Study of Different Selection Strategies for Implementing Optimization Problem

Pooja[1] Bhumika Garg[2]
[1]M.Tech Student [2]Assistant Professor
[1,2]Department of Computer Science& Engineering
[1,2]Modern Institute of Engineering and Technology, Mohri, Haryana, India

Abstract— Genetic algorithm is a search technique used in computing to generate useful true or approximate solutions to optimization and search problems like travelling salesman problem, Dejong's function, job scheduling problems. Selection phase play an important role in genetic algorithm to get better solutions. Selection is the phase of genetic algorithm in which we select the individual (chromosome) based upon the fitness value. The goodness of each individual depand on its fitness. Fitness can be determined by an objective function. Different selection methods are applied on these objective functions to determine the fitness of each individual. Initialization, selection, crossover and mutation are four main operation of GA. The aim of this paper is to study and compare different types of selection methods used in Genetic algorithm.

Key words: Genetic Algorithm (GA), Fitness, Roulette Wheel Selection, Rank Selection, Tournament Selection, Elitism Selection, Hybrid selection

I. INTRODUCTION

The theory of genetic algorithms (GAs) was originally developed by John Holland in 1960 and was fully developed in his book " Adaptation in Natural and Artificial Systems ", published in 1975 . His initial goal was not to develop an optimization algorithm , but rather to model the adaptation process, and show how this process could be useful in a computing system. The GAs are stochastic search methods using the concepts of Mendelian genetics and Darwinian evolution. Such heuristics have been proved effective in solving a variety of hard real-world problems in many application domains including economics, engineering, manufacturing, bioinformatics, medicine, computational science, etc [1]. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Genetic algorithms (GAs) are subclass of evolutionary algorithms (EAs) where the elements of the search space are binary strings or arrays of other elementary types. Genetic algorithms (GAs) are computer based search techniques patterned after the genetic mechanisms of biological organisms that have adapted and flourished in changing highly competitive environment [2].. Genetic Algorithms [3] are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and natural genetics. Genetic algorithms are categorized under evolutionary algorithms. They are mainly used for solving optimization problems. Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. Likewise, Genetic algorithms work on coding space and solution space alternatively. In nature, coding space refers to the genotypic space and solution space refers to phenotypic space. The genotype represents all the information stored in the chromosomes and allows us to describe an individual on the level of genes [3]. The phenotype describes the outward appearance of an individual. A transformation exists between genotype and phenotype, also called mapping, which uses the genotypic information to construct the phenotype. A chromosome refers to a string of certain length where all the genetic information of an individual is stored. Genetic algorithm is a part of evolutionary algorithm. In genetic algorithm [4] for representation, we can use the fixed length bits strings. Genetic algorithm is also dependent on the fitness function. The fitness functions find out the quality of solution such as measurable and returning the fitness value. They perform very well for large scale optimization problem. The main reason to use the genetic algorithm because there are multiple local optima.

II. BASIC PROCESS FORGENETIC ALGORITHM

Genetic algorithms (GAs) can be applied to any process control application for optimization of different parameters. GAs are population based search techniques that maintain populations of potential solutions during searches [5]. A potential solution is represented by a string with a fixed bit-length. Objective functions are used to evaluate these potential solutions. GA starts searching after the evolution function and representation scheme is determined. 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.

The workability of genetic algorithms (GAs) [2] is based on Darwinian's theory of survival of the fittest. Genetic algorithms (GAs) may contain a chromosome, a gene, set of population, fitness, fitness function, breeding, mutation and selection. Genetic algorithms (GAs) begin with a group of individuals known as population. After that an individual is selected from the population and used to form a new population, expecting that the new population will be better than the old one. Solutions are selected according to their fitness to form new solutions, that is, offsprings [2]. The above process is repeated until some condition is satisfied. Algorithmically, the basic genetic algorithm (GAs) is outlined as below:

A. *Initialization:*

Create an initial population. This population is usually randomly generated and can be any desired size, from only a few individuals to thousands.

