A Preliminary Survey on Genetic Algorithm Techniques

Mayilvaganan M#1, Geethamani G.S”2

#Associate Professor, Department of Computer Science, P.S.G College of Arts and Science, India, Coimbatore.

#Assistant Professor, Research Scholar, Department of Computer Science, Karpagam University, India, Coimbatore.

Abstract— In recent years, data mining and Genetic algorithms is an essential aspect for searching and generating association rules among the large number of itemsets. Genetic algorithms maintain a population pool of candidate solutions called strings or chromosomes. Each chromosome p is a collection of building blocks known as genes, which are instantiated with values from a finite domain. Associated with each chromosome is a fitness value which is determined by a user defined function, called the fitness function. The performance of a GA is dependent on the genetic operators in general and on the type of crossover operator, in particular. Effective crossover in a GA is achieved through establishing the optimum relationship between the crossover and the search problem itself. In this paper, an preliminary studies have been carried out to enable the researcher to identify the various genetic algorithm methods.

Keywords — Data Mining, Genetic Algorithm, itemsets, chromosomes, crossover, fitness function.

I. INTRODUCTION

In 1975, John Holland was developed the Genetic Algorithm at University of Michigan. Genetic algorithms are inspired by Darwin’s theory about the evolution, termed as “Survival of the Fittest”. It also simulates natural evolution with a combination of selection, recombination and mutation to evolve a solution to the problem[1]-[4]. It randomly search the dataset to solve the optimization problems. It means that better and better solutions evolve from previous generations until a near optimal solution is obtained. It provides efficient, effective techniques for optimization and machine learning applications[3]. This algorithm is Widely-used today in business, scientific and engineering circles. Genetic algorithm is an iterative procedure that represents its candidate solutions as strings of genes called Chromosomes. A group of individuals (Chromosomes) called population. Population is modified in the each iteration of the algorithm. Genetic Algorithm’s iterations are called as generations. Standard Genetic algorithm apply genetic operators such as selection, crossover and mutation.

Genetic Operators

The GA maintains a population of n chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan (mutation and crossover). Consequently highly fit solutions are given more opportunities to reproduce (selected for next generation), so that offspring inherit characteristics from each parent[5][7]. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size (population size).

Selection: According to Darwin’s evolution theory, the chromosomes with higher fitness ratings are selected from the population to be the parents to crossover that should survive and create new offspring.

Crossover: It leads to effective combination of schemata (sub-solutions on different chromosomes). It means choosing a random position in the string and exchanging the segments either to the right or to the left of this point with another string partitioned similarly to produce two new offspring.

II. RELATED WORK

Wakabi-Waiswa, P.P., et al., proposed [16] “Generalized Association Rule Mining Using Genetic Algorithms”. In this paper, Association rule mining is designed for combining the Genetic Algorithms and a modified a-priori based algorithm. It yields very fast results. It generalized a
very large database of transactions, where each transaction contain a set of items, and a classification on the items, then the associations between items at any level of the classification have been found[6]-[9]. It improved the performance of minimum support and number of items. It also improves the various other characteristics limitless number of roots and levels in the classification, depth-ratio and number of transactions.

Ghosh S, Biswas S, Sarkar D and Sarkar P.P, “Mining Frequent Itemsets Using Genetic Algorithm”, proposed [6] the algorithm to find frequent itemsets using genetic algorithm. The association rule mining algorithm like apriori, partition, fp-tree, etc., generate the frequent itemsets. However, it takes too much time to compute the frequent itemsets. The main aim to introduce genetic algorithm is to reduce the computing time. Genetic algorithm performs as global search to generate the frequent itemsets[9][10][12]. The time complexity is less when compared to the association rule mining algorithm because the genetic algorithm is based on the greedy approach. Dou W, Hu J, Hirasawa K and Wu G, “Quick Response Data Mining Model Using Genetic Algorithm”,[4] proposed this paper to find the maximal frequent item sets using Genetic algorithm. In this paper, the authors defined some parameters because these parameters are used in the Genetic algorithm operators. The defined parameters are Individual Identity (IVI), Individual Fitness (IVF), Upgrade Index (UI), and Upgrade Genes (UG).

Individual Identity (IVI) contains the unique symbols of each chromosome in the individual. The individuals are distinguished by these symbols. Individual Fitness (IVF) has the number of items. If the individual cannot create a frequent itemset, then the IVF is set to 0, otherwise, IVF is the number of items and is set to 1. Upgrade Index (UI) is the negative number that shows the distance for getting the frequent itemset of the individual. The larger value of UI is, the more possible the frequent itemset is generated through using the Genetic operators. Upgrade Genes (UG) is the set of genes needed by the individual to enhance the UI. In more situations to know whether the individual can produce the frequent itemset and also which genes contained the chromosomes which are used to produce the frequent itemsets[12][15]. The parameter UG helps us to find both the individual and the genes.

