Progress In Electromagnetics Research B, Vol. 20, 1–17, 2010

APPLICATION OF ARTIFICIAL IMMUNE SYSTEM ALGORITHM TO ELECTROMAGNETICS PROBLEMS

O. Kilic and Q. M. Nguyen

The Catholic University of America
Washington, DC 20064, USA

Abstract—This paper investigates the use of clonal selection principles based on our immune system for optimization applications in electromagnetics. This concept is based on our immune system's ability to respond to an antigen and produce a pool of anti-body secreting cells. In addition to the common implementations of this algorithm where the affinity maturation and cloning principles of clonal selection principles are used, we utilize memory and the cross-over concepts that are common to other bio-inspired methods. The performance of the algorithm is investigated for well known mathematical test functions and its potential is demonstrated in the context of the design of a radar absorbing material and a planar phased array antenna with specific radiation and null characteristics.

1. INTRODUCTION

Classical optimization techniques typically require an initial estimate reasonably close to the final result in order to avoid stagnation at a local optimum point. They also tend to be computationally intensive as they require analytical calculations such as derivatives. Nature provides heuristic optimization methods that rely on the techniques devised by different species over thousands of years for survival and can be utilized in engineering applications. Some of these optimizations are utilized for survival by different species. These approaches tend to be agent based as they can simultaneously sample the optimization space for a certain number of randomly inspired possibilities, with each iteration adding more intelligence to the heuristic search steps involved. Some examples of nature inspired optimization techniques are the genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], ant colony optimization (ACO) [3], and the artificial immune system (AIS) [4], which is the main focus of this paper.

Corresponding author: O. Kilic (kilic@cua.edu).

The GA is based on the evolution theory and survival of the fittest. In each iteration a new generation; i.e., solution set, is produced that is supposed to be better than the previous generation. In the case of PSO, the bee behavior in their search for the best location in a field is emulated. Bees make their decisions on where to go next as they search for the best location based on the collective intelligence of the swarm (global best point attained so far) and their personal experiences (personal best position achieved so far). The ACO is inspired by the ability of ants in finding the shortest path between their nest and food despite being blind animals. This is achieved by laying a chemical called pheromone on the path traveled. All these algorithms have been applied to electromagnetics problems before, especially GA [5] and PSO [6]. The application of ACO to electromagnetics problems has been relatively scarce. Their applications to the synthesis of linear arrays have been investigated by [7]. The performance of ACO and PSO were compared for linear array antenna optimization in [8], and a parallelized version has been implemented on FPGA platform in [9].

There are numerous papers investigating the performance of AIS [10, 11]. However, like ACO, the application of AIS to electromagnetics has also not been mainstream. In [12], a real coded clonal selection algorithm is proposed eliminating the need to convert the optimization space to the binary domain. The modified algorithm is then applied to a minimizing the stray field in superconducting magnetic energy devices. The clonal selection principles have also been applied to linear array synthesis problems in [13, 14]. These papers have implemented the AIS algorithm utilizing cloning and affinity maturation principles of clonal selection principles.

This paper investigates the performance of a modified AIS algorithm where the clonal selection principles are enhanced by memory and the cross-over concepts, where best solutions from past iterations are recalled (similar to PSO) and good solutions are crossed (similar to GA) to create a new antibody. The algorithm is tested with three test functions (e.g., Griewank, Rastrigin and Rosenbock) to demonstrate the robustness of the algorithm. The modifications to the algorithm are shown to enable a reliable convergence rate with few antibodies. The potential of the modified AIS algorithm for electromagnetics applications is demonstrated in the context of two applications: multilayer radar absorbing material design, and planar array antenna synthesis for reduced interference for multibeam satellite communications. In addition, a detailed analysis on how the definition of the cost function affects the convergence of the algorithm is provided. This cost versus convergence rate analysis is critical in understanding how heuristic search algorithms operate, and how their potential can

be realized with a good understanding of the application at hand.

The rest of the paper is organized as follows: A detailed description of the AIS algorithm is presented in Section 2. The performance of the algorithm is demonstrated for three test functions in 3. Electromagnetics applications of the algorithm is discussed in Section 4. Finally, concluding remarks are given in Section 5.