B. *Evaluation:*

Each member of the population is evaluated. Then we calculate a 'fitness' value for that individual. The fitness value is calculated by how well it fits with our desired requirements. These requirements could be simple, 'faster algorithms are better', or more complex, 'stronger materials are better but they shouldn't be too heavy'.

C. *Selection:*

In selection, we select the individual based upon the fitness value. There are a few different selection methods but the basic idea is the same, make it more likely that fitter individuals will be selected for our next generation. Selection phase discard the bad designs and only keep the best individuals in the population. Using selection we can constantly improve our populations overall fitness

D. *Crossover:*

During crossover we create new individuals by combining aspects of our selected individuals. In crossover by combining certain traits from two or more individuals we will create an even 'fitter' offspring which will inherit the best traits from each of its parents.

E. *Mutation:*

We need to add a little bit randomness into our populations' genetics otherwise every combination of solutions we can create would be in our initial population. Mutation typically works by making very small changes at random to an individual's genome.

F. *And Repeat:*

Now we have our next generation we can start again from step two until we reach a termination condition.

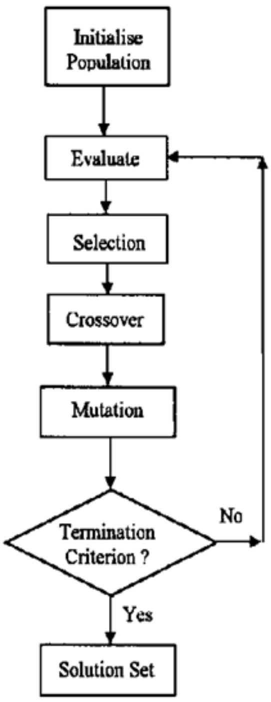

Fig. 1: Genetic algorithm process

G. *Termination:*

There are a few reasons why you would want to terminate your genetic algorithm from continuing it's search for a solution. The most likely reason is that your algorithm has found a solution which is good enough and meets predefined minimum criteria. Offer reasons for terminating could be constraints such as time or money.

III. SELECTION METHODS

The selection phase [5] plays an important role in genetic algorithm to find the better individuals and in maintaining a high genotypic diversity in the population. In selection phase, we will choose the individuals in the population that will create offspring for the next generation. The purpose of selection is, to emphasize the fitter individuals in the population in hopes that their offspring will in turn have even higher fitness. Selection has to be balanced with variation from crossover and mutation. A generic selection procedure may be implemented as follows [7]:

1) The fitness function is evaluated for each individual, providing fitness values. These fitness values are then normalized. Normalization means dividing the fitness value of each individual by the sum of all fitness values, so that the sum of all resulting fitness values equals 1.
2) The population is sorted by descending fitness values.
3) Accumulated normalized fitness values are computed. The accumulated fitness of the last individual should be 1 otherwise something went wrong in the normalization step.
4) A random number R between 0 and 1 is chosen.
5) First selected individual is the one whose accumulated normalized value is greater than R.

There are several classification criteria[5] of selection schemes :

A. *Extinctive Versus Preservative Selection:*

The term preservative describes a selection scheme that guarantees a non-zero selection probability for each individual, i.e.; every individual has a chance of contributing offspring to the next generation. On the other hand, in an extinctive selection scheme some individuals are definitely not allowed to create any offspring, i.e., they have zero selection probabilities.

B. *Left versus Right Extinctive Selection:*

In case of extinctive selection there is a major special case where the worst performing individuals have zero reproduction rates, i.e. they do not reproduce. This situation is referred to as right extinctive selection. Similarly, the best individuals are also prevented from reproducing in order to avoid premature convergence due to the existence of super-individuals. This is referred to as left extinctive selection.

C. *Elitist Versus Pure:*

A selection scheme that enforces a lifetime of just one generation on each individual regardless of its fitness is referred to as pure selection. In an elitist selection scheme some parents are allowed to undergo selection with their offspring.

