Dynamic Characterized Genetic Algorithm for Adaptive Beam Forming in WCDMA System

Tiong Sieh Kiong[1], Mahamod Ismail[2], Azmi Hassan[3]

[1] College of Engineering, Universiti Tenaga Malaysia (UNITEN)
km 7, Kajang-Puchong Road, 43009 Kajang, Selangor Darul Ehsan, Malaysia, siehkiong@uniten.edu.my

[2] Faculty of Engineering, Universiti Kebangsaan Malaysia (UKM)
43600 UKM Bangi, Selangor Darul Ehsan, Malaysia, mahamod@eng.ukm.my

[3] Faculty of Engineering, Universiti Kebangsaan Malaysia (UKM)
43600 UKM Bangi, Selangor Darul Ehsan, Malaysia, azmi@vlsi.eng.ukm..my

Abstract – **In this paper, a modified Genetic Algorithm named as Dynamic Characterized Genetic Algorithm (DCGA) is proposed. The new DCGA is then integrated into adaptive beam forming technique to reduce the power usage of adaptive antenna at WCDMA base station (Node B) for serving instantaneous mobile needs. As contradict to conventional GA, DCGA is able to adapt itself in chromosome representation and number of crossover points in order to reduce unnecessary searching space and thus, increase convergence rate efficiently. Power usage at Node B is used as fitness function to compare the performance of DCGA and GA. Simulation result has shown that DCGA converges faster and is superior in adaptive beam forming in the aspect of power usage at Node B as compared to conventional GA.**

Keywords – Adaptive beam forming, Dynamic Characterized Genetic Algorithm, WCDMA

I. INTRODUCTION

One of the major techniques to increase mobile communication system capacity is implementing adaptive antenna into the system. Adaptive antenna beam is capable of separating signals transmitted in the same frequency band provided the signals are separated by spatial domain [1],[2]. The adaptive beam forming process appropriately combines the signals received by the different antenna elements of an antenna array in order to form single output. There are many algorithms exist for adaptive beam forming and recently much attention was drown to algorithm with genetic algorithm (GA) [3],[4].

The optimization of the antenna array is typically subject to a range of constraints and the classic methods of array weight optimization may get trapped in local optima and resulting in a suboptimum beamforming performance [5]. Consequently, GA which is good for heuristic optimization is adapted for beamforming weight optimization problem. The GA assisted beamforming process requires a fitness function to evaluate the fitness of each individual throughout the evolutionary optimization. In this paper, a fitness function based on power usage of antenna system at Node B is developed for GA evaluation.

II. ADAPTIVE ANTENNA CONFIGURATION

In this paper, it is assumed that a linear circular array of L elements is placed at the far field of M uncorrelated point sources and the elements are separated by a distance of $d = \lambda/2$, where λ is the wavelength of the sources. Thus, the GA is employed to optimize a set of L complex array weights based on the fitness function. A snapshot of simulation is obtained to be presented in this paper, where there are 8 active unit equipments (UE) in a WCDMA cell in the particular simulation snapshot.

In the simulation, it is assumed that the configurations of adaptive antenna is a 9-array antenna ($L = 9$) which is placed in circular. Each antenna array can cover an area of 40 degree angle of its own and also coverage the areas of neighbouring arrays, as shown in Table 1. It is also assumed that the UEs distribution at the instantaneous moment is as shown in Table 2. Besides, Table 2 shows also the possible antenna arrays that can be used to serve the UEs in that particular simulation snapshot.

The authors acknowledge that this research is a part of project funded by government of Malaysia with IRPA 04-02-02-0062-EA289

Table 1. Adaptive Antenna Beam Coverage

Smart Antenna Element #	Coverage Angle (Degree)	Maximum Coverage (Degree)
0	$0^0 \sim 40^0$	$320^0 \sim 80^0$
1	$40^0 \sim 80^0$	$0^0 \sim 120^0$
2	$80^0 \sim 120^0$	$40^0 \sim 160^0$
3	$120^0 \sim 160^0$	$80^0 \sim 200^0$
4	$160^0 \sim 200^0$	$120^0 \sim 240^0$
5	$200^0 \sim 240^0$	$160^0 \sim 280^0$
6	$240^0 \sim 280^0$	$200^0 \sim 320^0$
7	$240^0 \sim 320^0$	$240^0 \sim 360^0$
8	$320^0 \sim 360^0$	$320^0 \sim 40^0$

