Modified cGA for Electromagnetic Optimization

***Bui Van Ha*[1], *F. Grimaccia*[1], *M. Mussetta*[1], *P. Pirinoli*[2], *R.E. Zich*[*1]**

[1]Politecnico di Milano, Dipartimento di Energia
Via La Masa 34, 20156 Milano, Italy
ha.bui@mail.polimi.it, riccardo.zich@polimi.it

[2]Politecnico di Torino, Dipartimento di Elettronica
Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract

Compact Genetic Algorithm (cGA), which uses a probability vector (PV) to represent the population, has been proposed as an alternative of the simple Genetic algorithm (sGA), which greatly reduces the memory storage requiring during its performance. The cGA, however, just performed equivalently to sGA. In this paper, a modified version of compact Genetic Algorithm (M-cGA), outperforming the standard cGA, is presented. The idea is to use more than one probability vector and add a suitable learning scheme to improve the cGA's capability. Numerical results of the application of M-cGA on high-order problem, i.e. four-bit problem, and electromagnetic optimization, i.e. thinned array synthesis, will be presented and compared with the results obtained by its ancestors and GA as well.

1. Introduction

Genetic Algorithms (GAs) [1] are a class of the most well-known optimization algorithms for applied engineering thanks to their ease of implementation. Their concepts are based on the idea that the next generation would perform better than the current population. The creation of next generation depends on two main operators: selection and mutation. Depending on the selection methods, there are two general approaches to GAs: Generational update and steady-state update, as depicted in Fig. 1. Generational approach selects a large number of individuals and makes a "big" update of population, while steady-state approach only selects two individuals and makes "small" update to the population.

Both of these approaches have their own advantages and drawbacks, e.g. generational scheme evaluates more individuals per one iteration, but requires less iterations than steady-state scheme. The generational scheme seems to gain popularity since most of current GA versions are applying this concept. The steady-state scheme, however, reduces computer resources needed to store and evaluate the population of GAs since every iteration the algorithm just needs to store and evaluate only two new generated individuals.

In [2], Harik implemented this idea and created a new algorithm, named compact Genetic Algorithm – cGA. The cGA uses a probability vector (PV) to represent the population, therefore it greatly reduces the memory storage required by the algorithm. The cGA's performances, however, are generally equivalent to the simple GA (sGA) with uniform crossover [2].

In [3], the cGA has been modified to improved its performance by adding elitism mechanism. Ahn proposed two modified versions of cGA: persistent elitist compact genetic algorithm (pe-cGA) and nonpersitent elitist compact genetic algorithm (ne-cGA). The elitism-based cGAs have shown theirs better capability in speeding up the convergence and

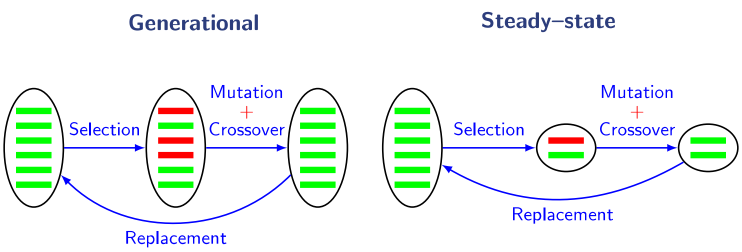

Figure 1: Generational update vs. Steady-state update.

978-1-4673-5225-3/14/$31.00 ©2014 IEEE

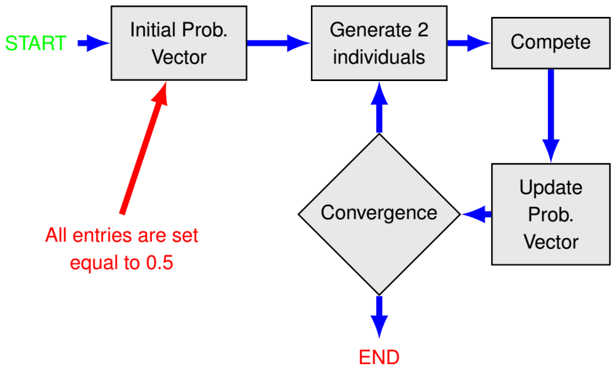

Figure 2: Flowchart of the cGA.

reducing the computational cost. However, these modifications couldn't improve the performance of cGA in term of solution quality, i.e. obtained better solutions.

