Impacts of Genetic Algorithm Parameters on the Solution Performance for the Uniform Circular Antenna Array Pattern Synthesis Problem

F. Yaman[1] , A. E. Yılmaz*[2]

[1,2] Electronics Engineering Department, Ankara University,
Tandogan, Ankara, Turkey, 06100
*aeyilmaz@eng.ankara.edu.tr
fyaman@eng.ankara.edu.tr

ABSTRACT
In this paper, the uniform circular antenna array pattern synthesis problem is solved by means of the real coded genetic algorithm (GA). At the same time, the impacts of the mutation rate and the crossover position on the GA performance are also investigated. For this purpose, a circular antenna array with uniformly spaced isotropic elements having identical excitation amplitudes is used as a model. Unlike the conventional GA (with fixed mutation rate and random crossover positions), typical GA implementations with variable mutation rate and restricted crossover position are considered for performance improvement. In conclusion, for the specific problem, decreasing mutation rate with negative derivative is observed to be outperforming the implementations with different mutation rate behaviors. Moreover, regarding the crossover technique, it is observed that imposing some restrictions on the crossover positions (rather than fully random position selection) yields better solutions.

Keywords: Circular antenna array, pattern synthesis, genetic algorithm, mutation rate, crossover point.

RESUMEN
En este trabajo, se le da solución a un problema de síntesis de patrones de arreglo de antenas circular uniforme por medio del algoritmo genético con codificación real (GA). Se investigan, al mismo tiempo, los impactos del índice de mutación y la posición de cruce sobre el desempeño del GA. Con tal propósito, se utiliza un arreglo de antenas circular con elementos isotrópicos espaciados uniformemente con amplitudes de excitación idénticas. A diferencia del GA convencional (con índice de mutación y posiciones de cruce aleatorias), se consideran implementaciones de GA típicas con índice de mutación variable y posición de cruce restringida para la mejora del desempeño. En conclusión, para el problema en cuestión, se observa que un índice de mutación descendiente con derivativa negativa supera las implementaciones con comportamientos de índice de mutación diferentes. Además, con relación a la técnica de cruce, se observa que imponer algunas restricciones sobre las posiciones de cruce (en lugar de la selección de posición completamente aleatoria) arroja mejores soluciones.

1. Introduction

The genetic algorithm (GA) is a probabilistic search method based on the principles of Darwin’s natural selection and evolution theory. So far, GA has efficiently been used for the solution of combinatorial optimization problems [1-4]. In this sense, it has been quite successful in many engineering applications. Additionally, it has also been applied to certain problems in the electromagnetic theory. The antenna array synthesis problem constitutes a major portion of such applications.

The main purpose of the construction of antenna arrays, which is one of the popular research topics in electromagnetics, is nothing but to obtain the desired pattern (which cannot be achieved with a single antenna generally) with minimum error. In such a problem, electrical specifications and geometrical placements of the array elements are important.

Generally, the main expectations of the designer from antenna arrays are radiation patterns consisting of narrow main beams in the desired direction, and low-level side beam. Quite different antenna array configurations might be constructed but, due to their simplicity, the most widely applied configurations are linear, planar and circular arrays [5-6].

In the literature, there are studies on antenna array design, synthesis and pattern forming using optimization methods such as the genetic algorithm and particle swarm optimization [7-9]. In [7] and [8], the side lobe reduction on circular antenna arrays was aimed by using the method of the genetic algorithms and particle swarm optimization. In [9], radiation pattern synthesis was performed through a hybrid GA algorithm consisting of simplified quadratic interpolation and real coded GA.

The effect of mutation and crossover on the GA performance has been the focus of attention of many researchers since the beginning of the 1990s. There are currently a considerable number of studies in the literature on this subject [10-23]. In [10], adaptive crossover and mutation probabilities have been realized in order to maintain the diversity in the population while, at the same time, sustaining the convergence capacity of the GA. In [11], a mutation rate strategy, which principally depends on a variable (per individual) mutation rate based on the individual's relative performance within a given generation, was proposed. In [12], in order to improve the performance of genetic algorithms, three variable mutation rate schemes were described for the fuzzy logic controller design problem. In [13], a per-individual mutation-rate strategy, which is based on determining the mutation rate according to the fitnesses of the individuals in the population, was proposed. In [14-16], the adaptive mutation rate was used for improvement in the GA performance. The floating point crossover and the mutation approach were discussed in [17]; some GA variants (hybridized or parallelized versions) with variable mutation and crossover rate have also been proposed and applied for optimization of bound constrained non-linear multi-modal functions in [18-20]. Studies conducted with similar motivations but concerning the mutation rates in genetic programming are also available in the literature (such as [21]).

Regarding the "restrictions" on the crossover, a considerable number of studies exists in the literature. Even though the nomenclature of "restricted crossover" can be found in some other publications (such as [22] and [23]), the term has been used in different senses. For example, in order to apply the genetic algorithm in permutation-type problems (such as the traveling salesman or the generalized assignment problem), due to the nature of the problems, specialized crossover operators should be defined in order to guarantee the validity of the relevant children [24]. But in this study, even though the problem imposes no specific and explicit restrictions to the crossover operator, we define a constraint yielding a "restricted crossover" operator while seeking some means of performance improvement.

In this study, unlike the other studies in the literature, methods for performance improvement are investigated in order to obtain a desired pattern of circular antenna array with GA. For this purpose, instead of the fixed mutation rate approach used in the classical GA, the mutation rate is reduced along with new generations, which become increasingly more qualified throughout the evolution process; and these decreasing mutation rates are categorized to compare the impact of the mutation rate variation on the performance.

As a second aspect, the impact of the crossover scheme to the solution performance is investigated. Instead of determining the crossover point in a totally random fashion, the probable crossover points have been kept limited in such a manner that the sub-gene groups representing the angular position of a particular element of the antenna array is preserved. In other words, in the restricted crossover scheme, the crossover is only applied to the joints of the angular positions of the antenna array elements.

As stated before, the effects of variable mutation and crossover rates as well as restricted crossover strategies have been studied many times so far. To our belief, the originality of this study comes from the following reasons:

- Even though in some previous publications (such as [12] and [18]), it has been mentioned that a decreasing mutation rate yields better performance, to our knowledge, there exists no publication explicitly explaining how to decrease the mutation rate ideally. From this aspect, the paper comes up with an answer to such a question.

- The experiments in this study are performed on an antenna array synthesis problem. In this study, for this particular problem, it is also demonstrated that preserving the integrity of fittest sub-gene groups corresponding to some parameters (i.e. the angular positions of the element in this problem) by means of a restricted crossover operator. This could be considered as a second aspect of contribution to further studies of this sort.

After this introductory section, in Section 2, basic definitions of GAs are revisited. In Section 3, formulations regarding the circular antenna arrays and the relevant synthesis problem are presented. In Section 4, material and method are presented together with the obtained results and relevant discussions. Section 5 concludes the paper with some remarks.

