Heuristics based Learning on Human Psychology

R. Lakshmi,
Assistant Professor
Department of Computer Science, Pondicherry University, Karaikal Campus

K. Vivekanandan,
Professor
Department of Computer Science,
Pondicherry Engineering College, Puducherry

ABSTRACT
This paper summarizes recent research on heuristic based learning procedures called Genetic Algorithms (GAs) and particularly focuses on genetic primary operators. There are different types of genetic operators existing that serve to improve the performance of genetic algorithms. Genetic Algorithm is composed of genetic operators and other genetic parameters. The primary genetic operators are selection, crossover and mutation. The performance of Genetic Algorithm mainly depends on its Genetic Operators. While solving a particular problem, 70% of the time will be spent in searching the appropriate genetic operators and their probability values. So it is much important to select the correct operator and their probabilities so as to provide optimal solutions for a given complete problem. The ultimate aim of genetic algorithm is to minimize the time and space complexities and produces optimized results. This research helps in how primary genetic operator is modified or enriched to improve the performance of genetic algorithms.

Keywords
Genetic Algorithms, Heuristics, Binary Encoding, Crossover Operator, Mutation, Fitness Score, Optimization, psychology.

1. INTRODUCTION
One approach to the design of more flexible computer systems is to extract heuristics from existing adaptive systems. Genetic algorithms (GAs) are adaptive methods which are used to solve search and optimization problems. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and "survival of the fittest inspired by Darwin's theory. Genetic algorithms (GAs) are stochastic generate-and test search techniques based on the principles of evolutionary simulation. GA is more useful when the search space is large, complex or poorly understood. If domain knowledge is scarce or expert knowledge is difficult to encode to narrow the search space, then GA is used to solve these problems.

An effective GA representation and meaningful fitness evaluation are the keys of the success in GA applications. Genetic Algorithms work with a population of "individuals", each representing a possible solution to a given problem .Each individual is assigned a "fitness score" according to how good a solution to the problem is. The highly-fit individuals are given opportunities to "reproduce", by "cross breeding" with other individuals in the population. This produces new individuals as "offspring", which share some features taken from each "parent". The least fit members of the population are less likely to get selected for reproduction, and so "die out". In this way, over many generations, good characteristics are spread throughout the population. By favoring the mating of the more fit individuals, the most promising areas of the search space are explored. If the GA has been designed well, the population will converge to an optimal solution to the problem. Exploration of the search space in genetic algorithms is performed by genetic operators simulating genetic crossover and mutation operation. These genetic operators can be carefully chosen by the user in advance, or this can be determined by a termination condition like the detection of convergence or the achievement of an upper bound for the fitness function.

1.1 Genetic Algorithm Pseudocode

1. Encode it as an artificial chromosome.
2. Discriminate the good and the bad solutions which are done by choosing intuitively better solutions over worse ones or we can use computer simulation or mathematical model (fitness function formulation) which is used for evolution of future generations.
3. Create an initial population of encoded solutions randomly or by using prior knowledge of good solutions.
4. Using genetic and selection operators, process the population iteratively to create a sequence of populations that will contain more good solutions as time goes on.
5. Apply breed operators crossover and mutation to form the next generation.
6. Iterate the process until the termination criteria is met.

Before generating the initial population as given in step 3, the problem has to be represented in any encoding form known as chromosome or individual. By doing permutations on individuals, the initial population is generated randomly. The idea of this paper is appropriately suitable for binary coded representation. Many real problems use binary coded representation. This research is demonstrated on human psychology problem which also uses binary encoded representation. In binary encoding, the chromosomes are represented by the combination of 0's and 1's. The next generation population (step 4 & 5) is obtained by applying three primary genetic operators namely selection, crossover and mutation operators. Instead of three operators this paper uses only two and does not use mutation operator. The results were compared and proved to be efficient than the conventional genetic algorithm which predominantly uses mutation operator.

2. GA APPROACH TO HUMAN PSYCHOLOGY PROBLEM

One of the challenges of human psychology problem is the great common factor of how about the person is? This paper measures the psychological features of a person with the help of evolutionary computation and returns how about the person is!. Genetic algorithm is a part of an evolutionary computation. GA is a method that human and evolutionary computation (EC) cooperate to search the optimal solutions in the psychological search space of the human. GA reduces implicit psychological space into parameter-features space which is solvable to computer.