Table 2. UE's Angle of Arrival

UE#	AOA	Possible Used Antenna
UE0	81^0	#1, #2, #3
UE1	95^0	#1, #2, #3
UE2	100^0	#1, #2, #3
UE3	107^0	#1, #2, #3
UE4	120^0	#1, #2, #3, #4
UE5	124^0	#2, #3, #4
UE6	134^0	#2, #3, #4
UE7	145^0	#2, #3, #4

III. CONVENTIONAL GA AND DYNAMIC CHARACTERIZED GA

Real parameter coding is applied in this research to represent the weights (starting and ending angle) of antenna beam. 16 memory bits are used to represent each weight solution.

(a) Conventional GA

In general, GA based optimization has to be able to perform six basic tasks [6];

1. Encode the potential array solutions with the aids of the genes of the GA,
2. Create a string of GA's genes for the sake of forming a chromosome,
3. Create an initial population of GA-based solutions,
4. Evaluate the assign fitness values to individuals in the population of the GA,
5. Perform reproduction with the aid of fitness-proportionate selection of individuals in the population for creating the next generation of better average fitness individuals,
6. Perform mutation of the individuals of the new generation by slightly perturbing the individuals for the sake of promoting a higher diversity of solutions in an effort to avoid getting trapped in local optima.

In the conventional GA, representation of the weights of all 9 elements adaptive antenna arrays is carried out at all times. Thus, a total of 144 memory bits representation is used for any solution optimization.

(b) Dynamic Characterized GA

There are 3 new implementations in DCGA; Dynamic Chromosomes Representation, Match Guided Selection (MGS) for crossover selection and Incremental Dynamic Crossover Points (IDCP) for number of crossover points in crossover process.

1. Dynamic Chromosomes Representation

In DCGA, the length of chromosomes that representing arrays weights (solutions) are not fixed. It is dynamically changed from time to time based on the complexity of the optimization problem. An additional process is incorporated to determine all antenna arrays that are possible to be used to serve all UEs at a particular moment. The weights of antenna arrays that are not possible to be used are dropped from the chromosomes representation. By referring to Table 2, a particular snapshot of simulation only needs to use 4 antenna arrays to serve all the UEs. As thus, DCGA will manipulate on a total of 64 bits chromosomes for array weights optimization.

2. Match Guided Selection (MGS)

Crossover process is the major process in GA operation. Thus, in order to improve the efficiency of solution searching, attention was drawn to the crossover selection. Figure 1 shows the process of new proposed crossover selection method, Match Guided Selection method (MGS). 2 storage pools (best storage pool and normal storage pool) are utilized under MGS method. Chromosomes are kept in the pools separately after random generation. The best storage pool is used to keep a fixed number of best chromosomes, meanwhile normal storage pool keeps the rest of the chromosomes. Roulette Wheel selection is then performed to select an individual from normal pool to be matched with an individual from the best storage pool. Roulette Wheel selection method is not applied to the best storage pool. All individuals in the best storage pool will take turn to be matched with individuals from normal pool. This method is develop due to the belief that parent chromosomes carrying better fitness values will more likely to produce better offspring after crossover compared to parent chromosomes with average fitness values.