2. Genetic Algorithm (GA)

The genetic algorithm (GA) was first introduced in 1975 by Holland [1]. Dense usage of this algorithm has been realized especially after Goldberg's studies [2]. The genetic algorithm provides the necessary solution yielding the global minimum or maximum values of multidimensional and complicated functions [2-3]. They are used widely for the solution problems which are considered very difficult for conventional optimization methods.

GA simulates the survival of the fittest among individuals over consecutive generations throughout the solution of a problem. Each generation consists of a population of character (usually binary) strings that are analogous to the chromosomes. Each individual represents a point in the search space and a solution candidate. The individuals in the population are then exposed to the process of evolution. Genes from good individuals propagate throughout the population. Thus, each successive generation will literally become more suited to its environment. In the optimization terminology, this corresponds to the situation that newer generations have better fitness values [1-4]. Figure 1 presents the general structure of the genetic algorithm.

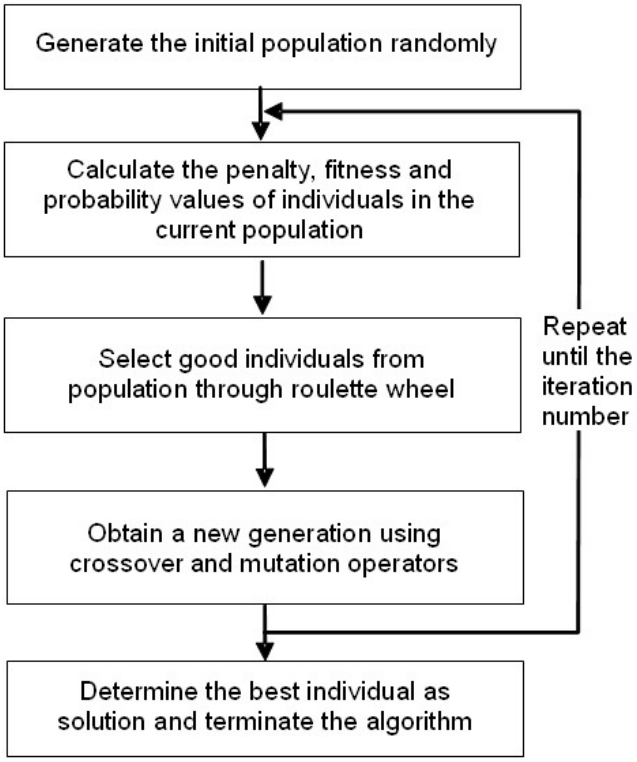

Generate the initial population randomly
Calculate the penalty, fitness and probability values of individuals in the current population
Select good individuals from population through roulette wheel
Obtain a new generation using crossover and mutation operators
Repeat until the iteration number
Determine the best individual as solution and terminate the algorithm

Figure 1. General structure of the genetic algorithm.

3. Circular Antenna Arrays

Antenna arrays are antenna systems which are created by combining different or similar antennas in different forms. Antenna arrays are used to provide the desired specifications such as low-level side lobe, narrow main lobe and high directivity [5-6].

Furthermore, thanks to adding together power of the elements in the array, high-power radiation pattern is achieved, and without requiring mechanical movement, the main beam can be moved to a desired direction [5-6]. This yields a very wide application spectrum for the antenna arrays. Arrays may have different geometric shapes. If elements of the array are located on a circle, this type of array is called a “circular antenna array” as seen in Figure 2.

The radiation pattern of the circular antenna array consisting of N isotropic elements can be expressed by means of the array factor. The array factor of the circular antenna array can be formulated as follows [8]:

$$AF(\phi, I, dm) = \sum_{n=1}^{N} I_n e^{jka\left[\cos(\phi-\phi_n)-\cos(\phi_0-\phi_n)\right]} \qquad (1)$$

Here, AF is the array factor, N is the number of elements in the array, I_n is the excitation amplitude of the nth element, a is the radius of the circle, $k=2\pi/\lambda$ is the wavenumber, ϕ_0 is angle of direction of the main beam, ϕ_n is the angular position of the nth element with respect to ϕ_0. ka and ϕ_n are given by

$$ka = \frac{2\pi a}{\lambda} = \sum_{i=1}^{N} dm_i \qquad (2)$$

$$\phi_n = \frac{2\pi \sum_{i=1}^{n} dm_i}{\sum_{i=1}^{N} dm_i} \qquad (3)$$

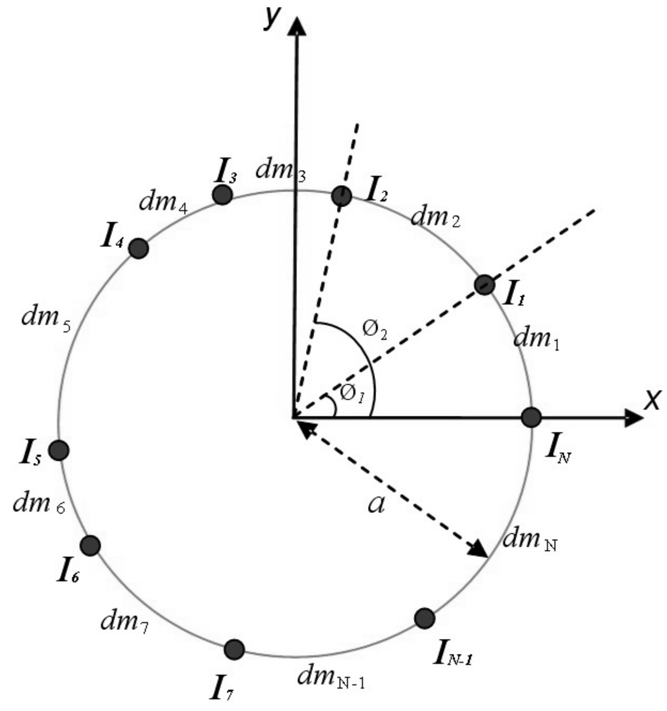


Figure 2. Geometry of a uniform circular antenna array with N isotropic elements.

4. Material and Method

In our study, a circular antenna array with 8 uniformly spaced isotropic elements having identical excitation amplitudes is used as a model.

For this purpose, the radiation pattern of a circular antenna array, for which

- the angle of direction for the main beam is equal to zero degree;

- the angular positions of the elements (ϕ_n) are 25, 70, 115, 160, 205, 250, 295 and 340°; and

- the radius of the circle is 20 cm,

is determined as reference. The Half Power Beam Width of the reference pattern is chosen to be 56°, and the operating frequency is chosen to be 750 MHz.

To obtain the desired pattern through the genetic algorithm, an initial population consisting of 100 individuals is created. Then, angular positions of 8 elements are converted to binary codes. Afterwards, the created population is subjected to the genetic algorithm operators: selection, crossover and mutation.

