A Novel Segmentation Approach in GA and its Application in Antenna Array

Bhargav Appasani and Nisha Gupta

Abstract — **Initial population has a significant effect on the optimum solution in genetic algorithms (GA). Traditionally, initial populations are generated using pseudo random numbers, and the initial population is then applied to reach to the optimum solution. In this paper, a pragmatic strategy employing a segmentation technique is proposed in order to explore the solution space in an efficient manner and to achieve the optimum solution. The proposed technique is applied over several benchmark test functions and its superiority is demonstrated both in terms of convergence speed and in improving the final solution. The flexibility of the technique is that it can be applied in conjunction with other variants of GA and also with optimization techniques. Finally, the technique is applied to the practical engineering problem of side lobe level (SLL) reduction in antenna arrays and its efficacy over the conventional GA is demonstrated.**

Keywords — **Segmentation, Genetic Algorithm, Global Optimization technique, Antenna array.**

I. INTRODUCTION

A simulation model of a real life problem is often complex, and the objective function to be minimized may be non-convex and have several local minima. Therefore, the global optimization methods are needed which are capable of exploring the entire solution space without stagnating into a local minimum. In the recent years, there has been a great deal of interest in developing methods for solving global optimization problems [1-3]. One of the most popular global optimization techniques is GA [4] which is meta-heuristics [5] used for solving problems with both discrete and continuous variables. Initial populations may have a significant effect on the optimum solution over several generations [6]. The genetic operations like crossover and mutation are just instruments for manipulating the population so that it evolves towards the final population including a “close to optimum” solution. Numerous strategies [7] have been developed to overcome the initial population problems and to improve the convergence speed of the GA such as the orthogonal genetic algorithm approach [8], the micro-GA [9], multi-population GA [10], nonlinear transformation of GA operators [11], a dynamic penalty function equipped GA [12], hybrid GA [13], use of differential operators [14], etc. However, these approaches are still plagued with slow convergence and poor competency to attain the global optimum [15].

Article history: Received May 30, 2015; Accepted November 15, 2017

B. Appasani is with the School of Electronics Engg., KIIT University, Bhubaneswar, INDIA-751024, E-mail: appybarkas@gmail.com

N. Gupta is with the Dept. Electronics & Communication Engg., Birla Institute of Technology, Mesra, INDIA-835215, E-mail: ngupta@bitmesra.ac.in

In this paper, a simple, pragmatic technique called “segmentation” is employed to improve the convergence speed and to obtain the optimum solution. In order to demonstrate the efficacy of the proposed technique, it is tested over the several benchmark test functions and results are compared. The proposed technique is flexible and can be implemented in conjunction with other variants of GA mentioned above [7-15] and also with other optimization techniques.

The organization of the paper is as follows: the subsequent section presents the proposed segmentation technique in GA. The technique is tested on 14 benchmark functions and the results are presented. The third section presents investigates the impact of a number of segments on the convergence speed and on the quality of solution. In the fourth section the proposed technique is applied to the practical engineering optimization problem of side lobe reduction in antenna arrays and in the last section conclusions and the future scope of research is presented.

II. SEGMENTATION TECHNIQUE

In this section a pragmatic approach to improve the convergence speed in GA is presented. This technique may be called as segmentation technique as it involves segmentation of the chromosomes into two parts in conjunction with the segmentation of the generations into several segments and applying the GA to each of these segments. This approach is depicted in Figure 1.

