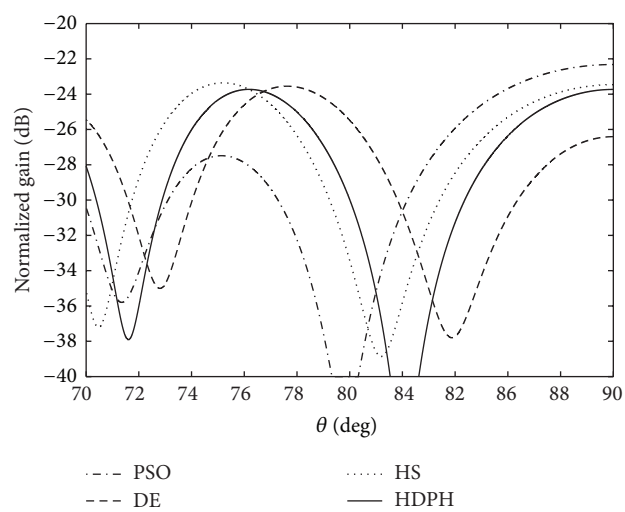


Figure 5: The back sidelobe of position-phase synthesis of 32-element linear array obtained by using DE, PSO, HS, and HDPH algorithms.

performance at each run. The values obtained by HDPH clearly produce more suppressed sidelobes compared to the other three algorithms. Considering the performance of HDPH in optimizing antenna arrays, we see that it can be a good candidate for these kinds of problems. The parameter selection is performed to select the best control parameters for DE, PSO, and HS algorithms for the linear antenna array synthesis problem and these control parameters are used to run them for 30 times to find the statistical results. However, the proposed HDPH algorithm further has advantage because there is no need for control parameter selection. Only, it run for 30 times with randomized control parameters to obtain the statistical results.

We also recommend the HDPH algorithm for real-time application problems in the future. In real-time applications speed and convergence play an important role since time is limited for adjusting the control parameters. An algorithm that provides high solution quality in each run can be an efficient method for real-time applications. Also, the robustness of the proposed algorithm HDPH is an indication

that it can be used with an acceptable performance in these types of real-time application problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] H.-K. Kim, J.-K. Chong, K.-Y. Park, and D. A. Lowther, "Differential evolution strategy for constrained global optimization and application to practical engineering problems," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1565–1568, 2007.

[2] R. J. Kuo, S. Y. Hong, and Y. C. Huang, "Integration of particle swarm optimization-based fuzzy neural network and artificial neural network for supplier selection," *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 3976–3990, 2010.

[3] E. Deniz and S. Ülker, "Clonal selection algorithm application to simple microwave matching network," *Microwave and Optical Technology Letters*, vol. 53, no. 5, pp. 991–993, 2011.

[4] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "Harmony search optimization: application to pipe network design," *International Journal of Modelling & Simulation*, vol. 22, no. 2, pp. 125–133, 2002.

[5] Q.-K. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

[6] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.

[7] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE Press, December 1995.

[9] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[10] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[11] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 84–88, San Diego, Calif, USA, July 2000.

[12] A. S. D. Dymond, A. P. Engelbrecht, and P. S. Heyns, "The sensitivity of single objective optimization algorithm control parameter values under different computational constraints," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 1412–1419, New Orleans, LA, June 2011.

[13] S. Smit and A. Eiben, "Comparing parameter tuning methods for evolutionary algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computing*, pp. 399–406, IEEE, Trondheim, Norway, May 2009.

[14] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, "Particle swarm optimization: hybridization perspectives and experimental illustrations," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5208–5226, 2011.

[15] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters*, vol. 93, no. 5, pp. 255–261, 2005.

[16] N. Holden and A. A. Freitas, "A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 100–107, IEEE, June 2005.

[17] H.-Q. Li and L. Li, "A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems," in *Proceedings of the International Conference on Intelligent Pervasive Computing (IPC '07)*, pp. 94–97, October 2007.

[18] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 405–420, 2006.

[19] A. Ahrari, M. R. Saadatmand, M. Shariat-Panahi, and A. A. Atai, "On the limitations of classical benchmark functions for evaluating robustness of evolutionary algorithms," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3222–3229, 2010.

[20] J. Branke, "Creating robust solutions by means of an evolutionary algorithm," in *Parallel Problem Solving from Nature-PPSN V*, vol. 8, pp. 119–128, 1998.

[21] E. D. Ülker and A. Haydar, "Comparing the robustness of evolutionary algorithms on the basis of benchmark functions," *Advances in Electrical and Computer Engineering*, vol. 13, no. 2, pp. 59–64, 2013.

[22] E. D. Ülker and A. Haydar, "A hybrid algorithm based on differential evolution, particle swarm optimization and harmony search algorithms," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '13)*, pp. 417–420, September 2013.

[23] S. Yang, Y. B. Gan, and A. Qing, "Sideband suppression in time-modulated linear arrays by the differential evolution algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 1, pp. 173–175, 2002.

[24] T. B. Chen, Y. L. Dong, Y. C. Jiao, and F. S. Zhang, "Synthesis of circular antenna array using crossed particle swarm optimization algorithm," *Journal of Electromagnetic Waves and Applications*, vol. 20, no. 13, pp. 1785–1795, 2006.

[25] K. Guney and M. Onay, "Optimal synthesis of linear antenna arrays using a harmony search algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15455–15462, 2011.

[26] C. Lin, A. Qing, and Q. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 8, pp. 2553–2561, 2010.

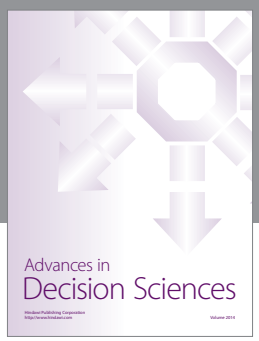
[27] D. G. Kurup, M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 9, pp. 2210–2217, 2003.

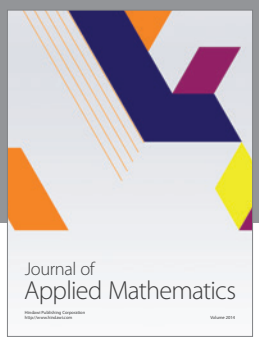
[28] Y. Chen, S. Yang, and Z. Nie, "The application of a modified differential evolution strategy to some array pattern synthesis problems," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 7, pp. 1919–1927, 2008.

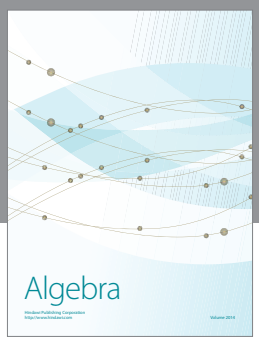
[29] S. K. Goudos, V. Moysiadou, T. Samaras, K. Siakavara, and J. N. Sahalos, "Application of a comprehensive learning particle

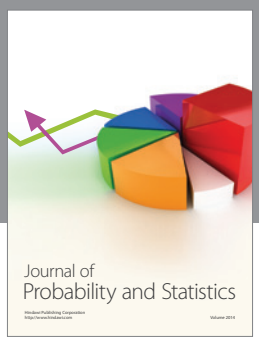
swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 125–129, 2010.

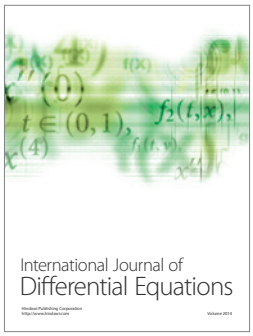
[30] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, pp. 2674–2679, 2005.











Hindawi

Submit your manuscripts at
http://www.hindawi.com