Throughout the study; all comparisons in all experiments were performed by considering independent executions of each GA scheme in order to eliminate the “chance factor” and to be able to perform a fair comparison. Table 1 lists the parameter setup used throughout the experiments in this study.

Parameter	Values used in GA
Number of individuals in the population	100
Element number of array	8
Number of iterations (generations)	250
Crossover rate	0.2
Initial mutation rate	0.01
Number of independent executions	300

Table 1. Parameter values used in the genetic algorithm.

In the implementation, the radiation diagram is divided into 360 equal parts in ($-\pi, \pi$) radians. Thus, a sensitivity of one-degree is achieved. GA is implemented in MATLAB 7.3.0, but no third party tool or another standard product or library (such as MATLAB Genetic Algorithm Toolbox) is used. The main reason for this is to preserve the flexibility to modify the architecture of the GA in our ongoing/future studies in order to investigate further performance improvement possibilities.

4.1. The Impact of the Mutation Rate

In addition to the constant mutation rate approach of the classical genetic algorithm, increasing and decreasing mutation rates are examined in this study. As mentioned in previous studies, such as [12] and [18], results show that a decreasing mutation rate along with the generations outperform constant and increasing mutation rates. Nevertheless, an increasing mutation rate along with the generations reduce the performance (i.e. worse than the constant mutation rate).

One of the main aims in this study is to determine the best mutation rate to obtain the best algorithm performance. For this purpose, an initial mutation rate mr_0=0.01 is chosen, and the performances of different mutation rates changes are examined. We use five decreasing mutation rate schemes, mainly. These are

- constant mutation rate,
- linearly decreasing mutation rate,
- decreasing mutation rate with increasing derivative,
- decreasing mutation rate with decreasing derivative,
- mixed type decreasing mutation rate.

In the equations, itn represents the iteration (generation) number.

Scheme-1: Constant Mutation Rate

In this scheme, the mutation rate equals the initial mutation rate for all iterations.

$$mr_1 = mr_0 \tag{4}$$

Scheme-2: Linearly Decreasing Mutation Rate

In this scheme, the mutation rate decreases linearly. In the following equation, c_1 (>0) determines the slope of the decay.

$$mr_2 = mr_0 - itn \times c_1 \tag{5}$$

In Figure 3, different mutation rate behaviors for different values of c_1 are given.

Among those, our tests show that mr_{22} scheme outperforms the mr_{21} and mr_{23} schemes, as seen in Figure 4.

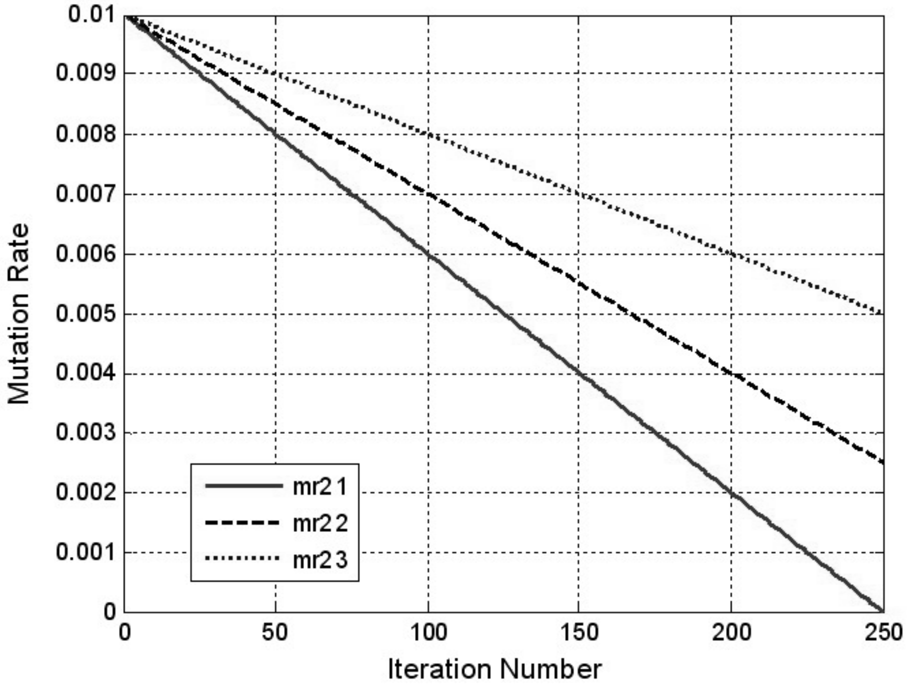
Scheme-3:Decreasing Mutation Rate with Increasing Derivative

In this scheme, the mutation rate decreases in a nonlinear manner so that its derivative is increasing. Such a characteristic can be parametrically expressed as

$$mr_3 = mr_0 - c_2 \times itn^{1/c_3} \tag{6}$$

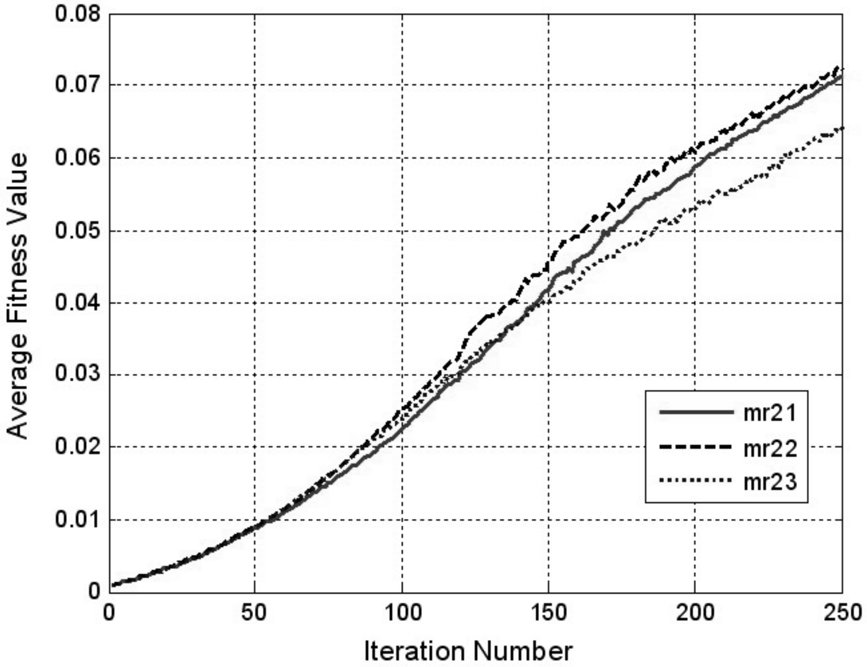
where c_2 and c_3 nonnegative real numbers. In Figure 5, different mutation rate behaviors for different values of c_2 and c_3 are given.

Among those, our tests show that mr_{33} scheme outperforms the mr_{31} and mr_{32} schemes, as seen in Figure 6.