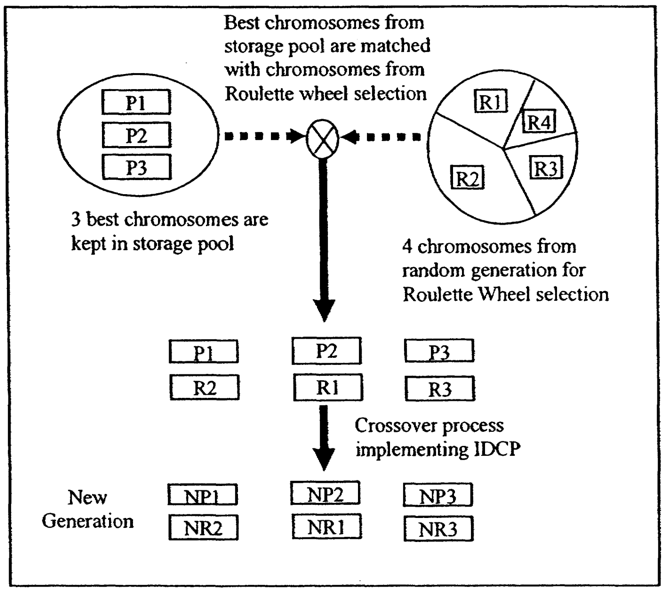

Fig. 1 Match Guided Selection Process

3. Incremental Dynamic Crossover Points (IDCP)

Another modified point in DCGA is the crossover operator. Literature has stated that real-parameter crossover operators are able to produce exploration or exploitation depending on the way in which they handle the current diversity of population [7]. Exploration will generate additional diversity starting from the current chromosomes and exploitation is to create better elements from the diversity [8]. Thus, an adaptive crossover operator namely Incremental Dynamic Crossover Points (IDCP) was proposed in this paper. As contradict to conventional GA where the number of crossover points is fixed throughout the crossover process, number of crossover points in IDCP is dynamically changed with fitness function value to obtain the benefit of exploring at the early generations and exploiting at the latter generations. The number of crossover points at certain generation depends on the ratio of fitness value with initial fitness value as shown in Equation 1.

$$\text{Crossover point, } \gamma = \sqrt[\eta]{\frac{f(g_B)}{f(g_0)}} \times \gamma_{max} \qquad (1)$$

Where γ_{max} is the maximum crossover points allowed, $f(g_B)$ is the best fitness value obtained, $f(g_0)$ is the initial fitness value and η is the IDCP factor. Through simulation, It is found that $\eta = 0.2$ gives the best simulation result.

(c) Fitness Function

Fitness function developed to evaluate the fitness of individual in the aspect of power usage at Node B is shown in Equation 2.

$$E(f) = \sum_{l}^{L}\prod_{m}^{M} U_{l,m} + \sum_{m}^{M}(W_m + A_m + \varepsilon P_m) \qquad (2)$$

where,

$$U_l \begin{cases} = 0 \text{ if } \omega_{m,min} \le UE_l \le \omega_{m,max}. \\ = 1 \text{ if } \omega_{m,min} \le UE_l \cup UE_l \ge \omega_{m,max} \end{cases}$$

$$W_m \begin{cases} = 1 \text{ if } \omega_{m,max} - \omega_{m,min} < \rho \\ = 0 \text{ if } \omega_{m,max} - \omega_{m,min} \ge \rho \end{cases}$$

where $\omega_{m,max}$ dan $\omega_{m,min}$ are maximum coverage angle and minimum coverage angle for smart antenna element #m and ρ ialah minimum separation angle between antenna beam.

$$A_m \begin{cases} = 1 \text{ if } ((\omega_{m,min} \cup \omega_{m,max} \le (m-1)\delta)) \cup \\ ((\omega_{m,min} \cup \omega_{m,max}) \ge (m+1)\delta) \\ = 0 \text{ if } (m-1)\delta \le (\omega_{m,min} \cup \omega_{m,max}) \le (m+1)\delta \end{cases}$$

where $\delta = 360/M$ and M is number of smart antenna elements and;

$\varepsilon P_m = p_m w_m \times 10^{0.25}$

Where p_m is poynting power transmitted, ε is the weight factor for p_m (ε =0.001 in this paper), w_m is solid angle of antenna beam. It is assumed that the cable loss at antenna element is 2.5dB.

III. SIMULATION PARAMETERS

Table 3 shows the main simulation parameters used in conventional GA and DCGA.