2. AIS ALGORITHM AND CLONAL SELECTION PRINCIPLES

AIS algorithm simulates human body's defense system against viruses [15]. Our adaptive immune system produces antibodies (Ab) whose aim is to bind to any foreign molecule, antigen (Ag), that can be recognized by immune system. In order to apply this to engineering problems we adopt the clonal selection principles, which involve (i) producing antibodies that recognize antigens, (ii) generating new random genetic modifications to antibodies via mutation. The best suitable antibody for the antigen is realized through cloning and mutation of the produced antibodies. For engineering applications of this biological process, Abs are used to represent a possible solution to the optimization problem. The optimization space is presented in binary form in order to emulate the gene behavior. Table 1 below summarizes the corresponding terminology in engineering applications to the biological terms involved.

Table 1. AIS terminology in engineering applications

AIS Terminology	**Engineering Terminology**
Antibody	Possible solution
Affinity (high)	Cost function (low)
Genes	Binary string representing a complete solution
Mutation	Random flipping of bits
Cloning	Duplication of binary strings
Crossover	Random mixing of parameters from a set of solutions

Since the biological process is based on gene representations, the optimization problem needs to be represented as a binary string for the AIS algorithm. The total length of the binary string for each antibody is the product of the number of parameters to be optimized and the number of bits, N_b used to represent them. Like in other heuristic search algorithms, the AIS algorithm starts with a random

sampling of the optimization space by assigning random values to all Abs. The available number of Abs, N_a, is an input parameter of the AIS algorithm. Next, the cost value, i.e., how close these values are to the desired solution, is computed for all Abs. The definition for the cost function is critical in the convergence rate of heuristic search methods, and will be discussed in more detail in Section 4.2. If one of the Abs satisfies the convergence criteria, i.e., the cost is within acceptable limits as set by the user, the algorithm terminates. If not, the iterative process continues.

2.1. Conventional AIS Algorithm

After the cost value is identified for each antibody, and if a desirable solution has not been realized, the process for the generation of a new population begins. This time instead of being purely random, intelligence is introduced to the search based on the information gathered so far by the Abs. First, the population is sorted in ascending order with respect to the cost, moving more desirable solutions to the top. Each Ab, Ab_i, is cloned at a rate, N_{ci}, that is inversely proportional to its cost value. The duplication is applied to the first $n\%$ of the antibodies, where n is a user defined parameter of the algorithm. In the conventional clonal selection algorithm, the original Ab and its clones form a sub-population. Cloning is followed by the mutation process, where each clone in a sub-population (not the original) undergoes a mutation process. The best of each subpopulation replaces the original Ab for the next generation, and the Abs that were not cloned are replaced by randomly selected new Abs. This results in $N_C + N_a$ calls to the function calculation per iteration, where $N_C = \sum_{i=1}^{n \times N_a} N_{ci}$ is the total number of clones and i is an index ranging from 1 to $n \times N_a$, with $i = 1$ corresponding to the best Ab.

2.2. Modified AIS Algorithm

The first step of the modified AIS algorithm is identical to the traditional approach where a random set of Abs are created within the optimization space. This step is demonstrated on the upper left corner in Fig. 1 for a two dimensional sinc function. For this particular example, where we search for the peak of the sinc function, the desired value is zero.

As in the traditional implementation of the AIS algorithm, the Abs are sorted and the top $n\%$ are cloned at a rate of $N_{ci} = round(\frac{\beta_c \times N_a}{i})$, where β_c is a user defined sensitivity parameter that adjusts the cloning

rate. At this stage, the number of Abs is identical to the conventional implementation; $N_C = \sum_{i=1}^{n \times N_a} N_{ci}$. The modified version does not use sub-populations, but applies mutation to the entire cloned set by randomly flipping some of the bits in each clone. The number of bits to be flipped is directly proportional to the cost value of each antibody, and is calculated by $round(\frac{i \times N_b \times \beta_m}{N_a})$, where β_m is a user defined sensitivity parameter that adjusts the mutation rate. The concept of cloning and mutation is demonstrated in Fig. 2. At the last stage of this process, the cloned and mutated Abs are sorted one more time, resulting in a solution set as depicted at the far right side in Fig. 2.