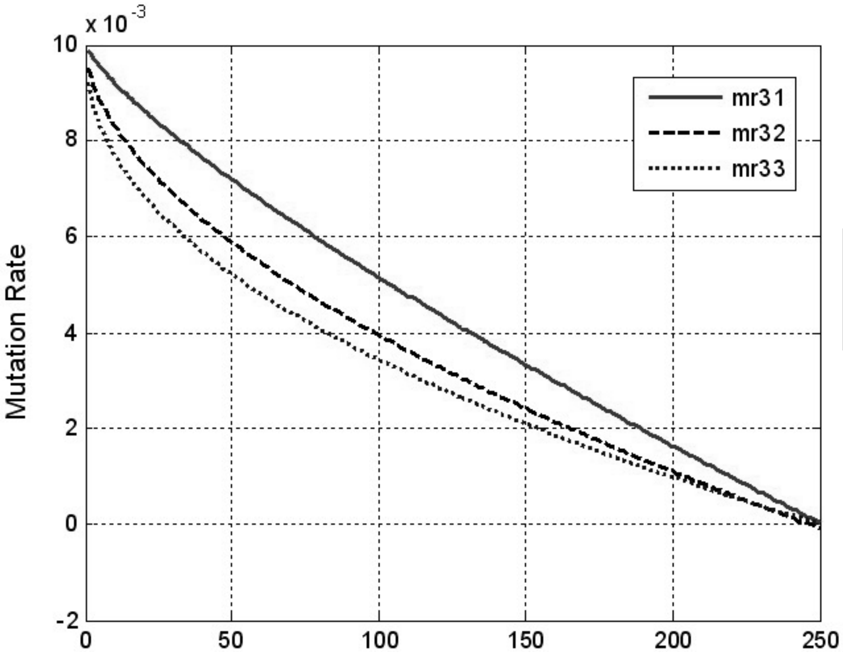
mr_{21} = 0.01 – 4×10^{-5}×itn
mr_{22} = 0.01 – 3×10^{-5}×itn
mr_{23} = 0.01 – 2×10^{-5}×itn

Figure 3. Linearly decreasing mutation rates (mr_{21}, mr_{22} and mr_{23} yielded by c_1=4x10$^{-5,}$ c_1=3x10^{-5} and c_1=2x10^{-5}, respectively).



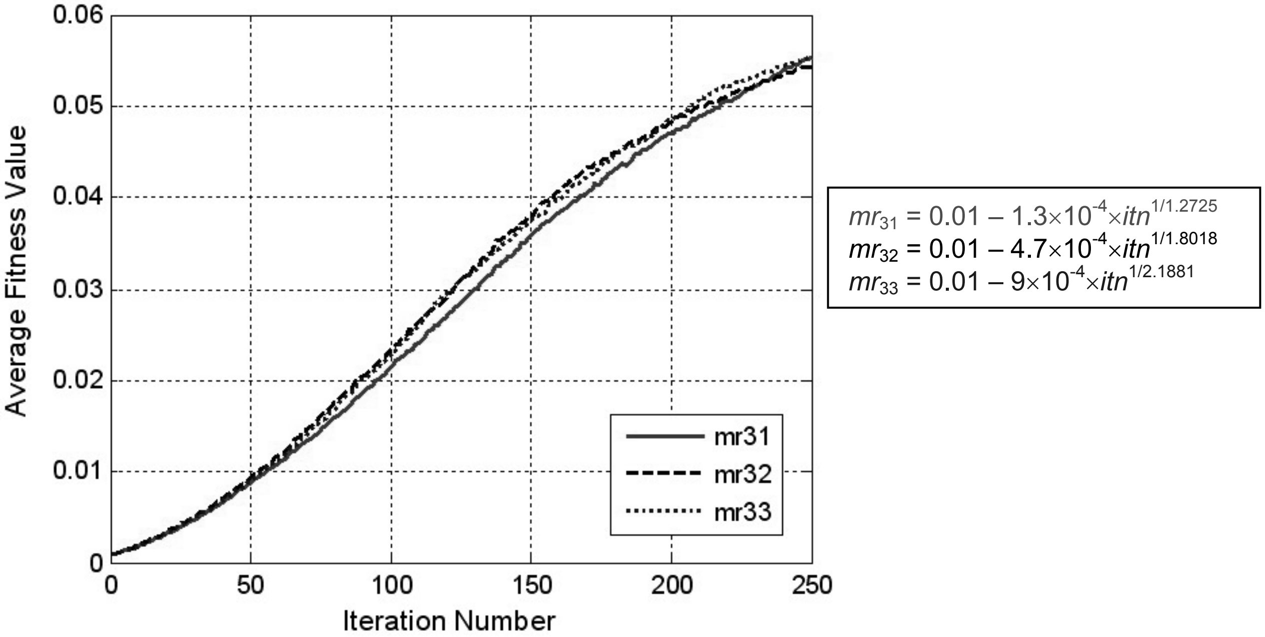
$mr_{21} = 0.01 - 4\times10^{-5}\times itn$
$mr_{22} = 0.01 - 3\times10^{-5}\times itn$
$mr_{23} = 0.01 - 2\times10^{-5}\times itn$

Figure 4. Population average fitness values of linearly decreasing mutation rates (computed after 300 independent experiments).



$mr_{31} = 0.01 - 1.3\times10^{-4}\times itn^{1/1.2725}$
$mr_{32} = 0.01 - 4.7\times10^{-4}\times itn^{1/1.8018}$
$mr_{33} = 0.01 - 9\times10^{-4}\times itn^{1/2.1881}$

Figure 5. Decreasing mutation rates with increasing derivative (mr_{31}, mr_{32} and mr_{33} yielded by (c_2=1.3x10^{-4}, c_3=1.2725), (c_2=4.7x10^{-4}, c_3=1.8018), and (c_2= 9x10^{-4}, c_3=2.1881), respectively).



$mr_{31} = 0.01 – 1.3\times10^{-4}\times itn^{1/1.2725}$
$mr_{32} = 0.01 – 4.7\times10^{-4}\times itn^{1/1.8018}$
$mr_{33} = 0.01 – 9\times10^{-4}\times itn^{1/2.1881}$

Figure 6. Population average fitness values of decreasing mutation rates with increasing derivative (computed after 300 independent experiments).