Table 3: GA and DCGA Simulation Parameters

GA Simulation Parameter	Value
Maximum Generations	5000
Length of Chromosomes	144 bits (GA) Varying (DCGA)
Population, p_o	100
Selection Method	Roulette Wheel (GA) MGS (DCGA)
Crossover rate, p_c	0.90
Mutation rate, p_m	0.05
Mutation point, m_p	1
Number of best Chromosomes	3
Number of Simulation , c_y	10
Power factor of IDCP, η	0.2 (DCGA)

IV. SIMULATION RESULT

Simulation results are shown in Figure 2 to 6. Figure 2 shows the memory allocation used by conventional GA and DCGA for the simulation snapshot. Memory bits in circle are the memory bits for representation of one chromosome. In Figure 2(a), it can be seen that conventional GA uses more memory bits in representing a chromosome compared to DCGA as shown in Figure 2(b).

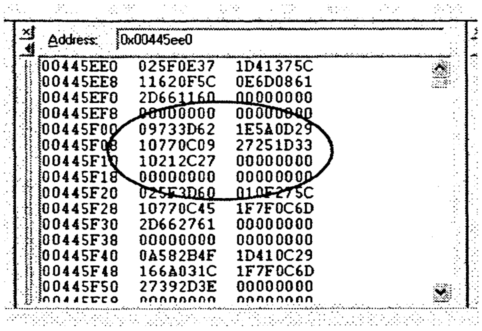

Fig. 2(a) Memory Usage with GA

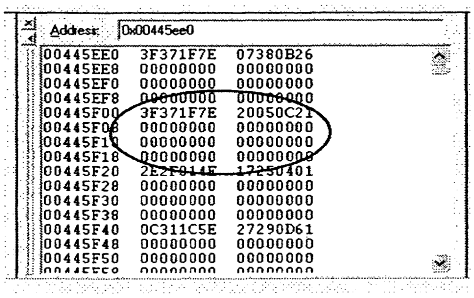

Fig. 2(b) Memory Usage with DCGA

Convergence rate is used as the main criteria to compare the efficiency of GA and DCGA in obtaining a solution for adaptive beam forming. Comparison of convergence rate comparison with different optimisation technique is shown in Figure 3. 10 simulations were run and the best result of every optimisation was obtained. The best fitness function value for the simulation scenario was calculated manually and used as benchmark, which is indicated as "ideal" for the simulation. It can be found that DCGA converged faster compared to GA as it converged to ideal fitness function value at the range of 2000 generations. However, GA can only converge to the ideal value at the range of 5000 generations.

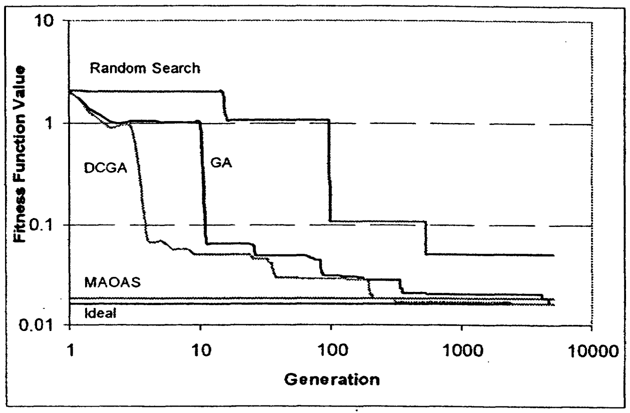

Fig.3 Convergence Comparison Of Beam Forming Assisted By Different Techniques

Figure 4 shows the fitness function value obtained from conventional GA and DCGA after 50 independent simulations with maximum number of generations set to 5000 generations. The fitness function values are sorted in ascending. It can be seen that most of the fitness function value with GA are more than 0.04; however most of the fitness function value with DCGA are less than 0.04. This shows that the convergence rate of DCGA is faster than conventional GA.