The genetic operator selection uses the value of IVF for getting the current maximal frequent itemsets. The operator crossover adopts heuristic crossover checks whether the parent chromosome can be replaced by another chromosome using the UI parameter[8]. The heuristic mutation is adopted by the genetic operator mutation uses the UG to judge which transaction has lower relationship.

Yan X, Zhang C and Zhang S, developed[13] “Genetic Algorithm-based Strategy for Identifying Association Rules without Specifying Actual Minimum Support”, for generating the association rule using the genetic algorithm without specifying the minimum support and the confidence is used as the fitness function.

First, genetic algorithm is developed for Boolean association rule mining. Initializing the select operator \(pop[i]\) to produce the new one \(pop[i+1]\). Then apply the crossover for the new population with probability \(cp\) to reproduce offspring. Each chromosomes is mutated with probability \(mp\) for producing the high quality chromosomes.

III. WORKING PRINCIPLE OF GENETIC ALGORITHMS (GAS)

The workability of genetic algorithms (GAs) is based on Darwinian’s theory of survival of the fittest. Genetic algorithms (GAs) may hold a chromosome, a gene, set of population, fitness, fitness function, breeding, mutation and selection. Genetic algorithms (GAs) begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is stimulated by the possibility that the new population will be better than the old one. Further, solutions are selected according to their fitness to form new solutions, that is, offsprings. The above process is repetitive until some condition is satisfied. Algorithmically, the basic genetic algorithm (GAs) is outlined as below[14]-[16]:

ISSN: 2231-2803 http://www.ijcttjournal.org
Step I [Start] Create random population of chromosomes, that is, suitable solutions for the problem.
Step II [Fitness] Estimate the fitness of each chromosome in the population.
Step III [New population] Create a new population by repeating following steps until the new population is generated.
   a) [Selection] Choose two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.
   b) [Crossover] With a crossover probability, cross over the parents to form new offspring, that is, children. If no crossover was performed, offspring is the exact copy of parents.
   c) [Mutation] With a mutation probability, mutate new offspring at each locus.
   d) [Accepting] Place new offspring in the new population.
Step IV [Replace] Use new generated population for a further run of the algorithm.
Step V [Test] If the end condition is satisfied, stop, and return the best solution in current population.
Step VI [Loop] Go to step 2.

The genetic algorithms performance is largely influenced by crossover and mutation operators.

IV. ENCODING TECHNIQUE IN GENETIC ALGORITHMS (GAS)

Encoding techniques in genetic algorithms (GAs) are problem specific, which transforms the problem solution into chromosomes. Various encoding techniques used in genetic algorithms (GAs) are binary encoding, permutation encoding, value encoding and tree encoding.

3.1 Binary encoding
It is the most common form of encoding in which the data value is converted into binary strings. Binary encoding gives many possible chromosomes with a small number of alleles. A chromosome is represented in binary encoding[8].

3.2 Permutation encoding
Permutation encoding is best suited for ordering or queuing problems. Travelling salesman is a challenging problem in optimization, where permutation encoding is used. In permutation encoding, every chromosome is a string of numbers[5].

3.3 Value encoding
Value encoding can be form number, real number on characters to some complicated objects. Value encoding is technique in which every chromosome is a string of some values and is used where some more complicated values are required[4].

3.4 Tree Encoding
It is best suited technique for evolving expressions or programs such as genetic programming. In tree encoding, every chromosome is a tree of some objects, functions or commands in programming languages. Locator/identifier separation protocol (LISP) programming language is used for this purpose. Locator/identifier separation protocol (LISP) programs can be represented in tree structure for crossover and mutation. In tree encoding, the chromosomes are represented. There are no specific directions for using the type of encoding scheme in the specified problem rather, it depends upon the applicability and the requirements of the problem[7].

4. Selection Techniques in Genetic Algorithms (GAs)
Selection is an important function in genetic algorithms (GAs), based on an evaluation criterion that returns a measurement of worth for any chromosome in the context of the problem. It is the stage of genetic algorithm in which individual genomes are chosen from the string of chromosomes. The commonly used techniques for selection of chromosomes are Roulette wheel, rank selection and steady state selection[3].

4.1 Roulette wheel selection
In this method the parents are selected according to their fitness. Better chromosomes, are having more chances to be selected as parents. It is the most
common method for implementing fitness proportionate selection. Each individual is assigned a slice of circular Roulette wheel, and the size of slice is proportional to the individual fitness of chromosomes, that is, bigger the value, larger the size of slice is. The functioning of Roulette wheel algorithm is described below[11]:

Step 1 [Sum] Find the sum of all chromosomes fitness in the population.
Step 2 [Select] Generate random number from the given population interval.
Step 3 [Loop] Go through the entire population and sum the fitness. When this sum is more than a fitness criteria value, stop and return this chromosome.

Figure 3 (a) shows Roulette wheel for six individuals having different fitness values. The Sixth individual has a higher fitness than any other, it is expected that the Roulette wheel selection will choose the sixth individual more than any other individual.