Scheme-4: Decreasing Mutation Rate with Decreasing Derivative

In this scheme, the mutation rate decreases in a nonlinear manner so that its derivative is also decreasing. Such a characteristic can be parametrically expressed as

$$mr_4 = mr_0 - c_4 \times itn^{c_5} \qquad (7)$$

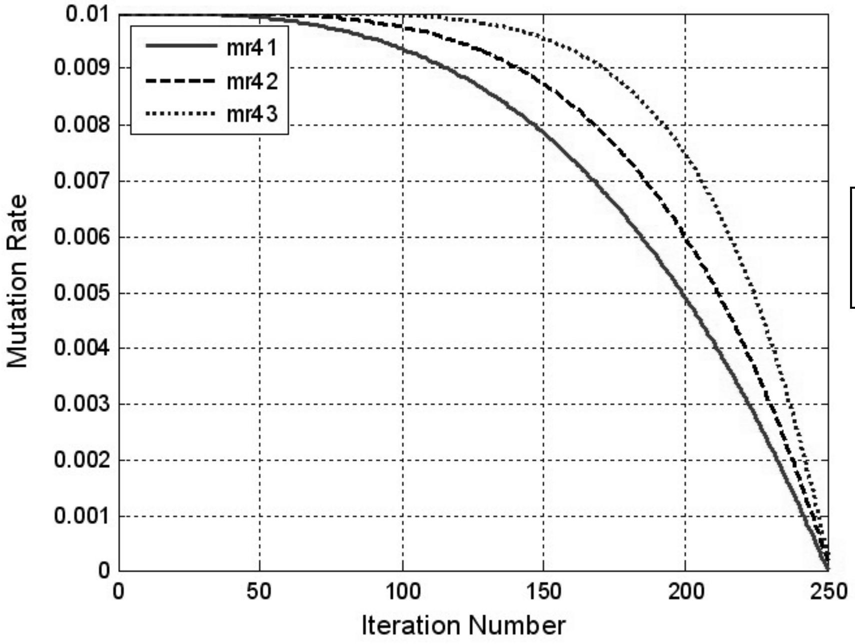
where c_4 and c_5 nonnegative real numbers. In Figure 7, different mutation rate behaviors for different values of c_4 and c_5 are given.

Among those, our tests show that mr_{42} scheme outperforms the mr_{41} and mr_{42} schemes, as seen in Figure 8.

Scheme-5: Mixed Type Decreasing Mutation Rate

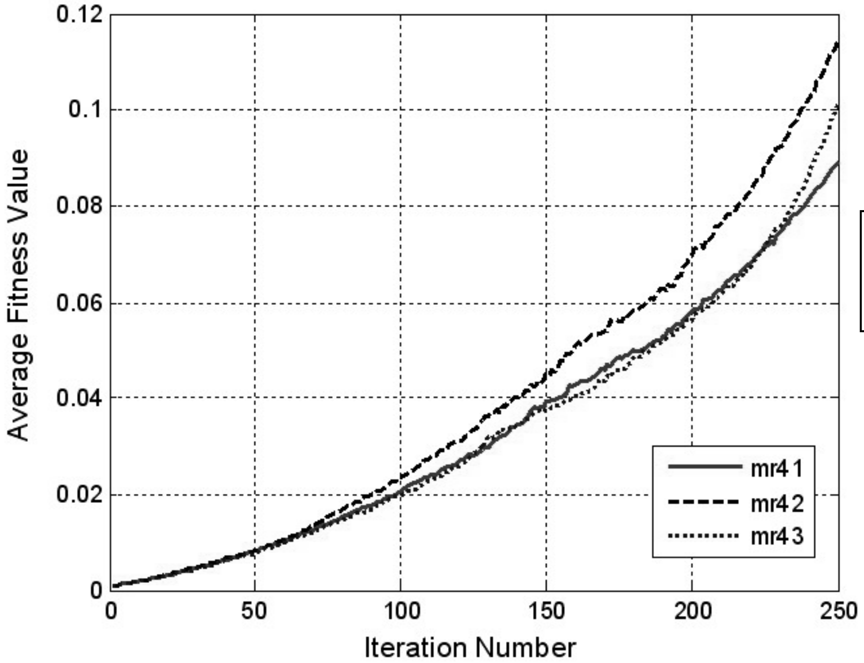
In this scheme, the mutation rate behavior is defined as a cascaded combination of two other schemes. During the first half of all iterations (i.e. from iteration 1 to 125), one of the schemes is applied whereas, during the second half (i.e. from iteration 126 to 250), another scheme is applied. Two particular examples of such numerous variations are given in Figure 9.

Among those, our tests show that mr_{51} scheme outperforms the mr_{52} scheme, as seen in Figure 10.



mr_{41} = 0.01 – 9.3×10$^{-10.163}$×itn^3
mr_{42} = 0.01 – 1×10$^{-11.6}$×itn^4
mr_{43} = 0.01 – 1×10$^{-16.4}$×itn^6

Figure 7. Decreasing mutation rates with decreasing derivative (mr_{41}, mr_{42} and mr_{43} yielded by (c_4=9.3x10$^{-10.163,}$ c_5=3), (c_4=1x10$^{-11.6}$, c_5=4), and (c_4=1x10$^{-16.4}$, c_5=6), respectively).



mr_{41} = 0.01 – 9.3×10$^{-10.163}$×itn^3
mr_{42} = 0.01 – 1×10$^{-11.6}$×itn^4
mr_{43} = 0.01 – 1×10$^{-16.4}$×itn^6

Figure 8. Population average fitness values of decreasing mutation rates with decreasing derivative (computed after 300 independent experiments).

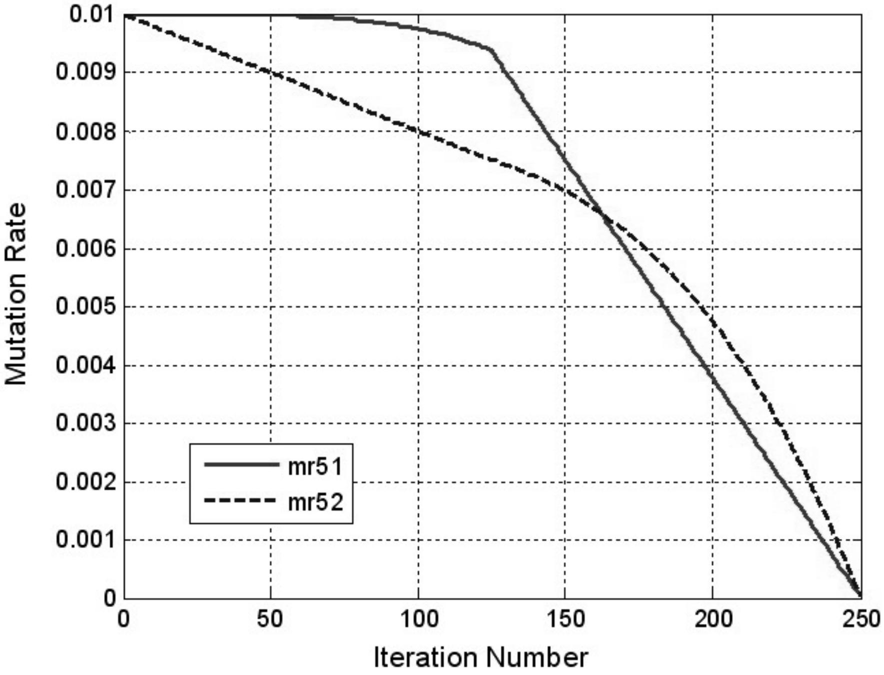


Figure 9. Mixed-type decreasing mutation rates.

