GAs with PDSS and Adaptive Parameters for Phased Array Synthesis

Sunday C. Ekpo[1], Edidiong-Obong U. Ekpo[2], and Armstrong A. Sunday[3]

[1]Don Manuel & Associates (Consulting Engineers)
7 Gibbs Street, P. O. Box 1838, Uyo 520001, AIS, Nigeria.
cookey_sunday@yahoo.com, scekpo@iee.org

[2]Don Manuel & Associates (Consulting Engineers)
7 Gibbs Street, P. O. Box 1838, Uyo 520001, AIS, Nigeria.
edidiongobongjc@yahoo.com

[3]Don Manuel & Associates (Consulting Engineers)
7 Gibbs Street, P. O. Box 1838, Uyo 520001, AIS, Nigeria.
armstrong_nokey@engineer.com

Abstract: Genetic algorithms (GAs) are search methods based on the principles and concepts of natural selection and evolution. They are commonly used to solve many optimisation and synthesis problems. An important issue facing the user is the selection of genetic algorithm parameters, such as selection strategy, mutation rate, mutation range, and number of crossovers. This paper demonstrates population decimation selection strategy (PDSS) that ensures proper convergence with adaptive parameters during the optimisation process, which is shown to outperform its best counterpart when used to synthesise phased array weights to satisfy a specified far field sidelobe envelope. When compared to other selection strategies implementations, the algorithm converges faster with the best solutions resulting in a great reduction of computation time.

Keywords: convergence, far field, GA parameters, PDSS, phased array

1. Introduction

1.1 Phased Array Notch Synthesis

The optimisation problem considered here is the synthesis of phased array weights to meet a far field sidelobe requirement including a 60dB notch on one side. The antenna is a linear phased array of one hundred half-wavelength spaced radiators. The cost measure to be minimised is the sum of the squares of the excess far field magnitude above the specified sidelobe envelope. This penalizes sidelobe above the envelope, while neither penalty nor reward is given for sidelobes below the specification. The lower the cost, the more fit the array distribution. [1]

1.2 Genetic Algorithm

The flow chart of a genetic algorithm [1] is shown in **Fig. 1**. Using processes analogous to genetic recombination and mutation from biology, an initial population of 100 random candidate array distributions is generated and evolved for a specified number of generations or iterations.

Selection strategies determine which chromosomes will take part in the evolution process (will be involved in the making of the next generation in terms of mating with other chromosomes, or simply being inserted unchanged). [2] There are three major selection strategies, viz: population decimation selection, proportionate selection and tournament selection. [2] The population decimation selection strategy PDSS [2] and tournament selection strategy TSS [1] are utilised in the optimisation problem. The final array distribution is taken as the most fit individual after the specified number of iterations.

Crossover refers to the mixing of information from both parents to create the children. The crossover point determines the point at which a chromosome will be dissected. Allowable input values for crossover point [2] is 0≤crossover point≤1 where 0 indicates a crossover point that is randomly chosen, and 1 indicates that no crossover will occur. For example, a crossover point of 0.5 would indicate that the chromosomes would be cut in half. [2] Two crossovers implies that the first child will receive both left and right sides of the array weights from the first parent and the middle section from the second parent, where the two crossover points are randomly chosen. Positive noninteger number gives the probability of rounding the number of crossovers up, versus rounding it down. [1] So if the number of crossovers is 2.25, then each pair of parents has a 25% chance of using three crossovers and a 75% chance of using two crossovers. For an integer number of crossovers this reduces to the conventional crossover definition. In this paper, there are seventeen possible number of crossovers: {0, .25, .5, …, 4}.[1]