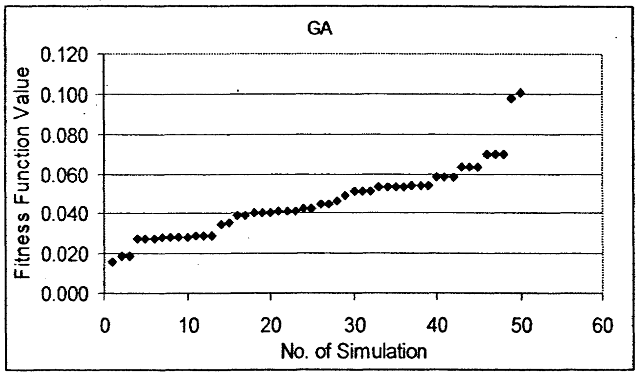

Fig. 4(a) Sorted Fitness Value for GA For 50 Simulations

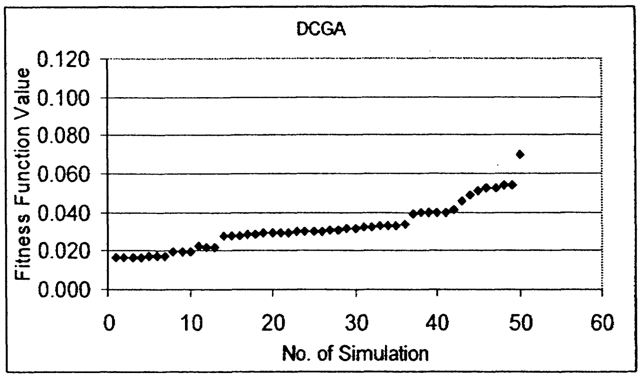

Fig. 4(b) Sorted Fitness Value for DCGA For 50 Simulations

Simulation result of different crossover selection method with different number of best chromosomes kept in pool is shown in Figure 5. Simulation result shows that Match Guided Selection (MGS) method gives the best result by producing lowest function value for all cases as compared to conventional selection method such as Roulette Wheel and Best Fit selection.

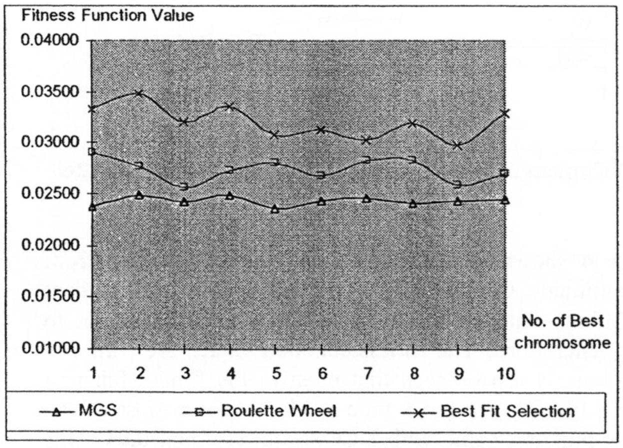

Fig. 5 Comparison of Fitness Function Value under Different Crossover Selection Method

Figure 6 shows the convergence performance by using constant number of crossover points (CCP) and incremental dynamic crossover points (IDCP) in crossover process. Simulation result showed that IDCP has performed better compared to CCP in both GA and DCGA. IDCP is able to produce lower fitness value, which correspond to lower power usage at the maximum crossover point (MCP) set to 15% onwards.

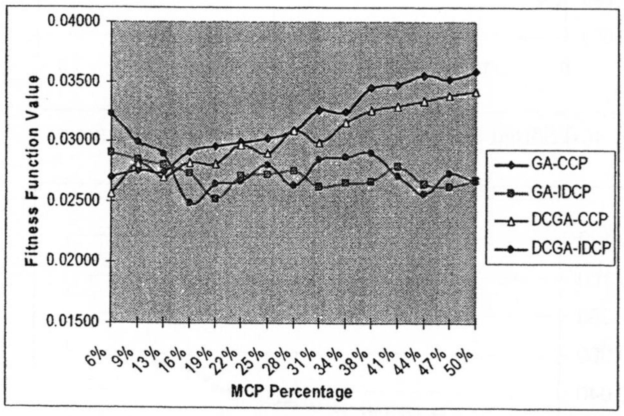

Fig. 6 Performance Comparison between CCP and IDCP

Simulation was also carried up for dynamic moving UE with Poisson distributed traffic model. 5000 UEs were simulated under various traffic loads condition in different cell size. Figure 7 and 8 shows the total power usage at Node B for all antennas elements. Simulation result showed that DCGA assisted adaptive antenna consumes the least power under different traffic load and cell size conditions.