2.1 Chromosome Representation

The problem described here is how to analyze the traits of human beings such as intellectual level of a particular person, level of intelligent quotient (IQ), stress, self confidence, self respect, humanity, integrity etc. This paper deals with these seven attributes as characteristics of each individual represented as a chromosome in a genetic algorithm is given below.

P1:	1	1	0	1	1	0	0
P2:	0	0	1	0	1	1	1

Fig 1: Binary Chromosome Representation

Figure 1 shows the chromosome representation as binary numbers indicates the selected characteristics of one of the individual in the population. These seven binary values are randomly permutated and generates the population size of 2^7 . Each row of binary numbers represents the traits of one of the individual in the population. Traits associated with parent1 indicates that the person is an intellectual person, his\her IQ is good, stress level is zero, having lot of self confidence and self respect but does not have the qualities of humanity and integrity. Looking into parent1 (P1) data, one could say that the person is a selfish one not the altruistic one. The values of P2 indicate that the person is an ordinary person.

2.2 Fitness Evaluation of each Individual

After generating the initial population, fitness of each and every individual has to be calculated. Based on their fitness score the individuals are pushed into the next generation. The fitness calculation of individuals is shown below.

$$\textbf{Score} = f(\text{individual}) = f(p1, p2, \ldots p_{n=7})$$

Where ‘p’ is a parameter holds the value of traits of each individual.

f1(1 1 0 1 1 0 0) = 4

f2(0 0 1 0 1 1 1) = 4

f3(1 1 1 1 1 1 1) = 7

f4(0 0 0 0 0 0 0) = 0

Fig 2: Fitness Score of each Individual

For example the fitness score of the above four individuals are 4, 4, 7 and 0. The fitness function counts the number of 1’s in a chromosome and the total number of 1’s is taken as a fitness score of each individual. Wherever ‘0’ presents in a chromosome, it says that the particular character is missing with the person. If you take the 4^{th} chromosome all parameter holds the value of 0’s, it means that the person is not having any of the good qualities, so his\her fitness score is ‘0’. After calculating the fitness of all individuals, apply primary genetic operators and breed the next generation.

2.3 Natural Selection

There are enormous selection mechanisms namely roulette wheel selection method, tournament selection, elite method, scaling selection, random selection etc. available. These are all another approach to natural selection and these operators are initially used for choosing the best individuals, that is, for selecting those individuals with higher fitness values. These operators cannot introduce any new individuals into the population meaning that they cannot find new points in the search space but direct the search towards some peak. These operators help in choosing better solutions or chromosomes (chromosomes whose fitness values are biased towards the best values) from the current generation and thereby generating more offspring from these chromosomes in the next generation. The solutions selected for the next generation therefore produce best optimal fitness and this exhibits the survival-of-the-fittest mechanism adopted in GA.

This Paper uses elite method and roulette wheel selection method as selection operators.

2.3.1 Elite Method

The best members of the population are guaranteed to be selected and are put into the next generation without alteration at each time step. The best individuals cannot be lost due to random sampling, and selection strength used to pick the other members of the population can be weaker to maintain diversity. This method sorts the fitness values in a descending order and chooses the top ‘n’ individuals whose fitness score is higher than others. Where ‘n’ is the elitist probability value for selecting the number of individuals to the next generation.

2.3.2 Roulette Wheel Selection

Also called as the fitness proportional selection (FPS selection), this method simply chooses the individuals in a statistical fashion based solely upon their relative (i.e., percentage) cost or fitness values. i.e. highly fit individuals will make a major portion of the next generation individuals than the weak individuals. This is in analogy with a biased roulette wheel, associated with each chromosome is a wedge of the wheel, whose size is proportional to the fitness of the chromosome. Selection is equivalent to a spin of the wheel.