In [4], the cGA has been further improved by implementing more than one probability vector to represent the population and introducing a learning mechanism in updating process (denoted as M-cGA hereafter). The M-cGA has been tested through mathematical and electromagnetic problems to prove its performance. In this paper, the behavior of M-cGA will be studied with higher order problem, i.e. 4-bit problem, and its application to Electromagnetic problem, i.e. synthesis of thinned array.

2. Compact Genetic Algorithm

Compact Genetic Algorithm cGA, first presented in [2], uses probability vector (PV) to represent a possible solution. In cGA, instead of using real population as in traditional Genetic Algorithm, it manages PV to get the distribution of good solutions. The length of PV corresponds to the number of variables of problem, and the value of PV measures the probability of one variable to get particular value e.g. the proportion of "1" in case of binary problem. A full treatment of the method can be found in [2, 3], but for the sake of clarity and uniformity of notation it is briefly summarized in the following section.

The flowchart of the cGA is described in Fig. 2. Initially, the PV is assigned a value of 0.5, i.e. assuming uniform distribution for every position. In each generation, two individuals are generated from current PV. They have to compete, and the winner will be responsible for updating the PV. The updating rule will increase or decrease probability vector by a factor $1/n$ (n-population size) according to the value of the winner. The cGA will stop when PV has value of 0 or 1 at all positions, i.e. finding optimal solution.

3. Improved Compact Genetic Algorithm

The cGA works well when problems consist of non-overlapping Building Blocks (BBs), i.e. low order behavior, however it shows limitations when dealing with higher order BBs. To overcome this drawback, Harik [2] introduced a modification of the cGA by increasing the number of generated offspring and applying tournament competition, i.e. simulating higher selection pressure. This allows the cGA to solve the problem with higher order BBs. However, the modification has the drawback of the increase of the computational cost since it needs to store and evaluate a considerable number of individuals.

In [3], Ahn proposed new versions of cGA introducing elitism. He created two different mechanisms: persistent elitism cGA (pe-cGA) and non-persistent elitism cGA (ne-cGA). The elitism-based cGAs outperform the original cGA in term of function evaluations. The reason for this is that the elitism can prevent the loss of low salience genes of chromosomes, which is equivalent to increase the selection pressure. Unfortunately, pe-cGA and ne-cGA could not perform better in term of solution quality, while they have only the advantage of consuming less memory than all cGAs.

In all the above mentioned modifications, the authors attempted to manipulate only one probability vector in different ways in order to speed up the convergence of the algorithm. However, their performances decrease when the order of complexity of the problem increases. In our work, the cGA has been further improved by simultaneously dealing with more probability vectors. The idea is that using more than one PV enhances the exploration properties of the algorithm,

Figure 3: The updating of PVs in M-cGA.

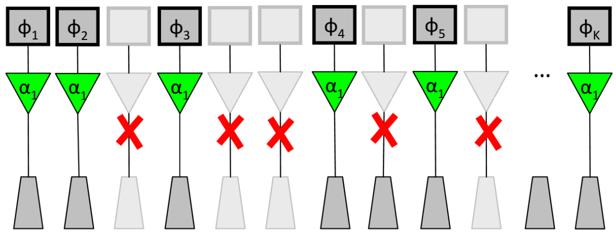

Figure 4: Thinned array from a regular array.

and this will increase the ability to avoid local trapping dealing with high-order BBs. Moreover, the learning mechanism of the information from different PVs may also enhance the exploitation itself of the algorithm speeding up the convergence.

The updating step in M-cGA is, therefore, needed to be modified to effectively control these PVs, as shown in Fig. 3. The PVs are involved in two updating processes: local update, as the original cGA, and global update, thanks to which each PV learn from the best PV. The preliminary results [4] have shown that the modified version of ne-cGA outperforms all the previous considered versions of cGA and GA as well. In the next section, the performance of M-cGA is studied through different optimization problems, and the numerical results will show the M-cGA's effectiveness.

4. Numerical Results

In this section, the numerical results obtained by the proposed algorithm are reported. The results are compared with the one obtained on the same problem with different cGAs ancestors both in terms of the obtained solution quality and in terms of the required numerical effort. In order to consider a reasonable statistics over the intrinsic stochastic of these approaches, all the numerical results have been averaged over 50 runs. The M-cGA implements 4 probability vectors, while the length of inheritance of non-persistent elitism is equal to 10% of population size.