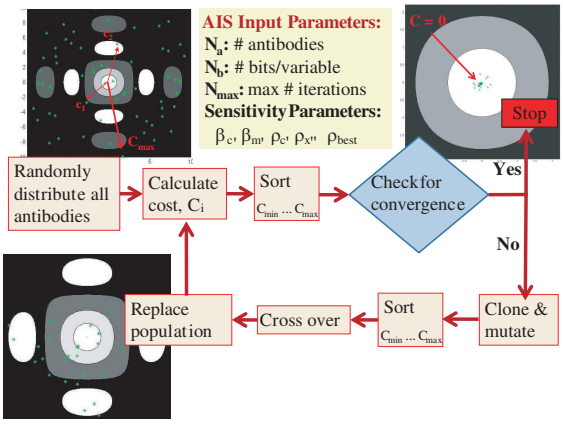


Figure 1. AIS block diagram.

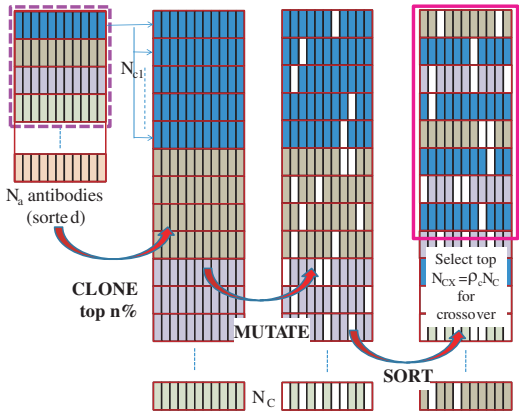


Figure 2. AIS antibody production.

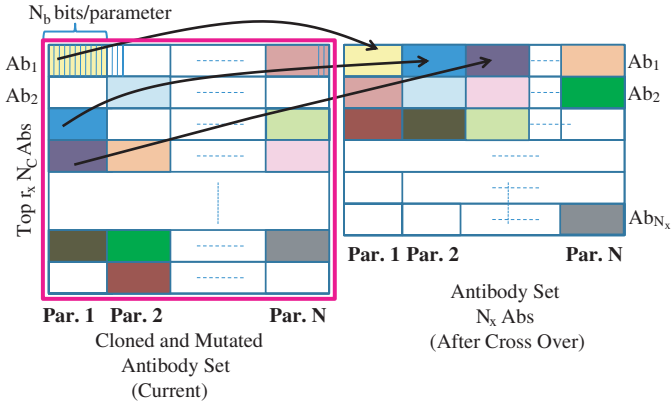


Figure 3. AIS crossover process.

Another modification is the use of cross-over, which is a process where a new set of antibodies is produced from the current set by randomly selecting the bit strings. This is carried out on a parameter basis; i.e., bit strings per parameter are selected randomly from the current set to create the new set of Abs as depicted in Fig. 2. The cross-over is limited to the top $\rho_c \times N_C$ Abs, where ρ_c is a user defined sensitivity parameter. The crossover process is carried out from this set to create a new set of N_X antibodies. The number of the new set of antibodies, N_X, is determined by the user via the parameter ρ_x such that $N_X = \rho_x \times N_C$. This process is depicted in Fig. 3. At this stage the new Ab set (N_X) created as a result of the crossover is combined with the original cloned and mutated set of Abs (N_C) resulting in an Ab population of $N_X + N_C$.

As a final step, the best Abs from the original set of antibodies (N_a) is added to this set, and the complete set is sorted based on the cost value. We have opted to use only the best solution from the original set in our implementation, resulting in a final set of N_X+N_C+1 Abs. The next iteration begins with the top $\rho_{best} \times N_a$ of these Abs, where ρ_{best} is another user defined sensitivity parameter. To complete the set to N_a antibodies as in the first step, the remaining $N_a-(\rho_{best} \times N_a)$ are created randomly to allow the algorithm to sample the optimization space randomly at each iteration.

3. AIS PERFORMANCE — MATHEMATICAL TEST FUNCTIONS

The introduction of cross-over and memory, as well as applying mutation to the entire cloned set and choosing the best from this

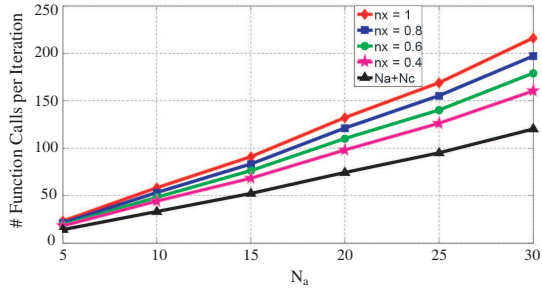


Figure 4. Rate of change in the number of function calls as a function of number of Abs and crossover size.

larger set, enables a more aggressive search of the optimization space per iteration. This is done at the expense of increased number of function calls to compute the cost. While the original AIS code would make $N_a + N_C$ calls to the cost function, the modification due to the crossover increases the calls per iteration to $N_a + N_C + N_X$. It is at the discretion of the user to limit the crossover population. For a given number of Abs, the number of calls per iteration depend on the cloning rate (i.e., β_c) and the size of the crossover population, N_X, which depends on the parameters n and ρ_x. Fig. 4 shows how the number of function calls per iteration changes as a function of N_a and ρ_x, where $\beta_c = 0.75$, and $n = 100\%$. The number of calls that would be made by the conventional approach is shown in black.

