International Journal of Research in Advent Technology, Vol.3, No.2, February 2015
E-ISSN: 2321-9637

Design and Development of an Optimized Approach Based on Genetic Algorithm for Predicting Best Product

Sagar S. Lawade [1], Gayatri S. Hedau[2], Apurva C. Ramgirwar[3],
Prof. S. V. Solanki[4]
B.E. 4th Year, Department of Computer Technology [1, 2, 3]
[4]Asst. Prof. Department of Information Technology,
K.D.K. College of Engineering, Nagpur [1, 2, 3, 4]
Email: sagarlawade@gmail.com[1], gayatri.hedau@gmail.com[2], apurvaramgirwar6@gmail.com[3], svsolankivnit@gmail.com[4]

Abstract- The design of an automatic system using method of optimization has been carried out with deterministic method. This method of Genetic Algorithm aims to derive the best product with desired features. In the genetic algorithm a solution, i.e., a point in the search space is represented by a finite sequence of zero's and ones, called a chromosome. The paper discusses implementation of a strategy to predict the best candidate (of chromosome) among any number of candidates using Genetic Algorithm. This problem belongs to the category of NP-Hard problem in which complexity increases with the increase in the problem size.

Index Terms: - Optimization, Genetic algorithms, NP-Hard Problem, Chromosome.

1. INTRODUCTION

Genetic algorithm is one of the heuristic search techniques. Heuristic search technique provides a useful solution for the problems like optimization and search. Genetic algorithm works on two types of spaces mainly called as a genotype space and phenotype space. Genotype space is working with the operator crossover and mutation while phenotype operator works with evolution and selection operator of GA. It is considered as the possibility of the solution to the problem may be represented as a set of characteristic. Characteristic is called as genes. This characteristics are combined together to form stings of values called as chromosome. Each element in chromosome is called as gene. Let us discuss the working of the Genetic algorithm briefly.

1.1. *Genetic algorithm:-*

Genetic Algorithms are powerful and widely applicable stochastic search and optimization methods based on the concepts of natural selection and natural evaluation. GAs are applied for those problems which either cannot be formulated in exact and accurate mathematical forms and may contain noisy or irregular data or it take so much time to solve or it is simply impossible to solve by the traditional computational methods.
How Genetic algorithm works:-

The General structure of the genetic algorithm is as follows:

```
Begin
{       t=0,
        Initialize p(t);   //population initialization

     Evaluate p(t);
    While (true)
     Begin
     { Apply crossover and mutation to p(t) to get c(t)
                Find c(t)  // offspring
                Select t=t+1 from p(t) and c(t)
}
        End
}
```

This algorithm explain the flowchart shown in Fig(1).
Flowchart:-

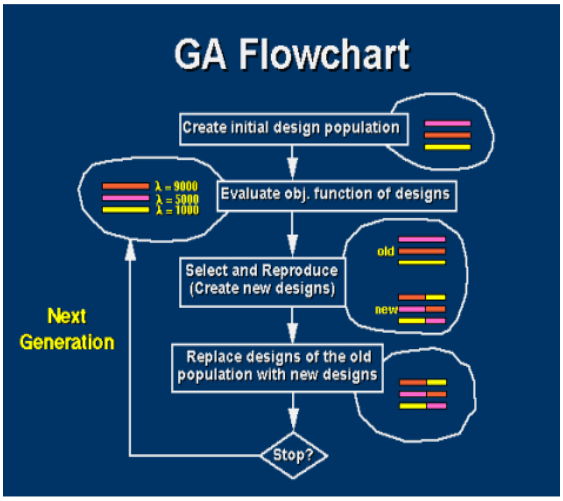

Fig.1

2. LITERATURE REVIEW

Seonho Kim and Jon B. Weissman[1] introduces a novel Genetic Algorithm based strategy that decomposes data onto communication and computation resources. The introduced GA-based scheduler gets benefit of the parallelism of decomposable Data Grid function to accomplish the required performance stage. GA is able to narrowing the search area around the required decision in a short time. However, because of stochastic nature of search strategy, completing the task can take considerable amount of time. Moreover, in scheduling tasks the initial information is represented as sets of discrete elements, which are connected with each other in non-trivial way. A number of studies have been devoted to methods of increasing of GA efficiency.