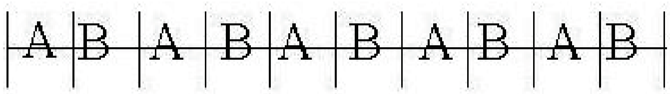

Fig. 1. Proposed segmentation technique

The entire solution set (chromosomes) is divided into two sets of chromosomes: set ‘A’ and set ‘B’ and the entire range of generations is divided into several intervals. Chromosomes in set ‘A’ and ‘B’ are optimized alternatively in the successive intervals. In the present case it is taken as ten. In the first segment the chromosomes in a set ‘A’ are optimized using GA keeping chromosomes in set ‘B’ constant. In the second segment chromosomes in set ‘B’ are optimized using GA, keeping chromosomes in the set ‘A’ constant (their value being equal to the values obtained after the first segment) and the procedure is then repeated for all the segments. The flow chart for this technique is shown in Fig.2. The Segmentation Technique is then applied to 14 benchmark test functions [16-

17] ad its performance is evaluated. These benchmark functions are usually the best possible means to test the effectiveness of any new technique. This can also be used for comparison of optimization techniques. Some functions have multiple local minima and are called multimodal. Some are non-separable, i.e. they cannot be expressed as the sum of the functions of individual variables. Some are regular, i.e. they are differentiable at each point in their domain. The nature of the functions [18-19] is given in Table I. Each of the functions is assumed to have 20 independent variables. The fitness value is calculated for 100 numbers of simulations and the average is calculated for each test function as shown in Table II.

TABLE I [16]
BENCHMARK FUNCTIONS

	Multimodal?	Separable?	Regular?	Range of each dimension
Ackley	Yes	no	yes	[-30,30]
Fletcher-Powell	Yes	no	no	[-π,π]
Griewank	Yes	no	yes	[-600,600]
Penalty #1	Yes	no	yes	[-50,50]
Penalty #2	Yes	no	yes	[-50,50]
Quartic	No	yes	yes	[-1.28,1.28]
Rastrigin	Yes	yes	yes	[-5.12,5.12]
Rosenbrock	No	no	yes	[-2.048,2.048]
Schwefel 1.2	No	no	yes	[-65.536,65.536]
Schwefel 2.21	No	no	no	[-100,100]
Schwefel 2.22	Yes	no	no	[-10,10]
Schwefel 2.26	Yes	yes	no	[-512,512]
Sphere	No	yes	yes	[-5.12,5.12]
Step	No	yes	yes	[-200,200]

TABLE II
AVERAGE FITNESS VALUE FOR 100 SIMULATIONS

	Segmentation (Ten segments)	Segmentation (Five segments)	Segmentation (Two segments)	Without Segmentation
Ackley	0.00081551	0.0105	0.3002	0.6305
Fletcher-Powell	2.81E+06	2.75E+06	2.74E+06	1.25E+06
Griewank	1.5802E-06	2.2563E-08	2.6785E-07	0.00092091
Penalty #1	2.76E-06	3.25E-07	3.80E-03	1.81E-02
Penalty #2	0.0024	0.0045	0.0065	0.0045
Quartic	2.16E-09	2.19E-10	2.87E-10	8.67E-05
Rastrigin	5.2787	6.3495	8.0691	9.9832
Rosenbrock	1.00E+01	1.39E+01	1.81E+01	1.12E+01
Schwefel 1.2	9.7387	16.4931	24.9346	0.5267
Schwefel 2.21	6.83E-02	1.01E-01	1.57E-01	1.27E+00
Schwefel 2.22	0.0687	0.6126	0.1254	1.2071
Schwefel 2.26	-7.89E+01	-7.89E+01	-7.89E+01	-7.89E+01
Sphere	9.6825E-07	1.0844E-07	1.8663E-06	0.01
Step	0.00E+00	0.00E+00	4.50E-01	1.59E+00