We provide a sensitivity analysis of the AIS algorithm to the user defined parameters, namely, N_a, β_c, β_m, ρ_x, ρ_c and ρ_{best}. Three functions (Griewank, Rastrigin and Rosenbrock) are considered for testing the AIS algorithm. The common features of all these functions is that they are multi-dimensional and their known minimum value, zero occurs at $x = 0$. We will be searching for the zero value with the AIS algorithm. Each of these functions has a specific characteristics that makes it suitable for testing the robustness of an optimization algorithm. For instance, the Rosenbrock function has a very wide range of values corresponding to a small range of input values; i.e., $1 \leq x \leq 10$ corresponds to $0 \leq f(x) \leq 10^5$. The Rastrigin function is highly oscillatory and varies between $0 \leq f(x) \leq 100$ when $-5 \leq x \leq 5$. In the case of Griewank function, we observe an oscillatory nature as well, however the range of the function values remain small; i.e., $0 \leq f(x) \leq 2$ when $-3 \leq x \leq 30$. The functions are described as follows:

Rosenbrock function

$$f(x) = \sum_{i=1}^{N} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right) \tag{1}$$

Rastrigin function

$$f(x) = \sum_{i=1}^{N} \left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right) \tag{2}$$

Griewank function

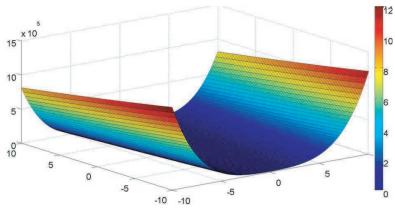
$$f(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{3}$$

These functions are plotted in Fig. 5 for $N = 2$ for ease of visualization. We will use $N = 5$ to test the algorithm in each case. In search of their minimum value, we will be using the following convergence criteria: $f_{Rosenbrock}(\vec{x}) \leq 10^{-4}$, $f_{Rastrigin}(\vec{x}) \leq \sqrt{0.02}$, and $f_{Griewank}(\vec{x}) \leq 10^{-3}$.

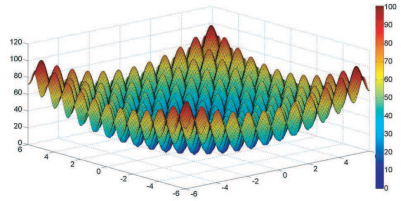
A summary of the performance of the AIS algorithm for these three functions is presented in Table 2 for the special case when $N_a = 15$, $n = 100\%$, $\beta_c = 0.5$, $\beta_m = 0.5$, $\rho_c = 0.75$, $\rho_x = 0.4$, and $\rho_{best} = 0.9$. The algorithm was run 100 times for each function to provide the statistics. It should be noted that the conventional AIS performs consistently worse than the modified version when applied to these functions. For instance, the conventional method failed to converge in any of the 500 simulations for the Griewank function when run with identical parameters.

Table 2. AIS performance for the test functions

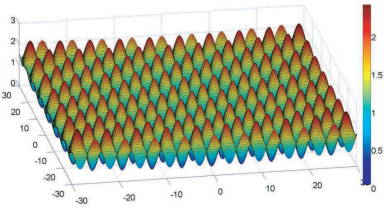
	Rosenbrock	**Rastrigin**	**Griewank**
% Conv	99	100	89
# Max Iter	500	205	500
# Min Iter	18	6	9
# Avg Iter	98	23	102
Best Cost	0	$2.3\ 10^{-5}$	$1.0\ 10^{-7}$
Worst Cost	0.88	0.14	0.09
Avg. # Func. Calls	5481	1288	5700