In Electromagnetic, the synthesis of thinned array, i.e. involving the procedure of removing some radiating elements from a regular periodic array as shown in Fig. 4, can be considered as a binary optimization problem. The excitation of each element can be assigned a value of 1/0 corresponding to the state "ON/OFF" of that element in the array. From the view of the performance in the previous section, the M-cGA seems to be a good candidate for the design of thinned array. In this section, the application of the M-cGA is examined and presented.

The considered problem is the 200-element linear array with 23% of elements turned off. The array is chosen to compared with available literature results on the same problem. The objective is to minimize the peak side-lobe levels (PSL) of the array.

Several methods have been proposed to effectively control the PSL in the thinned array: the application of stochastic approach, e.g. Genetic Algorithm [5], the deterministic approaches such as Almost Different Set (ADS) [6], the iterative Fourier transform (IFT) [7], and the hybrid approach based ADS-GA combining the advantage of stochastic and deterministic approaches [8]. Table 1 reports the optimized PSLs by different methods. The results show that the M-cGA outperforms all the other algorithm including GA or its hybridization. Finally, the pattern of the optimized array is shown in Fig. 5, showing all the side-lobes are well-controlled by M-cGA.

Table 1: Obtained best PSLs

PSLs (dB)			
GA [5]	**IFT** [7]	**GA-ADS** [8]	**M-cGA**
-22.09	-22.85	-23.05	-23.75

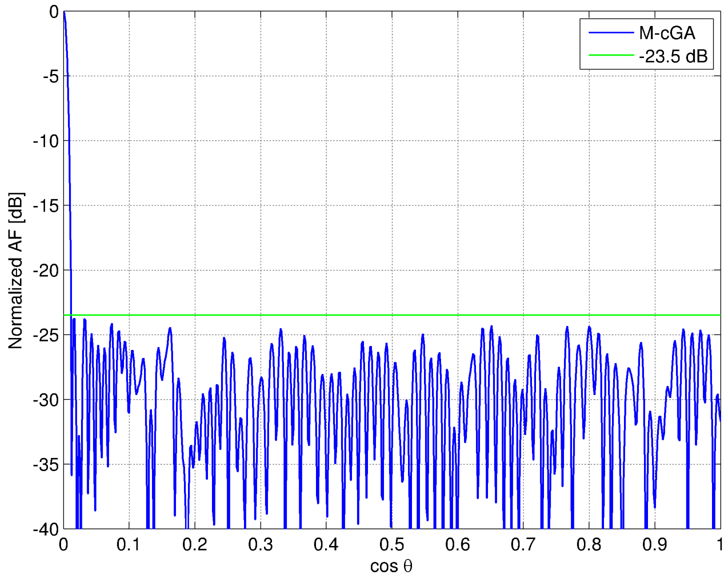

Figure 5: Normalized array factor of 200-element thinned array obtained by M-cGA.

5. Conclusions

In this paper, an improved version of the cGA has been presented and studied through high-order BBs and thinned array problem. The numerical results have shown the superior performance of M-cGA with respect to its ancestor and the GA as well. The results also indicate a promising implementation of M-cGA to optimize several complex electromagnetic designs.

References

[1] D. E. Goldberg, "Genetic Algorithms in search, optimization and machine learning," *Addison – Wesley*, 1989.

[2] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The Compact Genetic Algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 287-297, Nov. 1999.

[3] C. W. Ahn and R. S. Ramakrishna, "Elitism-based Compact Genetic Algorithm," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 367-385, Aug. 2003.

[4] B. V. Ha, R. E. Zich, M. Mussetta, P. Pirinoli, and D. N. Chien, "Improved compact genetic algorithm for EM complex system design," *Proc. ICCE*, Hue, Vietnam, Aug. 2012, pp. 389-392.

[5] Randy L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 42, no. 7, Jul. 1994.

[6] G. Oliveri, L. Manica, and A. Massa, "ADS-based guidelines for thinned plnar arrays," *IEEE Trans. Antennas Propag.*, vol. 58, no. 6, pp. 1935-1948, 2010.

[7] Will P. M. N. Keizer, "Linear array thinning using iterative FFT techniques," *IEEE Trans. Antennas Propag.*, vo. 56, no. 8, pp. 2757-2760, Aug. 2008.

[8] G. Oliveri and A. Massa, "Genetic Algorithm enhanced almost difference set based approach for array thinning," *IET Microwave, Antennas Propag.*, vol. 5, pp. 305-315, 2011.