TABLE III
BEST FITNESS VALUE IN 100 NOS. OF SIMULATIONS

	Segmentation (Ten segments)	Segmentation (Five segments)	Segmentation (Two segments)	Without Segmentation
Ackley	0.000054938	0.000059686	0.000064454	0.0102
Fletcher-Powell	8.64E+05	1.06E+06	1.16E+06	6.17E+05
Griewank	8.1002E-10	2.3025E-10	2.6824E-10	9.5098E-06
Penalty #1	6.38E-10	4.03E-11	2.21E-10	4.02E-06
Penalty #2	4.8884E-10	4.7749E-10	3.092E-10	5.4457E-07
Quartic	3.72E-16	1.67E-17	1.90E-16	4.48E-08
Rastrigin	0.001	1.9899	2.9849	2.0181
Rosenbrock	1.22E-02	1.87E-02	5.02E-02	3.40E-03
Schwefel 1.2	1.2182	6.8636	7.3334	0.051
Schwefel 2.21	3.05E-04	2.59E-04	1.50E-03	3.62E-01
Schwefel 2.22	0.00033606	0.3761	0.00027478	0.1651
Schwefel 2.26	-7.89E+01	-7.89E+01	-7.89E+01	-7.89E+01
Sphere	8.3075E-09	2.6785E-09	6.0637E-09	0.00015035
Step	0.00E+00	0.00E+00	0.00E+00	0.00E+00

The best value for 100 numbers of simulations for each test function is shown in Table III. MATLAB® is used for this purpose. In the proposed work an initial population size of 20 and the dimension of each function as 20 are considered. The upper and lower bounds for each function are given in Table I. The initial population is generated using random generators and are all real numbers. The maximum number of generations is taken as 1000 and the function tolerance is set to '0'. The mutation rate is set to 0.35 and the mutation function is taken as Gaussian function. Elite count is taken as '2'. A two point cross over function is used with crossover fraction '0.8'. The same parameters are also used for the segmentation technique in GA. The following observations are evident from the results obtained using proposed technique. The results are tabulated in Tables II and III.

I. For most of the benchmark functions the segmented GA performs better than the conventional GA without segmentation both in terms of average fitness and best fitness values. The exceptions are Fletcher-Powell and Schwefel 2.26 and the reason is explained later.

II. The average fitness values indicate the rate of convergence and the value of best fitness indicates the optimum solution achieved. To have a greater understanding about the convergence of the GA with and without segmentation, standard deviation values are calculated for each of the fitness functions for hundred simulations. Box plots are drawn for Ackley, Griewank and Penalty#1 functions, which are shown in Fig. 3.

III. Computation time depends mainly upon the cost function evaluation and hence the computational time using the segmentation technique is almost same as the computational time required for the conventional GA without segmentation. However, the segmented GA achieves an improved final solution for the same number of cost function evaluations.

No special effort was made to tune any of the optimization algorithms. Tuning of the algorithms may result in significant change in the overall performance. The real world problems may not be directly related to the benchmark functions used. Also, we may arrive at different conclusions if the problem setup changes (for example, varying the maximum generation limit, population size etc.). In spite of all these, the benchmark results show a new promising technique for optimization Similar parameters are considered while comparing the GA using segmentation technique and GA without any segmentation. If tuning of parameters is used with the conventional GA, then the same is to be applied to the GA using segmentation. In the present work, the performance of conventional GA is compared with GA using segmentation and no emphasis is made to compare it with other evolutionary algorithms. It is expected that the segmentation technique when applied to other evolutionary algorithms would lead to improvement in the results in comparison to the results obtained using the conventional evolutionary techniques.

From the standard deviation values and the box plots of three functions, it is clear that segmentation improves the convergence of the function and also it converges to an optimum solution as shown by the best fitness values in Table III.