(a) Rosenbrock

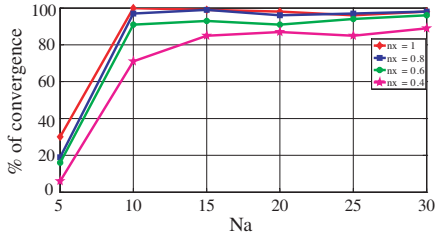


(b) Rastigrin

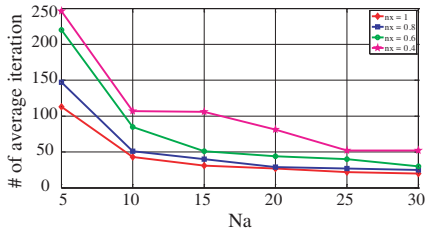


(c) Griewank

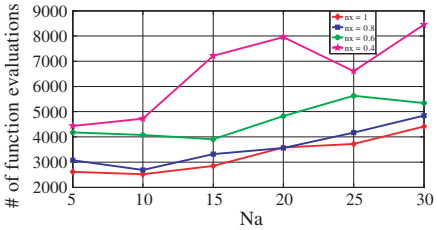
Figure 5. Test functions, plotted for $N = 2$.



(a) Convergence Rate



(b) Average # Iterations



(c) # Function Calls

Figure 6. Sensitivity analysis for rosenbrock function.

3.1. Sensitivity Analysis for the Number of Abs and Crossover Size

We demonstrate the effect of the number of Abs and the cross-over population size on the convergence rate of the algorithm in this section.

Fig. 6 shows this analysis as a function of N_a while ρ_x is varied between [0.4–1.0], and the other parameters are set as $\beta_c = 0.5$, $\beta_m = 0.5$, $n = 100\%$, $\rho_c = 0.75$ and $\rho_{best} = 0.9$. We only demonstrate the performance of Rosenbrock as the other functions follow a similar trend; i.e., as N_a increases the convergence rate increases and it takes the algorithm fewer iterations to converge. All test cases seem to respond well to ρ_x values lower than 1, meaning keeping the crossover of the few best Abs is sufficient. The main difference among the three cases is that on average Rastrigin function required the least number of function calls to converge, while Griewank required the most.

4. AIS PERFORMANCE — ELECTROMAGNETICS APPLICATIONS

The AIS algorithm is applied to two different electromagnetics problems: (i) the design of a radar absorbing material (RAM), (ii) null steering of a planar array for multiple beam satellite communications applications. In the RAM design the antibodies search for the right material type and thickness, whereas in the planar array case they represent the amplitudes of a phased array antenna. Since electromagnetics applications can be numerically intensive, we would like to show that the efficiency we have observed with the test functions prove AIS to be a viable tool in this field.

4.1. RAM Design

The RAM design consists of multiple layers of materials with different constitutive parameters and arbitrary thicknesses as depicted in Fig. 7. The reflection coefficient from a multiple layer of slabs can be computed recursively as follows: [16, 17].

$$\Gamma_i = \frac{\rho_i + \Gamma_{i-1} e^{-2j\delta_{i-1}}}{1 + \rho_i \Gamma_{i-1} e^{-2j\delta_{i-1}}} \text{ for } i = 2, 3, \ldots N+1 \tag{4}$$

where i is an index over the interfaces between slabs, and slab i lies between interfaces i and $i+1$. The total reflection from ith interface is denoted by Γ_i and ρ_i is the corresponding Fresnel reflection coefficient. The phase delay along the normal direction in each slab is denoted by $\delta_i = k_{z_i} d_i$.

For the design criteria of the RAM, normal incidence is assumed and the reflection coefficient is limited to a maximum value of $-30\,\text{dB}$ over the frequency range of $[0.1\text{–}1.0]\,\text{GHz}$. The RAM structure is limited up to five layers. The constitutive parameters in each slab is selected among the 16 different materials as provided in [16]. The

unknowns of the optimization problem are the material type and the thickness of each slab. Since we are limited to 16 different material types, four bits were allocated to the selection of material, and 10 bits for the slab thickness, resulting in 14 bits to define each slab. Since there are five slabs, 70 bits represent a possible solution (i.e., Ab). The total number of Abs in the simulation was chosen as $N_a = 40$, and the maximum iteration number was set at 200. The sensitivity parameters were set at $\rho_x = \rho_{best} = 1.0$, $\rho_c = 0.75$ and $\beta_c = \beta_m = 0.5$. The optimized reflection coefficient as a function of frequency is plotted in Fig. 8. It is observed that the design criteria of -30 dB is achieved across the band. The material thicknesses were 1.9863, 1.9883, 1.4878, 0.8485, and 0.7742 mm and material types (following the list in [16]) were #4, #4, #7, #16, #11, for $i = 1, 2, \ldots, 5$, respectively.