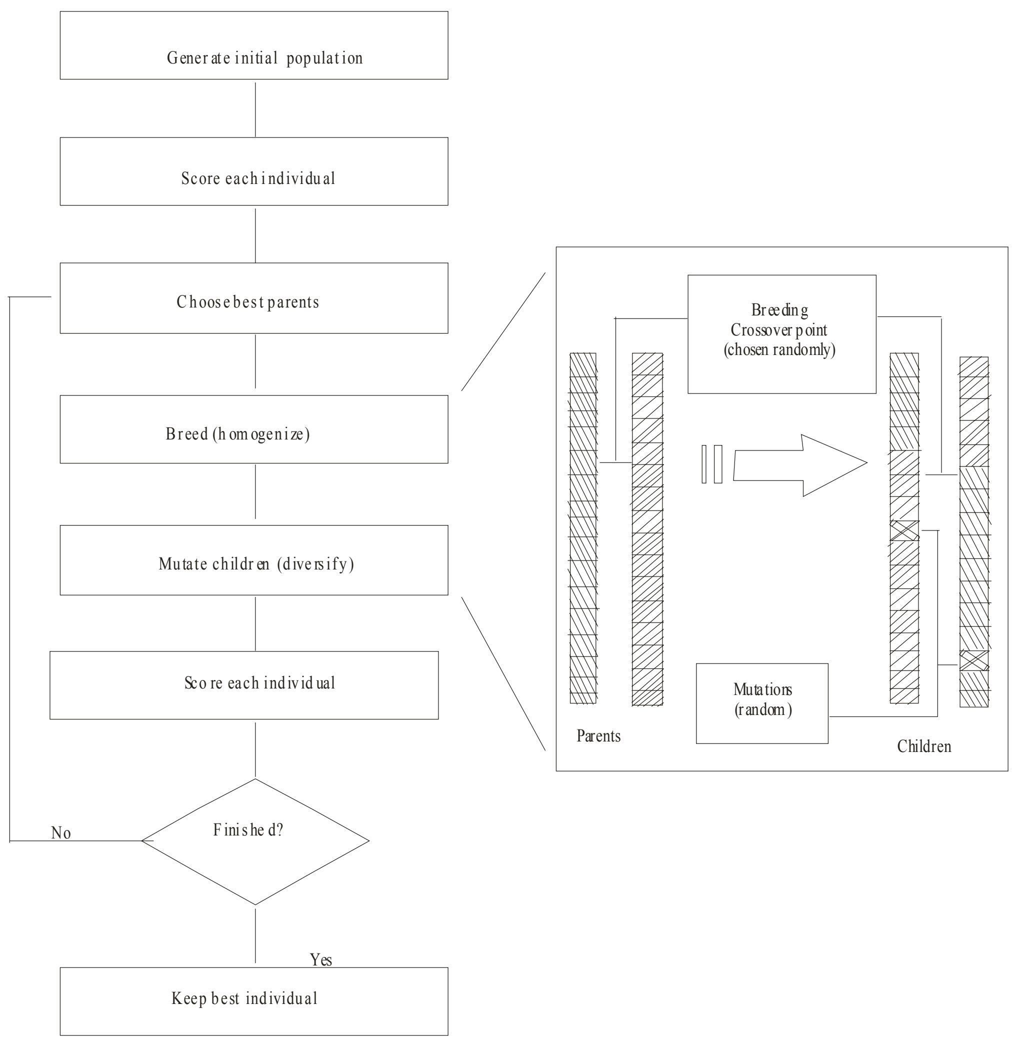

Fig. 1. Flowchart Illustrating the steps of a genetic algorithm
Breakout: Illustration of Breeding and Mutation

Next, mutations [1] are applied to the children, which means replacing a radiating element's excitation current with a randomly chosen value. The probability, $P_{mutation,}$ [2] that a particular element is mutated is governed by the mutation rate. Here ten possible mutation rates are used: {2%, 4%, …, 20%}. [1] The mutation range [1] governs how far a mutated element weight may be from its original (pre-mutation) value. For the mutation range r ($0 \le r \le 1$), the mutated value is chosen within the fraction r of the allowed range from the original value x with uniform probability. [1]

2. PDSS and TSS with Adaptive Parameters

Consider now, the automatic adaptation of the genetic algorithm parameters with PDSS and TSS selection strategies. A dynamic parameter adjustment strategy updates specified parameters to maximise relative improvement. [1] The algorithm therefore follows a descent procedure, adjusting each parameter in the direction that gives the most cost function improvement. Since each parameter is tested simultaneously over multiple iterations, some averaging of results occurs which helps to mitigate the inherent noisiness of the cost function improvements. [1] [2]

In this paper, the efficiency of PDSS over TSS is illustrated using a 6-element Yagi Udah [2] operating at 300MHz and optimised with the requirements set as follows: [2]

- All the director elements are of the same length.
- The spacing between all elements is the same except between reflector and feed.
- The radii of all elements are 0.008m
- Element lengths are allowed to vary from 0.35 – 0.65m
- Element spacing is allowed to vary from 0.1 - 0.25m
- All antenna parameters vary with a 1mm resolution
- The reflector is set to a length of 0.481m
- The input impedance of the system is 75 ohms

The array is optimised twice for 80 generations (**Fig. 2**).