Paper [2] represents the actual implementation of Genetic Algorithm in the process of optimization. The methodology of finding out the best candidate or student from any number of students ignites the idea of implementation. It implements each step of GA very efficiently. The random features of random population are generated. On this the GA is applied which enables us to come out with the student who is best among the population.

Paper [3] represents combinations of GA and "traditional" search techniques (e.g. a method of gradient searches, multidimensional minimization and so on). GAN Guoning,

HUANG Ting-Iei, GAO Shuai [4] discusses an algorithm for task scheduling based on genetic simulated annealing algorithm for cloud computing environment and its implementation are also given in the paper. This optimized algorithm considers different QOS requirements for different type tasks. Paper [5] discusses an optimized algorithm based on genetic algorithm to schedule independent and divisible tasks adapting to different computation and memory requirements. Though GA is designed to solve combinatorial optimization problem, it's inefficient for global optimization. So researches in optimized genetic algorithm are also proposed.

.

3. PREDICTION STRATEGY IMPLEMENTATION

The construction of a genetic algorithm for the prediction problem can be divided into four parts: The choice of representation of individual (candidate) in the population; the calculation of the fitness function; the design of genetic operators.

The following outline summarizes how the genetic algorithm works:-

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generations to create the next population. To create the new population, the algorithm performs the following steps:

    1. Scores each member of the current population by computing its fitness value.
    2. Scales the raw fitness scores to convert them into a more usable range of values.
    3. Selects members, called parents, based on their fitness.
    4. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
    5. Produces children from the parents. Children are produced either by making random changes to a single parent—*mutation*—or by combining the vector entries of a pair of parents—*crossover*.
    6. Replaces the current population with the children to form the next generation.

The algorithm stops when one of the stopping criteria is met.

3.1 Population initialization:-

Population initialization is a dynamically predict solution within a minimum amount of time. For the implementation of genetic algorithm population initialization is the basic need and it directly influence the performance of the algorithm. In genetic algorithm technique, every solution is encoded as a chromosome having n genes which is nothing but the chromosome length. For the implementation, a chromosome may be represented by its features (genes):-

(1)

{"Engine displacement", "Engine power", "Gearbox", "Seating capacity", "Fuel tank capacity", "Mileage", "Boot space, Tyres"}

These gene values are assigned to the chromosomes.

3.2 Fitness value:-

Each individual combined with the fitness value. To find the optimal fitness value is the main goal of GA. By adding values of the genes, the fitness of individual is calculated. For this implementation we applied following rules.

```
int I;
sum=0;
```

```
For(I = 0; I < gene.length; I++)
        Sum = sum + gene[I];
```

The fitness value of the following chromosome is calculated as-

{0.217, 0.54, 0.64, 0.5, 0.4, 0.7, 0.503, 0.314}

This values corresponds to the genes

{“Engine displacement”, “Engine power”, “Gearbox”, “Seating capacity”, “Fuel tank capacity”, “Mileage”, “Boot space”, “Tyres”}

Fitness value = 3.814.

3.3 Crossover to generate offspring:-

Crossover operation is carried out between the two chromosomes. A random pair is selected of the chromosome and the random point is chosen for crossover operation using Roulette Wheel selection operator. The crossover is a process by which a string is divided into segments, which are exchanged with the segments corresponding to another string. With these process two new strings different to those that produced them are generated. It is necessary to clarify that the choice of strings crossed inside those that were chosen previously in the reproduction process is random. A crossover operator is used to recombine two strings to obtain a better string. In crossover operation, recombination procedure makes different individuals in the consecutive generations by combining material from two individuals of the previous generation. In selection of reproduction, superior strings in a population are probabilistically allocated a larger number of copies. It is vital to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are formed by exchanging information among strings of the mating pool. The two strings taking part in the crossover operation are identified as parent strings and the resulting strings are recognized as children strings. Consider example given below-

C1 {**0.217, 0.54, 0.64, 0.5**, 0.4, 0.7, 0.503,0.314}
C2 {0.181, 0.66, 0.5, 0.5, **0.37, 0.716, 0.5, 0.315**}
The crossover point is denoted by underlining the gene value. So the new offspring is as
C3 {0.217, 0.54, 0.64, 0.5, 0.37, 0.716, 0.5, 0.315}

3.4 Mutation:-

Mutation is a genetic operator that changes the gene sequence by altering one or more gene values in a chromosome from its initial state to reproduction state. This may have the probability of resulting in entirely new gene values being added to the gene mating pool. With these new gene values, the genetic algorithm may be able to arrive at better solution. Mutation adds new information in a arbitrary way to the genetic search procedure and ultimately supports to avoid getting trapped at local optima. It is an operator that initiates diversity in the population whenever the population tends to become uniform due to repeated utilization of reproduction and crossover operators. Mutation may cause the chromosomes of individuals to be diverse from those of their parent individuals.