4.2 Rank selection method
The application of Roulette wheel selection method is not satisfactory in genetic algorithms (GAs), when the fitness value of chromosomes differs very much. It is a slower convergence technique, which ranks the population by certain criteria and then every chromosome receives fitness value determined by this ranking. This method prevents quick convergence and the individuals in a population are ranked according to the fitness and the expected value of each individual depends on its rank rather than its absolute fitness. For example, if the best chromosome fitness is 80 percent, its circumference occupies 80 percent of the roulette wheel and then other chromosomes will have minimum chances to be selected. On the other hand, the rank selection first ranks the population according to their fitness and then every chromosome receives ranking. The worst will have fitness 1, the second worst will have a fitness of 2, and the best one will have a fitness value n, where n is the number of chromosomes in the population.

4.3 Steady-state selection
This method replaces few individuals in each generation, and is not a particular method for selecting the parents. Only a small number of newly created offsprings are put in place of least fit individual. The main idea of steady-state selection is that bigger part of chromosome should retain to successive population.

5. Genetic Algorithms (GAs) Operators
Genetic algorithms (GAs) can be applied to any process control application for optimization of different parameters[12]-[14]. Genetic algorithms (GAs) use various operators viz. the crossover, mutation for the proper selection of optimized value. Selection of proper crossover and mutation technique depends upon the encoding method and as per the requirement of the problem.

5.1 Crossover
It is the process in which genes are selected from the parent chromosomes and new offspring is created. Crossover can be performed with binary encoding, permutation encoding, value encoding and tree encoding.

5.1.1 Binary encoding crossover
In binary encoding, the chromosomes may crossover at single point, two point, uniformly or arithmetically. In single point crossover, a single crossover point is chosen and the data before this point are exactly copied from first parent and the data after this point are exactly copied from the second parent to create new offsprings. Two parents in this method give two new offsprings.

5.1.2 Uniform Crossover
In uniform crossover, data of the first parent chromosome and second parent chromosome are randomly copied.

5.1.3 Arithmetic Crossover
In arithmetic crossover, crossover of chromosomes is performed by AND and OR operators to create new offsprings.
5.1.4 Permutation encoding crossover
In permutation encoding crossover, one crossover point is selected. The permutation is copied from first parent chromosome up to the point of crossover and the other parent chromosome is exactly copied to ensure that no number is left to be put in the offspring. Further, if the number is not yet in the offspring, it is added to the offspring chromosome. Travelling salesman problems and task ordering problems can be easily solved by permutation encoding.

5.1.5 Value encoding crossover
It can be performed at single point, two point, uniform and arithmetic representation as in binary encoding technique.

5.1.6 Tree encoding crossover
In this type of crossover, one point of crossover is selected in both parent tree chromosomes, which are divided at a point[15]. The parts of tree below crossover point are exactly exchanged to produce new offsprings. The choice of the type of the crossover is strictly depends upon the problem.

5.2 Mutation
Premature convergence is a critical problem in most optimization techniques, consisting of populations, which occurs when highly fit parent chromosomes in the population breed many similar offsprings in early evolution time. Crossover operation of genetic algorithms (GAs) cannot generate quite different offsprings from their parents because the acquired information is used to crossover the chromosomes. An alternate operator, mutation, can search new areas in contrast to the crossover. Crossover is referred as exploitation operator whereas the mutation is exploration one. Like crossover, mutation can also be performed for all types of encoding techniques.

5.2.1 Binary encoding mutation
In binary encoding mutation, the bits selected for creating new offsprings are inverted, which is illustrated in Figure 5 (a). In binary encoding mutation, if the bit 1 is converted into bit 0, it decreases the numerical value of the chromosome, and is known as down mutation. Similarly, if the bit 0 is converted into bit 1, the numerical value of the chromosome increases and is referred as up mutation.

5.2.2 Permutation encoding mutation
In permutation encoding mutation, the order of the two numbers given in a sequence are exchanged.

5.2.3 Value encoding mutation
In value encoding mutation, a small numerical value is either added or subtracted from the selected values of chromosomes to create new offsprings.

5.2.4 Tree encoding mutation
Tree encoding mutation, mutates the certain selected nodes of the tree to create new offspring.

6. Genetic algorithms (GAs): Issues
Genetic algorithms (GAs) can be applied in complex non-linear process controllers for the optimization of parameters. Some issues are important to be considered for proper implementation of genetic algorithms (GAs) to a plant to be optimized.

V. CONCLUSION & FUTURE SCOPE
In this paper, various genetic algorithms are discussed, moreover Genetic algorithm find optimal solutions among the search space with the operators like crossover and mutation. Genetic algorithms are very effective techniques of quickly finding a reasonable solution to a complex problem[13]-[14]. Most of the researchers used the genetic algorithm to find the frequent itemsets and association rules. However, GA is used for optimization in our future research it has been proposed to use GA to compare the efficiency of various genetic techniques in large datasets. optimize the large input dataset.

REFERENCES


[16] www.elearning.najah.edu/OldData/pdfs/Genetic.ppt