2.4 Crossover Operation

Crossover is one of the genetic operators that can be applied to the individuals of a population. It takes two parents, fix up the cut points randomly and swap these segments to produce its offspring to the next generation. Crossover operator is applied in the hope that the new offspring will be better than its parents. Crossover is used to explore different promising areas of the search space, allowing the convergence to the optima (exploration). Crossover is generally applied with higher probability. If the population size is too small, the GA without recombination outperforms the GA with recombination. If the population is large or the search space is more the GA with crossover outperforms the GA without crossover. The number of points in crossover is the number of times the genetic material in an offspring is inherited from the first parent, second parent and vice versa. By concatenating these two segments after the crossover points form a new offspring. This paper uses multi point crossover whose working principle is as follows.

2.4.1 Multi Point Crossover

Multi point crossover will amplify the effect of recombination. Multi point crossover operator is categorized on the basis of allele frequencies or on segmentation or numerical operations on real valued alleles. For multi-point crossover, *m* crossover positions are chosen at random with no duplicates and sorted in ascending order. Then, the genes between successive crossover points are exchanged between the two parents to produce two new offspring. The section between the first variable and the first crossover point is not exchanged between individuals.

Consider the following two individuals with 7 binary variables each:

Individual 1 0 1 1 1 0 0 1

Individual 2 1 0 1 0 1 1 0

The chosen crossover positions are:

Cross positions (m=3) 2 4 6
After crossover the new individuals are created:

Offspring 1 0 1| 1 0 | 0 0 | 0

Offspring 2 1 0| 1 1 | 1 1 | 1

From the above example, the fitness value of an individual 1 is 4, after doing crossover operation its value is decreased into 2 ie. The stronger individual becomes weaker one and it is also moved to the next generation without ignoring it. If we take the second individual its score is 4 and after crossover takes place its score becomes 6, the individual becomes stronger. Likewise the crossover operation is performed on many individuals and form the next generation.

2.5 Mutation Operator

Mutation is a genetic operator which involves only one individual in the creation of a better offspring with improved parental traits and is identical to the parent. It helps in the exploitation principle of GA. It can introduce traits that are needed in the solution but were not present in the initial population. By itself mutation represents a “random walk” in the neighborhood of a particular solution. If done repeatedly over a population of individuals, we might expect the resulting population to be indistinguishable from one created at random. This paper uses single point mutation operator which is explained below.

Individual 1: 0 1 1 1 0 0 1
Fitness Score: 4

Randomly chosen mutation position is allele number ‘4’.

After mutation the new offspring is
Offspring 1: 0 1 1| 0 | 0 0 1
Fitness Score: 3

3. DIRECTIONS OF THIS RESEARCH

As far as the genetic algorithm is concerned, before get into the GA steps the problems (input parameters) have to be clearly encoded using any of the representation systems and generate the initial population at random. The next procedure is computing the fitness of all individuals the fitness function varies from one problem to the other problem. Based on their fitness score and the primary operators which GAs use, the next generation of the individuals are generated. Conventional Genetic algorithms always work in this nature for any NP hard problem. This research slightly modifies the existing GA ie. it does not apply mutation operator for breeding. Only crossover operator can be used to breed the next generation. If any problem can be represented as binary information then this research says that for binary encoded chromosomes performing mutation is a merely time consuming process. Though sometimes it gives feasible offspring the time taken to produce this feasible offspring is not negligible. Also the mutation mechanism creates duplicates to correct the duplicate chromosome an additional operator is being used which doubles the work and time. This concept is applied on human psychology problem and the results were analyzed with the conventional GA.

4. TESTS AND RESULTS

The input data (parameters: p1, p2, p3, p4, p5, p6, p7) are chosen as traits of each individual. These data are collected from the students of Pondicherry University, Karaikal Campus. This concept was employed in genetic algorithm, the outcome of the implementation measures or rates the attitude of each student and also predicts their generations. The collected data in binary form as shown below in table 1.

Table 1. Traits of Students as Sample Input Data

Student Sl. No.	P1	P2	P3	P4	P5	P6	P7
1	1	1	1	0	1	0	0
2	0	0	0	1	1	1	1
3	0	0	1	0	1	0	1
4	1	0	1	0	1	0	1
5	0	1	0	1	0	1	0
6	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1
10	1	1	0	0	1	1	0

Table1 contains only few sample data but the actual population size used in this work 128 (2^7). After running the programs using genetic algorithms without mutation operator and with mutation operator, the results show that the GA without mutation outperforms than GA with mutation operator in terms of time consumption and space complexity as shown in the following tables.