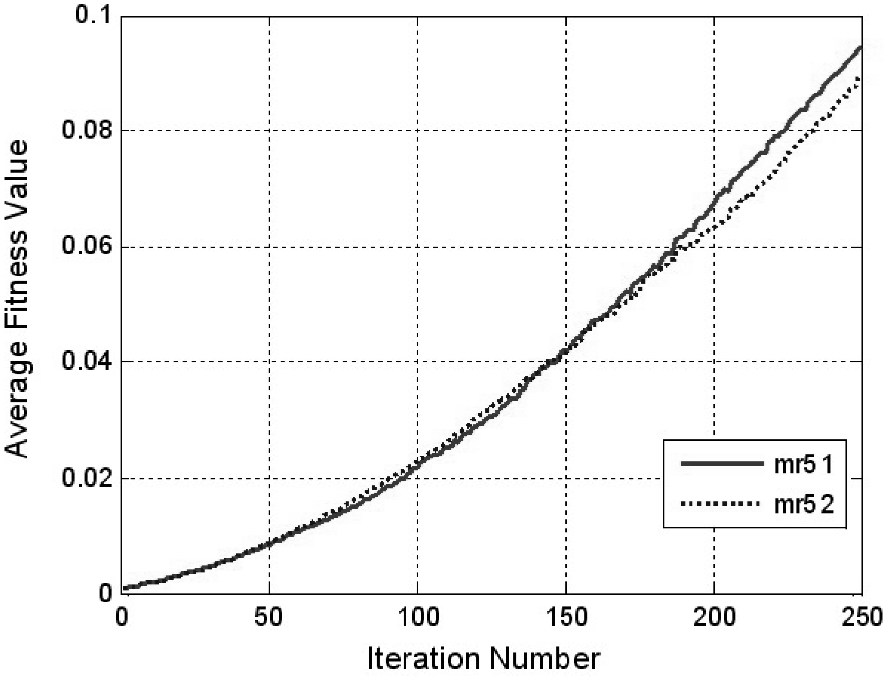


Figure 10. Population average fitness values of Mixed-type decreasing mutation rates (computed after 300 independent experiments).

Next, the outperforming mutation rates of all schemes (mr_1, mr_{22}, mr_{33}, mr_{42} and mr_{51}) are compared, where the scheme with mr_1 corresponds to nothing but the conventional GA. The best average fitness values of each scheme are shown in Figure 11. It is seen that for this particular problem, all mutation rates proposed by us yielded equal or better results than the conventional GA (indicated by mr_1). However, the best GA performance can be obtained via the scheme with decreasing mutation rate with decreasing derivative, more specifically with the mr_{42} scheme.

A qualitative comment on this result can be made as follows: In the early iterations (generations), where the overall population quality is relatively low, mutation is a performance-improving factor yielding diversity in the search process. On the other hand, as long as the population gets qualified, mutation starts to show a negative impact distracting the focus and the concentration of the population.

Angular positions of the circular antenna array elements obtained through five different mutation schemes are shown in Table 2 for comparison.

4.2. The Impact of the Crossover Point

In addition to the study on the impact of different mutation rates on the algorithm performance, the impact of the crossover scheme (i.e. the method of selection/ determination of the crossover point) is also examined. For this purpose, the random crossover point scheme is compared to the so-called “restricted crossover point scheme”. In the restricted crossover point scheme, the crossover position is randomly determined in such a manner that separation of a sub-gene group representing the angular position of a particular array element is not allowed.

In this study, each solution candidate (i.e. candidate antenna array proposed as a solution) has 8 elements; where each element consists of 9 genes. Thus, each individual has 72 genes. The main idea of the restricted crossover point scheme is illustrated in Figure 12.

When compared, it has been observed that the restricted crossover point scheme outperforms the fully random crossover point scheme. The results of both methods are shown in Figure 13. For both methods, the mr_{42} mutation rate is used.

The desired radiation pattern together with the obtained pattern by means of the GA (mr_{42} mutation rate and restricted crossover scheme) is shown in Figure 14.

5 Conclusions

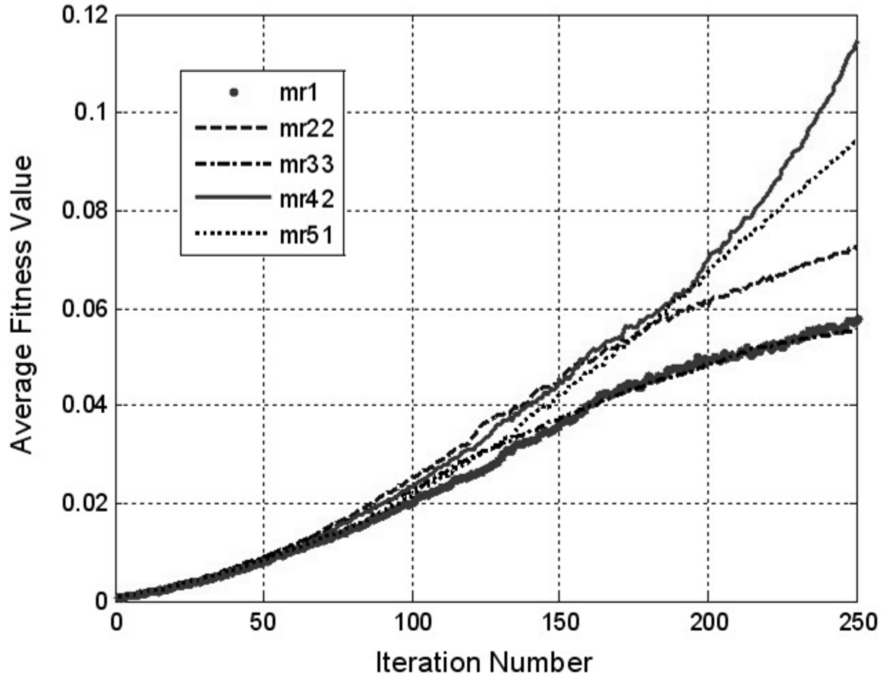
In this study, the impact of the mutation rate and the crossover point on genetic algorithm performance is investigated. For this purpose, a circular antenna array with uniformly spaced isotropic elements having identical excitation amplitudes is used as a model. A fixed mutation rate method, which is used in the conventional genetic algorithm, is compared with several variable mutation rate schemes. Moreover, the ideal scheme for the determination of the crossover point is also investigated.

In general, the following observations can be made regarding the mutation operator: In the early generations, the overall population is of relatively low quality; and hence, mutation is a performance-improving factor, which creates unexpected diversity, but when the population gets qualified, mutation starts to cause loss of high-quality individuals, causes abnormal ones, and eventually decreases the average of the population. Hence, a decreasing mutation rate scheme outperforms others as expected. Moreover, decreasing mutation rate with decreasing derivative (causing the rate to decrease dramatically for late generations) increases the performance more and more.