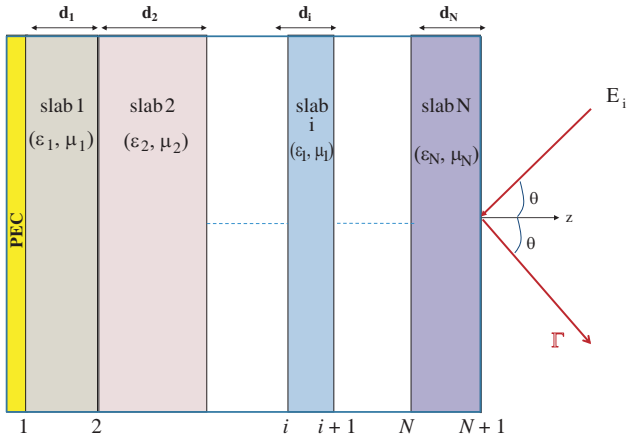


Figure 7. Multilayered RAM structure.

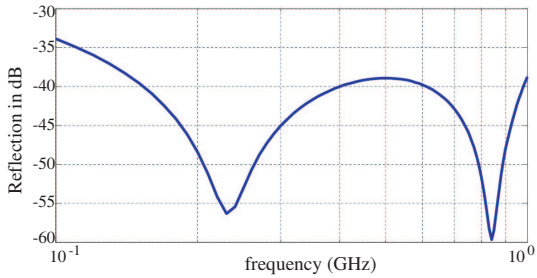


Figure 8. Optimized performance of the RAM structure.

4.2. Cost Value Calculation and Its Effects on Performance

The performance of heuristic search algorithms rely heavily upon how the cost function is determined. The convergence rate is not only affected by the process of generating intelligent solutions based on the sampling of the optimization space, but how the calculated values at each position relate to a "cost" value. The cost is the key in the creation of the next generation of the solution set, yet its calculation is typically arbitrary in random search methods. It is up to the user to devise an effective function for the application at hand.

We investigated three different cost functions for the RAM design, as depicted in Fig. 9. The first case has an increasing slope as the deviation from the desired solution increases. The second case provides a high slope in the vicinity of the desired solution. Finally, the third case has a linear slope as we deviate from the desired value. We observe that for the given application and sensitivity parameters used, Case 2 performs better in terms of the convergence rate. For this particular case, the algorithm does a good job bringing all Abs close to the desired range rapidly. The steeper slope of Case 2 in the vicinity of the desired solution enables a faster convergence compared to the other cases. It should be noted that the convergence criteria was strict and no values higher than -30 dB were deemed acceptable no matter how close.

The modified AIS algorithm we present in this paper performs better in the RAM design as well. The conventional approach converged 77% of the time for Case 2 when run with identical parameters. The average number of iterations for the 100 simulations was 222, with the best value of reflection coefficient achieved being -33.38. In order to achieve the identical number of function calls per iteration to the modified approach, N_a was set at a higher value (51) for the conventional case.

Table 3. Cost function effects on AIS performance — RAM design

100 Simulations	**Case 1**	**Case 2**	**Case 3**
Convergence Rate (%)	74	98	88
Best # iter	9	8	7
Worst # iter	500	500	500
Average # iter	228	117	145
Min. total thickness(mm)	6.07	5.43	5.73
Max. total thickness (mm)	9.43	8.75	8.90
Best reflection (dB)	-34.01	-34.39	-34.18

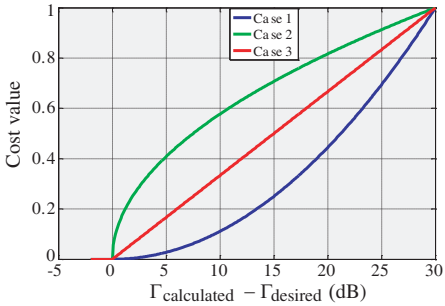


Figure 9. Cost analysis for performance of RAM optimization.