3.5 Select parents and offspring to form the new population for the next generation:-

At last, the chromosomes from this customized population are evaluated again. This finishes one iteration of the GA. The GA discontinues when a predefined number of evolutions are reached or all chromosomes. At the end best solution is achieved after number of evolutions.

4. IMPLEMENTATION RESULTS

Prediction problem using GA can used for problem-solving and for modeling. The implemented prediction approach is much generalized that can fit into any real time problem. The generalized problem considers car database having any number of cars of different features. The prediction problem finds out the best car with desired features among any number of cars using genetic algorithm. This problem belongs to NP Hard problem as complexity increases with increase in car count.

Implementation 1:

Figure 2: Initial screen showing generalized features of individual at the left side and number of individuals, number of iterations and mutation index at right side.

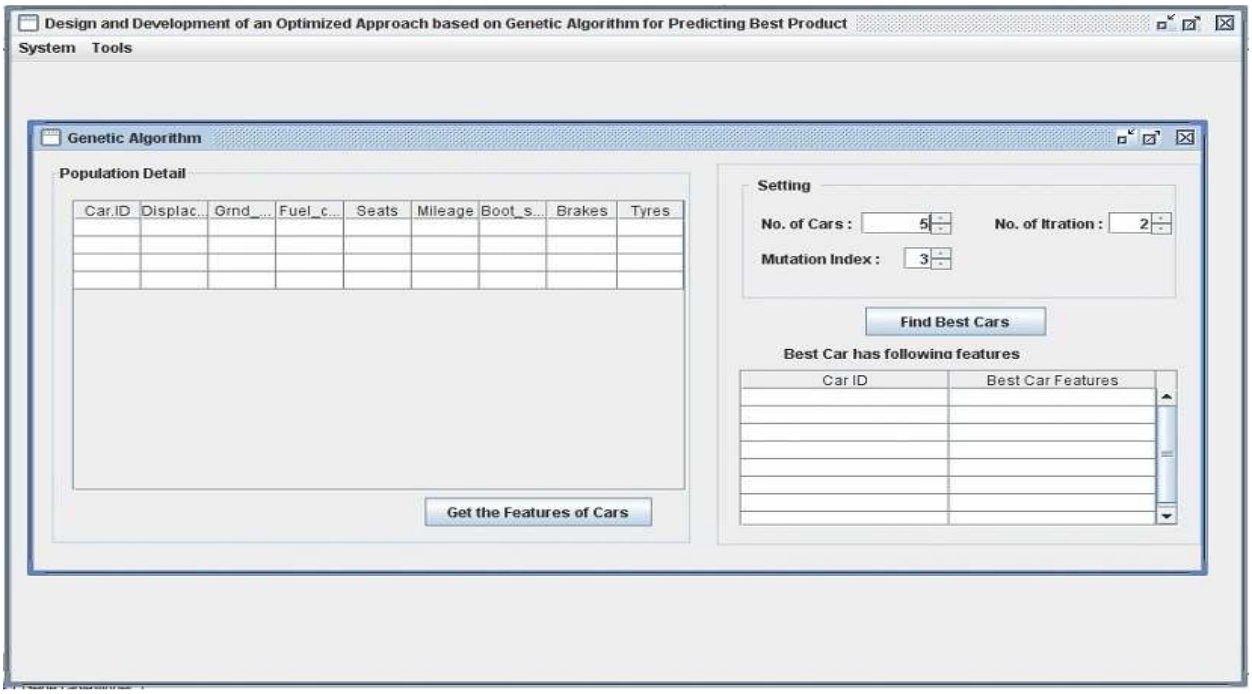

Fig. 2

Implementation 2:

1

Figure 3: It depicts the corresponding gene values or the feature values of cars that has the desired feature types.