Regarding the crossover, it can be concluded that preserving the sub-gene groups corresponding to high-quality solutions (throughout the crossover might) yields better results as seen in the particular problem studied here.

In conclusion, for this particular type of antenna array, the scheme of decreasing mutation rate with decreasing derivative yields the best results compared to others. Regarding the crossover, the restricted crossover point scheme outperforms the fully random crossover point scheme.

It should be noted that these results are valid for this particular problem, i.e. the synthesis of this type of antenna array only. Generalization of these results to all genetic algorithm applications would be incorrect. As future work, similar analyses will be performed over several benchmark functions in order to deduce some generalization.



● ● ● ● ● ***mr_1* = 0.01 (Conventional GA)**

—··—··— *mr_{22}* = 0.01 – 3×10^{-5}×*itn*

—·—·—·— *mr_{33}* = 0.01 – 9×10^{-4}×*$itn^{1/2.1881}$*

——— *mr_{42}* = 0.01 – 1×$10^{-11.6}$×*itn^4*

- - - - - - - *mr_{22}* (Mixed-Type)

Figure 11. Population average fitness values of different mutation rate schemes (computed after 300 independent experiments).

Angular Positions	**ϕ_n (degree)**							
Desired	25	70	115	160	205	250	295	340
Obtained via mr_1 (Conventional GA)	25	63	115	159	205	250	295	340
Obtained via mr_{22}	25	63	116	159	206	250	295	340
Obtained via mr_{33}	26	63	110	160	208	257	295	339
Obtained via mr_{42}	**24**	**70**	**115**	**160**	**201**	**250**	**296**	**340**
Obtained via mr_{51}	26	69	115	159	205	256	295	340

Table 2. Angular positions obtained via GA with different mutation rate schemes.

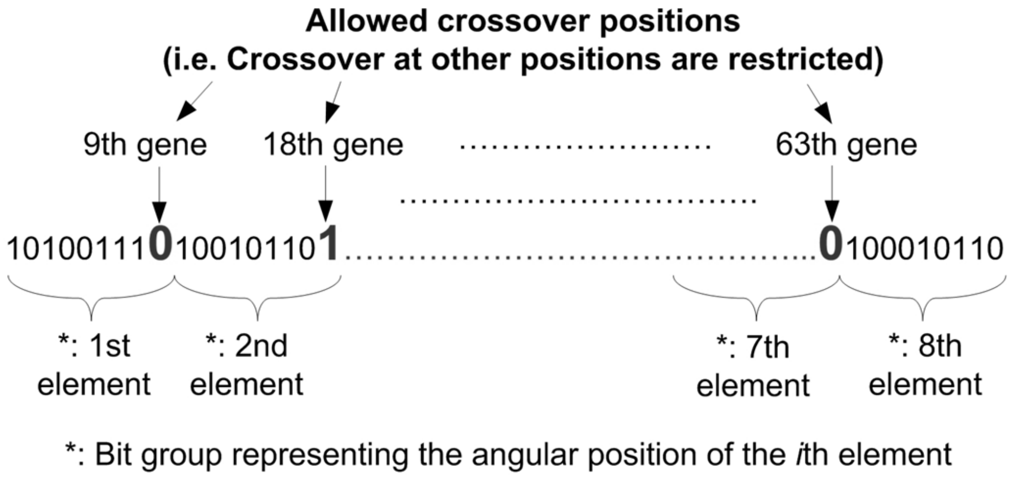


Figure 12. llustration of the restricted crossover point scheme.

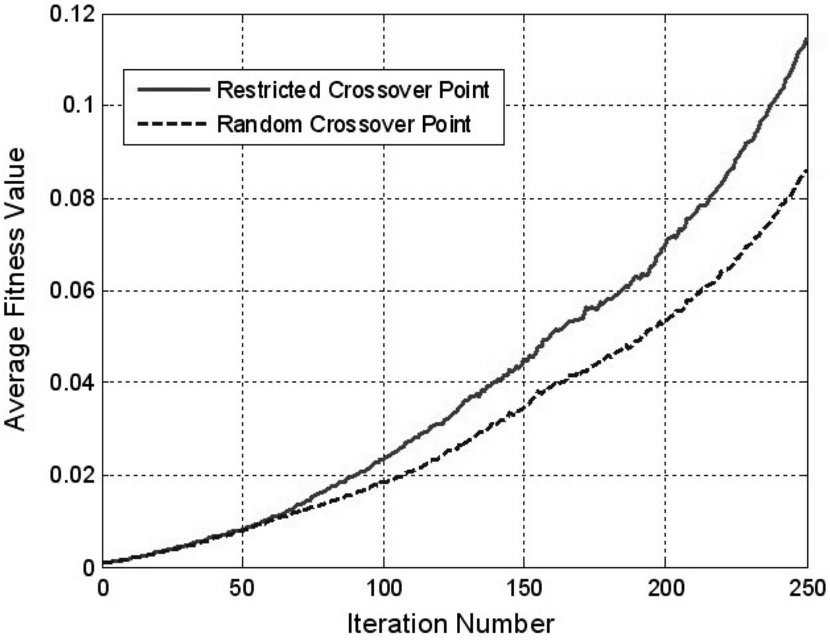


Figure 13. Population average fitness values obtained with restricted and random crossover point methods.

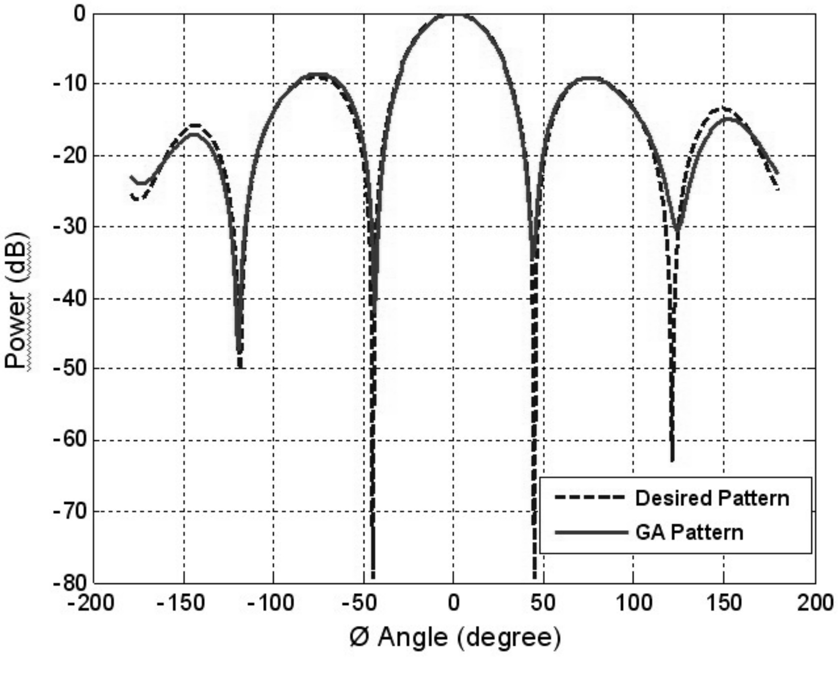


Figure 14. The comparison of desired and obtained radiation patterns.