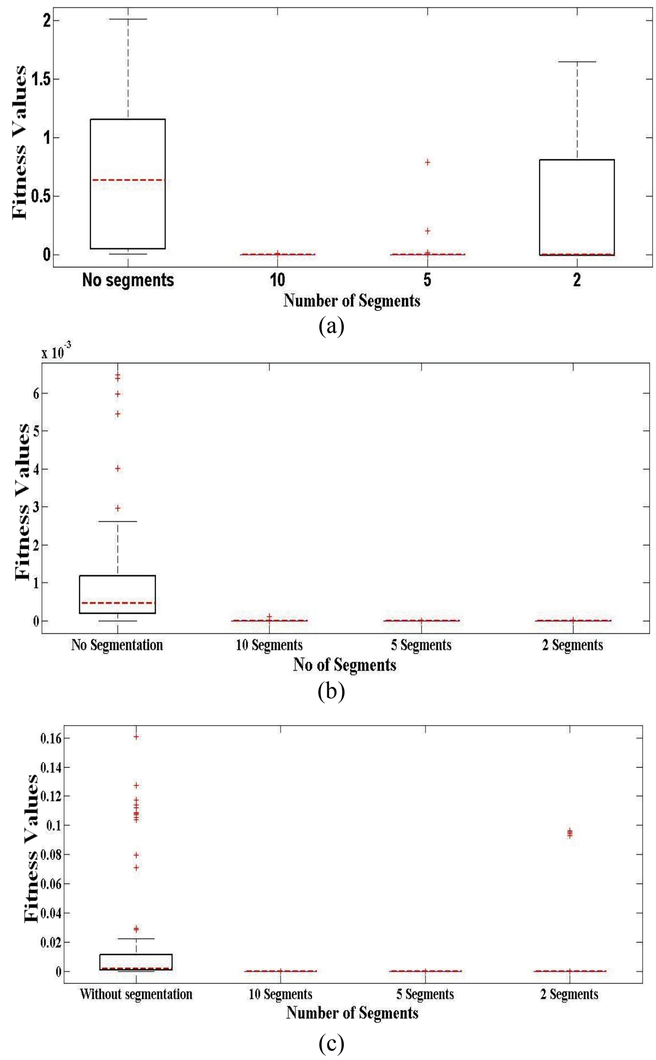

Fig. 3. Box Plots (a) Ackley function (b) Griewank function (c) Penalty#1 function

III. Impact of Number of Segments

Next, the effect of increasing the number of segments on the optimum solution obtained is investigated. Increasing the number of segments result in a better optimized solution as it has been demonstrated from results obtained in Table II and Table III. However, as the number of segments is increased, the GA does not have sufficient number of generations within each segment to effectively optimize the solution. The fitness value increases again by increasing the number of segments after a certain limit. In Table V, the average fitness values for the 14 benchmark functions are calculated with the solution size of 20 and the maximum number of generations as 1000 for the number of segments. This is the reason why the Fletcher-Powell function and the Schwefel 2.26 functions do not perform well when segmentation is applied to them.