3. RESULTS

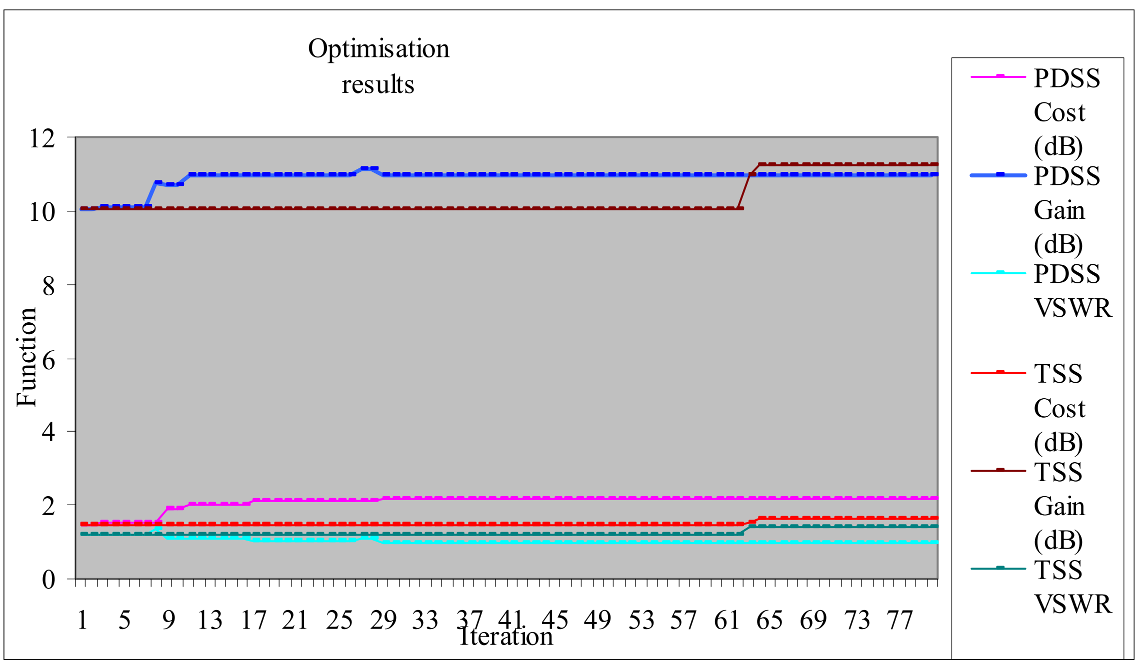

Fig. 2. PDSS and TSS Optimisation Function curves for a Yagi Udah

The results distinctly show the greater convergence efficiency and robustness of PDSS over TSS in optimisation and synthesis problems. **Fig. 2** shows the performance of each selection strategy in 80 iterations. The best configuration obtained in each iteration is plotted.

The results show PDSS fully converging at the 29th iteration and TSS, 64th (**Table 1**).

Table 1: PDSS and TSS Optimisation results for a Yagi Udah Array

Iteration	PDSS			TSS		
	Cost (dB)	Gain (dB)	VSWR	Cost (dB)	Gain (dB)	VSWR
29	2.1938	10.9800	1.0029	1.4681	10.0600	1.1962
64	2.1938	10.9800	1.0029	1.6468	11.2200	1.4027

The dynamic parameter adjustment technique with TSS is now applied to the problem of synthesising a far field sidelobe notch for phased array. [1] The optimised far field pattern converges from completely random numbers to a distribution that completely meets the sidelobe specifications everywhere in 112 iterations. [1] But with the PDSS, the sidelobe specifications are met in just 51 iterations.

4. CONCLUSION

The population decimation selection strategy has been shown to meet assembly optimisation and synthesis constraints and specifications with the best solutions that outperform its closest counterpart.

The proposed PDSS, with adaptive parameters, promises the most computation time-saving selection strategy when used to synthesise phased array weights to meet a far field requirement.

REFERENCES

[1] D. W. Boeringer and D. H. Werner; "Genetic Algorithms with Adaptive Parameters for Phased Array Synthesis" http://labs.ee.psu.edu/labs/dwernergroup/Projects.htm

[2] "SuperNEC Genetic Algorithm Optimiser V1.30" http://www.supernec.co.za/snec/html/sngaum.html