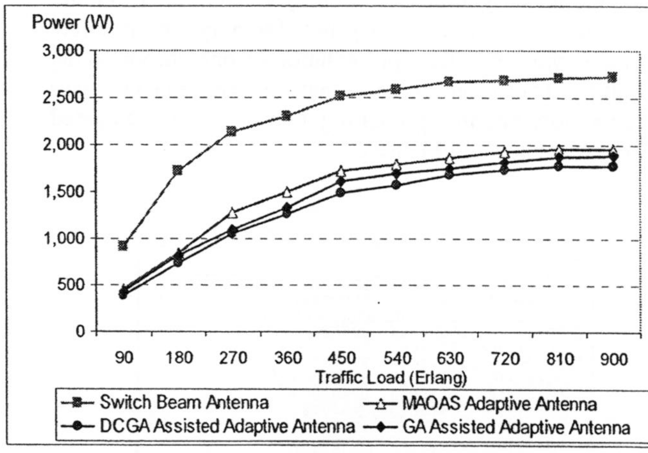

Fig. 7 Total Power Usage at Node B under Different Traffic Loads Condition.

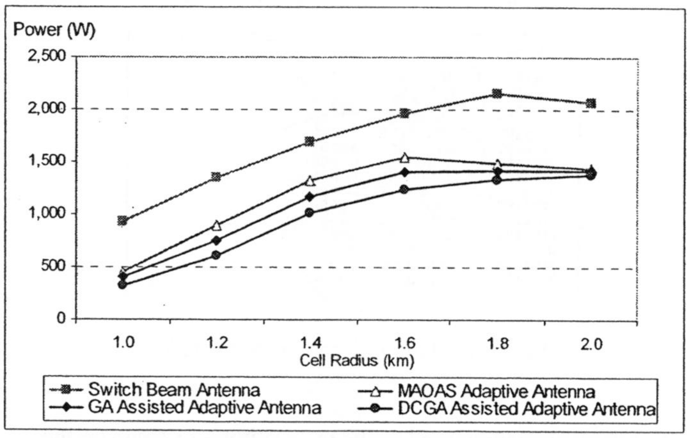

Fig. 8 Total Power Usage at Node B with Different Cell Sizes.

V. CONCLUSION

Simulation results have shown that Dynamic Characterized Genetic Algorithm (DCGA) performs better as compared to conventional GA in assisting adaptive beam forming. In this paper, the comparison study was carried out by comparing convergence rate and fitness value obtained by both DCGA and GA for adaptive antenna beam forming. In most of the cases, DCGA converges faster and produce better fitness value which corresponds to lower power usage of adaptive antenna at Node B.

REFERENCES

[1] M. Vitale, G. Vesentini, N.N. Ahmad, L. Hanzo, "Genetic Algorithm Assisted Adaptive Beamforming", *Proceedings for IEEE 56th Vehicular Technology Conference* 1:601-605.

[2] J.S. Blogh, L. Hanzo, "Third-Generation Systems and Intelligent Wireless Networking – Smart Antennas and Adaptive Modulation", John Wiley, 2002, p.408

[3] R. L. Haupt, "Phase-Only Adaptive Nulling with a Genetic Algorithm", IEEE Transactions on Antennas and Propagation, vol. 45, No. 6, June 1997. pp. 1009-1015.

[4] Y. Yashchyshyn and Piasecki M., "Improved Model of Smart Antenna Controlled by Genetic Algorithm", VI-th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics. Ukraine, 2001. pp. 147-150.

[5] J. Litva, T.K.Y.Lo, "Digital Beamforming in wireless Communications" Artech House Publishers, 1996.

[6] J.M. Johnson and Rahmat-Samii, "Genetic Algorithms in Engineering Electromagnetics", *IEEE Antennas and Propagation Magazine*, vol. 39, pp.7-21, 1997.

[7] F. Herrera, M. Lozano and A.M.Sanchez. "Hibrid Crossover Operators for Real-Coded Genetic Algorithms: An Experimental Study", *Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain.* 2003.

[8] H.G. Beyer and K. Deb. "On self-Adaptive Features In Real-Parameter Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation* 5(3), 2001. pp. 250-270.