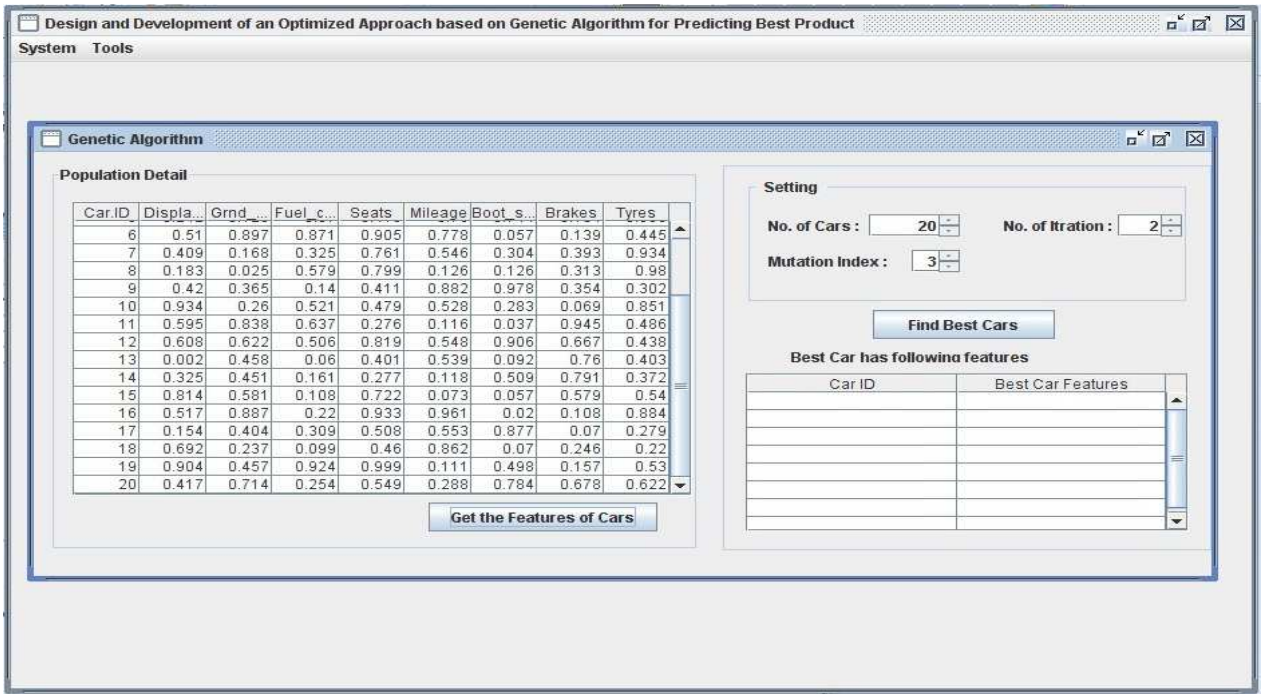

Fig. 3

5. CONCLUSION

Genetic Algorithms can be used to various real time problems, from optimization problems to inductive concept learning, scheduling, and layout problems. The results can vary from problem to problem. These applications, be they
profitable, educational and scientific, are increasingly reliant on this algorithms, the Genetic Algorithms. The progress of the project is in some steps. The database of the cars data is ready. It includes various features of the car and their corresponding values in required genotype range. The system implements the few steps of Genetic Algorithm. The population has the car identification using Car_ID followed by the car features. The features have the gene values. These values are useful for further evaluation steps of Genetic Algorithm. In this work, we are pondering upon explaining development of an optimized approach for predicting the best product using GA and different implementations are also presented.

REFERENCES

[1] Seonho Kim and Jon B. Weissman, "A GA-based Approach for Scheduling Decomposable Data Grid Applications."
[2] Sachin V. Solanki and Dr. A. R. Mahajan, "An Optimized Strategy Based on Genetic Algorithm for Predicting Best Candidate."
[3] D. Bogdanovich, A. Ziyangirov, "Genetic algorithm structure optimization for the schedule task solution", IEEE 2011.
[4] GAN Guo-ning, HUANG Ting-Iei, GAO Shuai," Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment" IEEE 2010, pp 60-63.
[5] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu," Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing" 2009 IEEE.