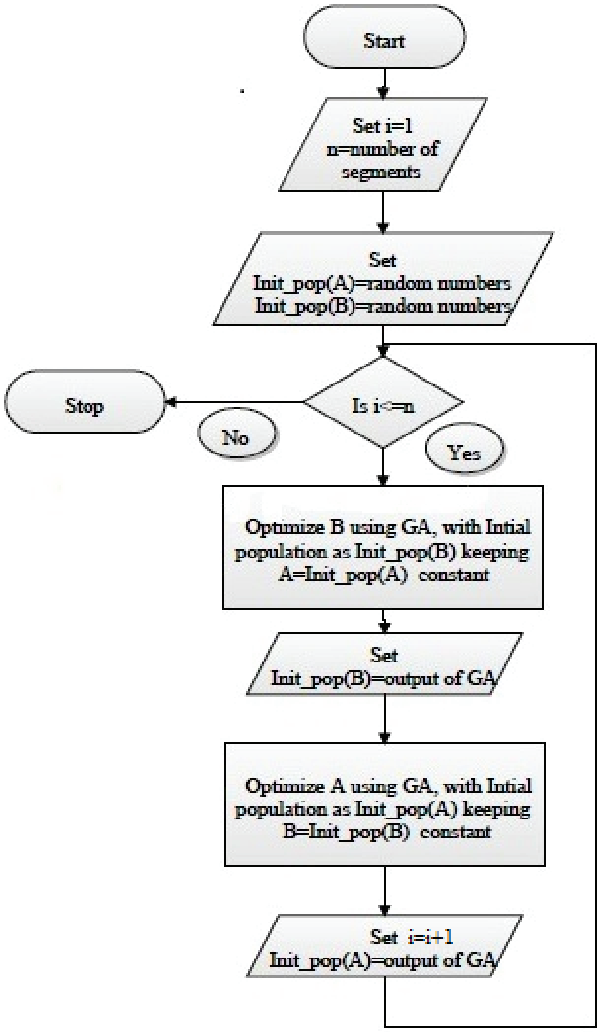

Fig. 2. Flow Chart

IV. Application in Antenna Array

Finally, the proposed technique is applied on a linear antenna array problem. The radiation characteristics of a linear antenna array depend on relative magnitude and phase of the excitation current of each radiating element and also on the separation distance between the array elements [20-23]. By controlling these parameters an antenna array can be designed to produce almost any arbitrary desired pattern. A similar technique to that of the segmentation technique was employed by the authors in [18] to minimize Side Lobe Level (SLL) and beam width. In [24] a three segment technique was repeated in cyclic fashion to minimize SLL and beam width. The array factor of an element array assuming that an element is placed at the origin is given as follows:

$$\mathrm{AF}(\theta)=1+\sum_{i=1}^{n-1} a_i e^{-j(\beta d_i \cos(\theta)+\phi_i)} \quad (1)$$

where n - number of elements, d_i = distance of the i^{th} element from origin, a_i - amplitude excitation of the i^{th} element, and Φ_i - phase of the i^{th} element.

In this paper only the spacing between the elements are optimized to reduce the SLL. The phase of each element is taken as zero and the amplitude of excitation is taken as unity. The antenna geometry of a uniformly spaced array is shown in Fig.4. Here "λ" indicates the wavelength of operation.

TABLE IV

STANDARD DEVIATION VALUE FOR 100 SIMULATION

	Segmentation (Ten segments)	Segmentation (Five segments)	Segmentation (Two segments)	Without Segmentation
Ackley	0.0017	0.0814	0.4899	0.6012
Fletcher-Powell	9.71E+05	9.85E+05	8.49E+05	2.61E+05
Griewank	1.18E-05	8.77E-08	8.53E-07	1.30E-03
Penalty #1	7.10E-06	1.21E-06	1.87E-02	3.68E-02
Penalty #2	4.60E-03	8.10E-03	7.70E-03	7.50E-03
Quartic	1.6444E-08	1.5248E-09	1.5731E-09	0.00018324
Rastrigin	1.9154	2.1104	2.7014	3.3578
Rosenbrock	14.8099	18.0496	17.9675	19.7035
Schwefel 1.2	3.6666	6.2465	8.5124	0.3
Schwefel 2.21	0.0868	0.1213	0.1855	0.5779
Schwefel 2.22	8.27E-02	9.99E-02	1.38E-01	6.03E-01
Schwefel 2.26	1.23E-06	2.24E-07	1.21E-07	1.60E-03
Sphere	2.3942E-06	7.2617E-07	8.3144E-06	0.01
Step	0	0	0.5925	1.0833

TABLE V

AVERAGE FITNESS VALUE FOR 100 SIMULATIONS

	No. of Segments (20 segments)	No. of Segments (25segments)	No. of Segments (50 segments)	No. ofSegments (100 segments)	No. of Segments (125 segments)	No. of Segments (200 segments)
Ackley	0.0011	0.001	0.0045	0.0236	0.0468	0.1353
Fletcher-Powell	2.65E+06	2.55E+06	2.62E+06	2.71E+06	2.63E+06	2.65E+06
Griewank	1.82E-06	5.93E-07	6.52E-06	7.84E-05	2.18E-04	1.30E-03
Penalty #1	1.10E-03	8.74E-06	8.28E-05	2.80E-04	5.06E-04	3.20E-03
Penalty #2	1.40E-03	1.60E-03	1.22E-04	4.58E-05	2.00E-04	1.80E-03
Quartic	1.95E-09	9.08E-09	3.62E-07	5.94E-06	3.53E-05	8.32E-04
Rastrigin	3.273	2.6746	0.6656	0.2696	0.4601	2.5541
Rosenbrock	10.1701	8.7405	7.3607	9.9479	10.3792	14.58
Schwefel 1.2	4.0333	2.2925	0.5106	0.1593	0.1958	0.4641
Schwefel 2.21	0.0417	0.0241	0.0168	0.0645	0.1313	0.3427
Schwefel 2.22	0.3678	0.3403	0.2506	0.1721	0.1706	0.187
Schwefel 2.26	-7.89E+01	-7.89E+01	-7.89E+01	-7.89E+01	-7.89E+01	-7.88E+01
Sphere	4.41E-06	7.37E-06	5.81E-05	8.88E-04	0.0029	0.0169
Step	0	0	0	0	0	0