References

[1] Holland, J., Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.

[2] Goldberg, D.E., Genetic algorithms in search optimization and machine learning, Addison–Wesley Longman, 1989.

[3] Michalewicz, Z., Genetic algorithms + data structures = Evolution programs, Springer-Verlag, Berlin, 1992.

[4] Whitley, L.D., Foundations of genetic algorithms-2, Morgan Kaufmann Publishers Inc., California, 1993.

[5] Balanis, C.A., Antenna theory analysis and design, John Wiley & Inderscience Publications, New Jersey, 2005.

[6] Visser, H.J., Array and phased array antenna basics, John Wiley & Sons Publications, 2005.

[7] Panduro, M.A., Mendez, A.L., Dominguez, R., Romero, G, Design of non-uniform circular antenna arrays for side lobe reduction using the method of genetic algorithms, AEÜ - International Journal of Electronics and Communications, vol. 60, pp. 713-717, 2006.

[8] Shihab, M., Najjar, Y., Dib, N., Khodier, M., Design of non uniform circular antenna arrays using particle swarm optimization, Journal of Electrical Engineering, vol. 59, no. 4, pp. 216-220, 2008.

[9] Xu, Z., Li, H., Liu, Q.-Z., Pattern synthesis of conformal antenna array by the hybrid genetic algorithm, Progress in Electromagnetics Research (PIER), vol. 79, pp. 75–90, 2008.

[10] Srinivas, M. and Patnaik, L.M., Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics, vol. 24, no. 4, pp. 656-667, April 1994.

[11] Stanhope, S.A. and Daida, J.M., An individually variable mutation rate strategy for genetic algorithms, Proceedings of the 6th International Conference on Evolutionary Programming VI (Lecture Notes in Computer Science), vol. 1213, pp. 235-246, 1997.

[12] Pham, D.T. and Karaboga, D., Genetic algorithms with variable mutation rates: Application to fuzzy logic controller design, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 211, no. 2, pp.157-167, 1997.

[13] Marsili Libelli, S., Alba, P., Adaptive mutation in genetic algorithms, Soft Computing, vol. 4, pp. 76-80, 2000.

[14] Thierens, D., Adaptive mutation rate control schemes in genetic algorithms, Proceedings of IEEE Congress on Evolutionary Computation (CEC ’02), pp. 980-985, 2002.

[15] Uyar, S., Eryigit, G., Sariel, S., An adaptive mutation scheme in genetic algorithms for fastening the convergence to the optimum, Proceedings of the 3rd Asia Pacific International Symposium on Information Technology, pp. 461-465, 2004.

[16] Soyturk, M., Evaluation of canonical algorithms and adaptive algorithms for two known problems, Journal of Naval Science and Engineering, vol. 5, no. 3, pp. 26-37, 2009.

[17] Lee, S. and Rowlands, H., Introducing the swingometer crossover and mutation operators for floating-point encoded genetic algorithms, Proceedings of the 19th European Conference on Modelling and Simulation (ECMS 2005), , 2005.

[18] Jabeen, S.D. and Bhunia, A.K., Real-coded genetic algorithm with variable rates of cross-over and mutation: a basis of global optimization for multi-modal functions via interval technique, International Journal of Computer Mathematics, vol. 83, no. 12, pp. 853-866, 2006.

[19] Lazarova, M., Borovska, P., Mabgar, S., The impact of the mutation strategy on the quality of solution of parallel genetic algorithms, Proceedings of the 9th WSEAS International Conference on Evolutionary Computing, pp. 63-68, 2008.

[20] Hinterding, R., Gielewski, H., Peachey, T.C., The nature of mutation in genetic algorithms, Proceedings of the 6th International Conference on Genetic Algorithms, pp 65-72, 1998.

[21] Piszcz, A. and Soule, T., Genetic programming: Analysis of optimal mutation rates in a problem with varying difficulty, International Journal of Innovative Computing and Applications, vol. 1, no. 2, pp. 108-120, 2006.

[22] Vrajitoru, D., Crossover improvement for the genetic algorithm in information retrieval, Information Processing & Management, vol. 34, no. 4, pp. 405-415, 1998.

[23] Shuhai, F., Two problems in genetic algorithm with chromosome differentiation, Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling (KAM ’09), pp. 248-250, 2009.

[24] Akbulut, M.B., Yilmaz, A.E., A modified genetic algorithm for the generalized assignment problems, Istanbul University – Journal of Electrical and Electronics Engineering (IU-JEEE), vol. 9, no.2, pp. 951-958, 2009.

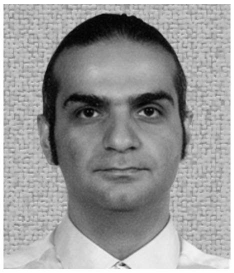
Acknowledgments

We would like to express our gratitude to the anonymous reviewers for their careful inspection of the paper and helpful comments for its improvement.

Authors´ Biographies

Fatih YAMAN

He studied electronics engineering at the Yıldız Technical University (Istanbul, TURKEY) and received his B.Sc. degree from the same university. He received his M.Sc. degree in electrical-electronics engineering from the Kırıkkale University (Kırıkkale, TURKEY) in 2005. His areas of interest are electromagnetics, antennas and heuristic methods such as genetic algorithm. He is currently working on communication systems and, at the same time, he teaches electromagnetics at undergraduate level. His Ph.D. education in electronics engineering at the Ankara University (Ankara, TURKEY) is still in progress.



Asım Egemen YILMAZ

He received his B.Sc. degrees in electrical-electronics Engineering and mathematics from the Middle East Technical University (Ankara, TURKEY) in 1997. He received his M.Sc. and Ph.D. degrees in electrical-electronics engineering from the same university in 2000 and 2007, respectively. Between 1996 and 2009, worked for the Turkish defense industry for different companies in different technical and administrative positions. He is currently with the Department of Electronics Engineering in Ankara University (Ankara, TURKEY), where he is an assistant professor. Also, he still provides technical consultancy to various institutions for various defense projects; he is also the Vice President of European Care Network Ltd. His research interests include computational methods, nature-inspired optimization and intelligent systems in addition to software development processes and methodologies. He has authored/ coauthored more than 70 peer reviewed technical papers. In credit of his scientific effort and contribution, he has been selected for Who's Who in Science and Engineering 2011-2012 (11th Edition)