4.3. Planar Array Design for Multibeam Satellite Communications

Satellite communication systems have limited bandwidth for the large areas they typically serve. Therefore, they reuse the available bandwidth in regions that are separated sufficiently apart from each other. This is known as frequency reuse and the beams that utilize the same bandwidth are referred to as co-channel beams [18]. Fig. 10 shows the multiple beams generated by a satellite antenna on earth. The right side of the figure demonstrates an example of frequency reuse factor of 7; i.e., the same channel is repeated after every 7 beams. Co-channel beams, denoted by identical colors, interfere with each other despite their separation due to their shear number in the large coverage area. To demonstrate the co-channel interference concept, the beam coverage on earth (white circles) is overlaid on the contour plot of a planar array antenna that creates the center beam on earth at the left bottom corner of the figure.

In an hexagonal geometry as shown in Fig. 10, co-channel beams reside on circles, also known as tiers. These tiers are identified by the large black circles in the figure. The location of the co-channel beams on each tier is identified by the black dots at their centers. One can observe six co-channel beams in each tier for the first three tiers. The strength of radiation from the center beam on to these co-channel beams can be seen from the antenna radiation pattern shown in the background. The overall interference can be reduced by synthesizing the array antenna so that radiation into these co-channel beams is reduced, [19]. Our objective in this example is to reduce the side lobe levels ($\leq -30\,\mathrm{dB}$) within the co-channel beams in the first tier by an amplitude control. The phase distribution is assumed uniform.

We consider a 26×26 planar array with elements separated at half wavelength on a square grid. The satellite is located at 9400 km

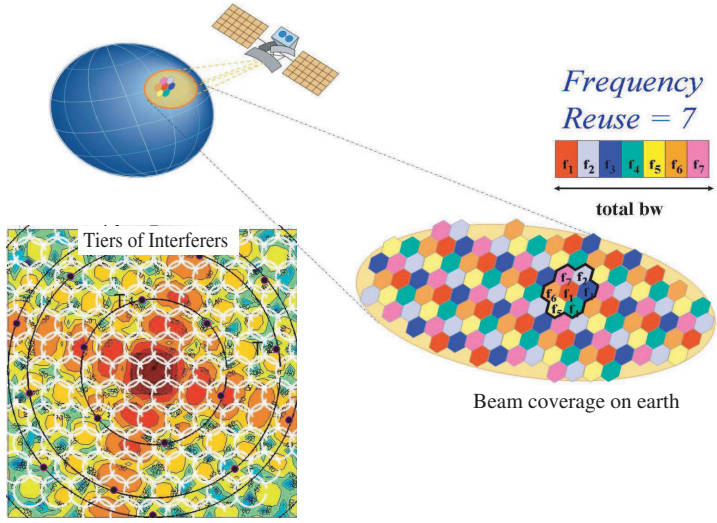


Figure 10. Co-channel beam interference in multibeam satellite communications.

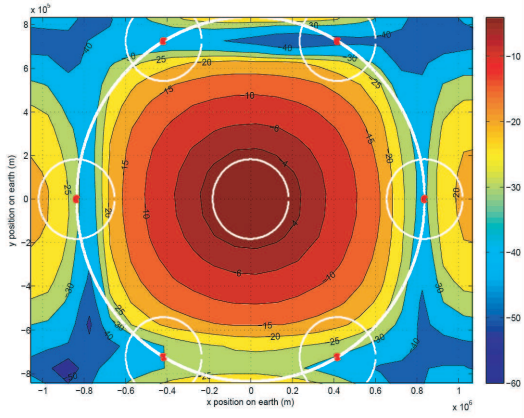


Figure 11. Optimized coverage for 26×26 element array.

altitude and operates at 1.9 GHz. The centers of the six co-channel beams at the first tier are located at 5.02 degrees off the center beam. We use the AIS algorithm to reduce the radiation from the center beam into the center of the first tier co-channel beams. The parameters of the algorithm are set as follows: $N_a = 80$, $n = 100\%$, $\beta_c = 1.0$; $\beta_m = 0.5$, $\rho_c = .75$, $\rho_x = .75$, and $\rho_{best} = 0.85$. Four bits are used to

represent amplitude values ranging between [0–1.0] for each element. This results in 2704 bits per Ab. The optimized pattern on earth is plotted in Fig. 11 demonstrating that the radiation from the center beam into center of the six co-channel beams have been reduced to −30 dB or below. The algorithm on average takes 80 iterations to converge.