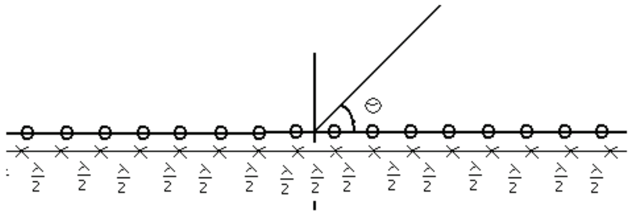

Fig. 4. The geometry of an *N* element symmetric linear array

The function which has to be minimized is given by

$$f(d_1, d_2, d_3, ..., d_{n-2}, d_{n-1}) = max(20 \log(mod(AF(\theta)))) \quad (2)$$

Subject to

$\theta = (0°, 90°-(BW/2)) \cup (90°+(BW/2), 180°)$

Constraints: $|(d_{i+1} - d_i)|/\lambda > 0.35$

This condition avoids mutual coupling between the adjacent elements of the array.

The distances $d_1, d_2, d_3 ... d_{n-2}, d_{n-1}$ are normalized with respect to λ.

First the problem is solved by using a genetic algorithm with the following parameters:

No. of Variables: 11
Population size: 20
Crossover rate: 0.8
Crossover function: Two point crossover
Mutation: Gaussian
Mutation Rate: 0.35
Initial population: random numbers
Scaling Function: Rank
Selection Function: Stochastic Uniform
No of generations: 10000

The program was run on an Intel Centrino Duo computer and the MATLAB GA toolbox is used for this purpose. The solution after 10000 generations is given along with SLL in Table VI.

TABLE VI
BEST FITNESS VALUE IN 100 SIMULATIONS WITHOUT SEGMENTATION FOR N=12

d_i's Distances in terms of λ from origin				SLL (dB)
1.3181	1.9410	2.7964	3.8171	-15.05
4.5400	5.0096	5.6885	6.5285	
7.2988	7.8876	8.4644		

Next the segmentation technique is used for optimization.

STEP 1: Organize the entire solution set into subsets. Here the entire solution set is sub grouped into two sets consisting of 5 and 6 variables each. This can be called as segmenting or grouping the entire solution set into subsets. The number of segments is taken as 20.

STEP 2: Optimize the first subset keeping the other subset constant for 500 generations.

STEP 3: Substitute the values for the variables in the first group obtained from second step and now optimize the second set of variables for the next 500 generations

Next, the segmentation technique is applied for minimizing the SLL. The number of segments is taken as 20. The result of optimization using the segmentation technique is shown in Table VII.

TABLE VII
BEST FITNESS VALUE IN 100 SIMULATIONS WITH SEGMENTATION FOR N=12

d_j's Distances in terms of λ from origin				SLL (dB)
1.0404	1.8159	2.6779	3.3985	-17.18
3.8282	4.5391	5.3487	6.0466	
6.713	7.5750	8.3315		

The results obtained are the best minima in 100 simulations. The SLL obtained by the proposed method is -17.18dB after 10000 generations while that obtained by the conventional GA is -15.05 dB confirming that the segmentation has resulted in achieving an optimum solution and the radiation patterns for both are shown in Fig.6. This also shows that SLL obtained using the segmentation technique is lower than that obtained with conventional GA. Both the radiation patterns obtained with and without segmentation had the same beam width of 14°. Also the results obtained for n=20 is shown in Fig. 6 and results are given in Table VIII and Table IX.