Table 2. Output Data using GA without Mutation

Offspring	P1	P2	P3	P4	P5	P6	P7
1	1	1	1	1	1	0	0
2	1	0	0	1	1	1	1
3	0	0	1	0	1	1	1
4	1	0	1	0	1	0	1
5	0	1	0	1	0	1	0
6	1	1	1	1	1	1	1
7	0	0	1	1	1	1	0
8	1	1	1	0	0	0	1
9	1	0	1	1	0	0	1
10	1	1	0	0	1	1	0

Table 3. Output Data using GA with Mutation

Offspring	P1	P2	P3	P4	P5	P6	P7
1	0	0	1	1	0	0	0
2	1	0	0	1	1	0	0
3	0	0	1	0	1	1	0
4	1	0	1	0	1	0	0
5	0	1	0	0	0	0	0
6	1	1	0	0	1	0	1
7	0	0	1	1	0	1	0
8	1	0	1	0	0	0	1
9	1	0	0	1	0	0	1
10	1	1	0	0	1	1	1

The table 2&3 shows the traits of their offspring as output data without and with mutation in a genetic algorithm. The efficiency of these programs is shown below in table 4.

Table 4. Results Comparison

Population Size = 2^7	Number of Runs/Converge Time
GA with Mutation	4553
GA without Mutation	2054

We have experimented with various rates (Roulette wheel selection rate and Multi Point Crossover rate) of genetic parameters on huge input data which also results huge amount of output data. Only few data samples have been taken and evaluated the efficiencies of both the systems. From the results it is clearly proved that the proposed system doubles speed of the simple genetic algorithm.

5. CONCLUSION

This paper strongly recommends that the binary genetic algorithm not necessarily to use mutation operator which results in increasing speed and performance of the simple GA. GA without one breed operator mutation which can be effective on binary genetic algorithm. The proposed systems avoid getting duplicate chromosomes during their recombination or breeding process. This excellent concept makes the conventional GA as fast GA (FGA) by not incorporating the mutation operator in the GA procedure. The proposed system achieves good results at less number of iterations than the conventional GA for the human psychology problem is successfully verified.

6. REFERENCES

[1] Arthur L. Corcoran, Roger L. Wainwright, "Reducing disruption of superior building blocks in genetic algorithms", Proceedings of the, ACM SIGAPP Symposium on Applied Computing February, 2003.

[2] Davis L. "Handbook of Genetic Algorithms" Van Nostrand Reinhold, 1991.

[3] Ayed A. Salman, Kishan Mehrotra, and Chilukuri K. Mohan, "Adaptive linkage crossover evolutionary computation", Evolutionary computation, Vol.8 1(3), September 2000.

[4] Dr.Sabry M. Abdel-Moetty "Enhanced Traveling Salesman Problem Solving using Genetic Algorithm Technique with modified Sequential Constructive Crossover Operator", IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.6, June 2012

[5] Chen, Y.-p., & Goldberg, D. E., "An analysis of a reordering operator with tournament selection on a GA-hard problem", Lecture Notes in Computer Science (LNCS), 2723, pp. 825–836, 2009.

[6] Chen, Y.-p., Peng, W.-C., & Jian, M.-c, "Particle swarm optimization with recombination and dynamic linkage discovery", IEEE Transactions on Systems, Vol. 37(6): pp.1460–1470, 2012

[7] European Graduate Student, "Workshop on Evolutionary Computation" 2012

[8] S.Siva Sathya, S.Kuppuswami, Department of Computer Science, Pondicherry University, "Gene Silencing for Course Time-Tabling with Genetic Algorithm".

[9] Anabela Simões, Ernesto Costa, 2003, "Improving the Genetic Algorithm‟s Performance when Using Transformation", in Proc. of the Sixth International Conference on Neural Networks and Genetic Algorithms (ICANNGA'03), Springer, pp. 175-181, April-2003