5. CONCLUSION

The use of clonal selection principles of AIS algorithm was investigated for electromagnetics applications in the context of a multilayered RAM design, and null positioning of a planar array antenna. The conventional AIS algorithm has been modified to include effects such as cross-over and memory as inspired by other nature based heuristic search algorithms such as the genetic algorithm and particle swarm optimization. As a result more sensitivity parameters than has been reported in earlier work has been introduced. The robustness of the algorithm was tested using the Rosenbrock, Rastrigin and Griewank functions. A sensitivity analysis was carried out for the different parameters of the algorithm. It has been demonstrated that the convergence rate of the algorithm is not only dependent on these sensitivity parameters, but also the definition of the definition of the cost function. The algorithm was successfully used in the design of a multilayered RAM over a broad range of frequencies. As a second application, a satellite array antenna pattern was optimized to reduce interference in co-channel beams of a multiple beam coverage on earth. The algorithm is very robust and presents itself as a suitable tool for challenging electromagnetics problems.

REFERENCES

1. Holland, J. H., "Genetic algorithms," *Scientific American*, 66–72, Jul. 1992.
2. Kennedy, J. and R. C. Eberhart, "Particle swarm optimization," *Proc. IEEE Conf. Neural Networks IV*, 1995.
3. Dorigo, M., V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Systems, Man, and Cybernetics*, Part B, Vol. 26, No. 1, 1–13, 1996.
4. Mori, K., M. Tsukiyama, and T. Fukuda, "Immune algorithm with searching diversity and its application to resource allocation problem," *TIEE Japan*, Vol. 113-C, No. 10, 872–878, 1993.

5. Haupt, R., "An introduction to genetic algorithms for electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 37, No. 2, 7–15, Apr. 1995.
6. Robinson, J. and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Trans. Antennas and Prop.*, Vol. 52, No. 2, 397–407, 2004.
7. Karaboga, N., K. Guney, and A. Akdagli, "Null steering of linear antenna arrays with use of modified touring ant colony optimization algorithm," *Int. Journal of RF and Microwave Computer-Aided Engineering*, Vol. 12, No. 4, 375–383, 2002.
8. Kilic, O., "Comparison of nature based optimization methods for multi-beam satellite antennas," *Proc. Applied Computational Electromagnetics Conf.*, 2008.
9. Kilic, O., "FPGA accelerated phased array design using the ant colony optimization," *Applied Comp. Electromag. Soc. Journal*, 7, Feb. 2010.
10. De Castro, L. N. and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evolutionary Computation*, Vol. 6, 239–251, 2002.
11. Chun, J., H. Jung, and S. Hahn, "A study on comparison of optimization performances between immune algorithm and other heuristic algorithms," *IEEE Trans. on Magnetics*, Vol. 34, No. 5, 2972–2975, 1998.
12. Campelo, F., F. G. Guimaraes, H. Igarashi, and J. A. Ramirez, "A clonal selection algorithm for optimization in electromagnetics," *IEEE Trans. Magnetics*, Vol. 41, 1736–1739, 2005.
13. Akdagli, A., K. Guney, and B. Babayigit, "Clonal selection algorithm for design of reconfigurable antenna array with discrete phase shifters," *Journal of Electromagnetic Waves and Applications*, Vol. 21, No. 2, 215–227, 2007.
14. Babayigit, B., A. Akdagli, and K. Guney, "A clonal selection algorithm for null synthesizing of antenna arrays by amplitude control," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 8, 1007–1020, 2006.
15. Ada, G. L. and G. Nossal, "The clonal selection theory," *Scientific American*, Vol. 257, 50–57, 1987.
16. Michielssen, E., J.-M. Sajer, S. Ranjithan, and R. Mittra, "Design of lightweight, broad-band microwave absorbers using genetic algorithms," *IEEE Trans. on Microwave Theory and Tech.*, Vol. 41, No. 6/7, 1024–1031, 1993.
17. Chambers, B. and A. Tennant, "Optimised design of Jaumann

radar absorbing materials using a genetic algorithm," *IEE Proceedings Radar, Sonar and Navigation*, Vol. 143, No 1, 23–30, 1996.

18. Macdonald, V. H., "The cellular concept," *The Bell System Technical Journal*, Vol. 58, No. 1, 15–41, Jan. 1979.
19. Kilic, O. and A. I. Zaghloul, "Antenna aperture size reduction using sub-beam concept in multiple-spot-beam cellular satellite systems," *Radio Science*, Vol. 44, RS3001, May 2009.