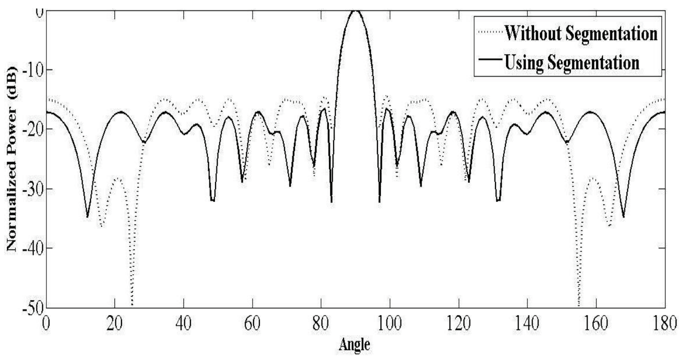

Fig. 5. Array pattern of 12 element array

TABLE VIII
BEST FITNESS VALUE IN 100 SIMULATIONS WITHOUT SEGMENTATION FOR N=20

d_i's Distances in terms of λ from origin				SLL (dB)
0.6400	1.7049	2.4544	3.1047	-17.66
3.9406	4.7404	5.1891	5.7991	
6.5550	6.9640	7.4523	8.4458	
9.1824	9.9845	10.7059	11.6743	
12.623	13.3029	14.3646		

TABLE IX
BEST FITNESS VALUE IN 100 SIMULATIONS WITH SEGMENTATION FOR N=20

d_j's Distances in terms of λ from origin				SLL (dB)
1.3500	2.1576	3.0169	3.7202	-18.77
4.5530	5.0282	5.4916	6.0605	
6.5751	7.1174	7.6555	8.2344	
8.9311	9.6178	10.0893	10.5541	
11.1223	11.9116	12.6719		

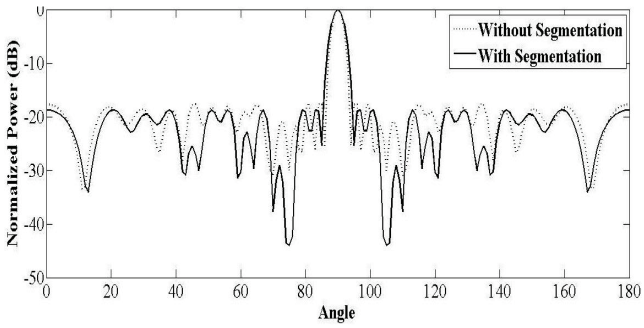

Fig. 6. Array pattern of 20 element array

The results obtained show the superiority of segmentation technique used in GA over conventional GA. However, the proposed GA is not compared with other evolutionary algorithms such as Particle Swarm Optimization (PSO), Firefly Algorithm (FA) etc. It is expected that in case the segmentation technique is explored with these evolutionary algorithms, it may result in improved performance.

V. Conclusion

A novel pragmatic approach for improving the GA is presented. The proposed technique is tested on 14 benchmark functions and it is found that the proposed technique outperforms the conventional GA. Also the effect of number of segments of the fitness value has been analysed. The application of this technique for the practical engineering problem of side lobe reduction in antenna arrays has been presented. Results show that the proposed technique is superior in improving the final solution i.e. in optimizing the fitness function. In the proposed segmentation technique, the search space is segmented at regular intervals, thereby improving the chance for the GA to optimize the fitness function. The proposed technique divides the chromosomes into two sets and the authors intend to consider the analysis regarding the number of sets (i.e. greater than two) in their future work. Also, minimizing the computational time would enhance the pragmatic applications of the proposed technique and the authors intend to address this issue in their future work.

References

[1] C. A. Floudas, and P. M. Pardalos, “Recent Advances in Global Optimization”, *Princeton University Press*, Princeton, NJ, 1992.
[2] J. Neter, W. Wasserman, and G. A. Whitmore, “Applied Statistics”, *Allyn and Bacon Inc*., Boston, 1988.
[3] P. M. Pardalos, and H. E. Romeijn, *Handbook of Global Optimization*, Kluwer Academic Publishers, Boston, 2002.
[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, 1989.
[5] F. Glover, and G. A. Kochenberger, *Handbook of Meta-heuristics*, Kluwer Academic Publishers, Boston, 2003.
[6] M. Heikki, K. Miettinen, and A. Penttinen, “On Initial Populations of a Genetic Algorithm for Continuous Optimization Problems”, *Journal of Global Optimization*, vol. 37, pp. 405-436, 2007.
[7] A. Prügel-Bennett, "Benefits of a Population: Five Mechanisms that Advantage Population-Based Algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 500-517, 2010.
[8] Y.-W. Leung, and Y. Wang, "An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41-53, 2001.
[9] K. Krishnakumar, “Micro-genetic Algorithms for Stationary and Non-stationary Function Optimization”, *SPIE: Intelligent Control and Adaptive Systems*, vol. 11, pp. 289-296, 1989.
[10] J. K. Cochran, Sh.-M. Horng, and J. W. Fowler. "A Multi-population Genetic Algorithm to Solve Multi-objective Scheduling Problems for Parallel Machines", *Computers & Operations Research*, vol. 30, no. 7, pp. 1087-1102, 2003.
[11] Z. H. Cui, J. C. Zeng, and Y. B. Xu, "A New Nonlinear Genetic Algorithm for Numerical Optimization", *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4660-4663, 2003.
[12] W. Paszkowicz, "Properties of a Genetic Algorithm Equipped with a Dynamic Penalty Function", *Computational Materials Science*, vol. 45, pp. 77-83, 2009.
[13] S. F. Hwang, R. S. He, "A Hybrid Real-parameter Genetic Algorithm for Function Optimization", *Advanced Engineering Informatics*, vol. 20, pp. 7-21, 2006.
[14] O. Hrstka, and A. Kučerová, "Improvements of Real Coded Genetic Algorithms Based on Differential Operators Preventing Premature Convergence", *Advances in Engineering Software*, vol. 35, pp. 237-246, 2004.
[15] D. Li, W. Wang, and F. Ismail, "An Enhanced GA Technique for System Optimization", *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, pp. 1471-1476, 2012.
[16] D. Simon, "Biogeography Based Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 12, no 6, pp. 702-713, 2008.
[17] M. Molga, and C. Smutnicki, "Test Functions for Optimization Needs", 2005, Available at http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.
[18] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 82-102, 1999.
[19] Z. Cai, and Y. Wang, "A Multiobjective Optimization-based Evolutionary Algorithm for Constrained Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 658-675, 2006.
[20] C. L. Dolph, “A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beamwidth and Side-lobe Level”, *Proc. IRE*, vol. 34, pp. 335-348, 1946.
[21] R. Harrington, “Sidelobe Reduction by Nonuniform Element Spacing”, *IRE Transactions on Antennas and Propagation*, vol. 9, pp. 187-192, 1997.
[22] M. I. Skolnik, G. Nemhauser, and J. W. Sherman, III, “Dynamic Programming Applied to Unequally Spaced Arrays”, *IEEE Trans. Antennas Propag.*, vol. 12 pp. 35-43, 1964.
[23] B. P. Kumar, and G. R. Branner, “Design of Unequally Spaced Arrays for Performance Improvement”, *IEEE Trans. Antennas Propag., Special Issue on Phased Arrays*, vol. 47, pp. 511-523, 1999.
[24] A. Bhargav, and N. Gupta, "Multiobjective Genetic Optimization of Nonuniform Linear Array with Low Sidelobes and Beamwidth", *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 1547-1549, 2013.