IV. TYPES OF SELECTION STRATEGIES

A. *Roulette Wheel Selection:*

Holland's original GA used fitness−proportionate selection [8] .The most common method for implementing this is "roulette wheel" .In this method every individual gives the slice of roulette wheel and the sizes of slice are directly proportional to the each individual fitness. . The wheel is spun *N* times, where *N* is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. Now this process is repeated until the desired number of individual is selected. If fitness is higher, the segment size is larger. [2] The fig 2: represents the Roulette Wheel 5 individuals with fitness values. Because the third individual gives the highest fitness values than any other individual in roulette wheel. so it will chooses the third individual than other. It is also called fitness-proportionate selection.

Roulette Wheel Selection [6] can be worked as in the following steps:

1) Step 1: Find the fitness value (fv) for each chromosome in the population using Fitness Function.
2) Step 2: Calculate sum fitness (Sf) for all the chromosomes in the population:

$$\mathbf{Sf} = \sum_{i=1}^{n} \mathbf{fv}\,i$$

Population	Fitness
1	25.0
2	5.0
3	40.0
4	10.0
5	20.0

Fig. 2: Roulette wheel selection for five individuals.

3) Step 3:Calculate average fitness (Af) in the Population:

$$\mathbf{Af} = \frac{\mathbf{Sf}}{\mathbf{n}}$$

4) Step 4:Find the expected fitness (Ef) for each chromosome in the population:

$$\mathbf{Ef}\,i = \frac{\mathbf{fv}\,i}{\mathbf{Af}}$$

5) Step 5:Calculate sum expected fitness (Sum Ef) for all the chromosomes in the population:

$$\mathbf{SumEf} = \sum_{i=1}^{n} \mathbf{Ef}\,is$$

6) Step 6:Generate random number (G) in the range [0,SumEf]:

$$\mathbf{G} = \mathbf{Rnd}()\,mod\,\mathbf{SumEf}$$

7) Step 7: Select the chromosome that added its fitness value to the previous chromosomes fitness value's to make (SumEf > = G).
8) Step 8: Go to step 6, repeat n times, where n is a population size.

1) *Advantages:*

1) It always gives a chance all of them to be selected.

2) *Disadvantages:*

1) Roulette wheel selection method cannot be used on minimization problems.
2) As population converges it loss of selection pressure.
3) Time complexity: O (n2)

B. *Rank Selection:*

Rank selection is used to solve the problem of roulette wheel. The application of Roulette wheel selection method is not satisfactory in genetic algorithms (GAs), if there is a lots of difference between the fitness value of chromosomes .It is a slower convergence technique. Rank selection is used to prevent quick convergence.[8] In rank selection first of all rank the population and after that from the ranking every genes receives the fitness. In rank selection we can sort from best to worst. The worst fitness has rank 1, next has rank 2 and the best rank has N, where N is no. of population. The rank selection method is shown in Figure. 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.

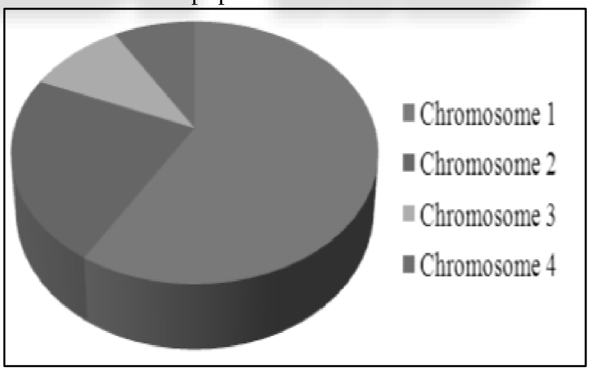

Fig. 3: Rank selection method

This can be worked as in the following steps:

1) Step1: Arrange the chromosomes in decreasing order according to its fitness values.
2) Step 2: Assign a rank value to each chromosome according to its arrangement in the set.
3) Step3: Calculate the new fitness value for each chromosome using the following equation [9]:

Pi = 1 / N * (Pworst + (Pbest - Pworst) *((I-1) / (N-1)))

Here, Pworst => Worst Fit Individual

Pbest => Best Fit Individual

N => Size of Population

I => Population Individual

1) *Advantages:*

1) To prevent too quick convergence

2) Disadvantages:

1) Populations must be sort on every cycle.
2) Time complexity: O (n ln n) +time of selection.

C. Elitism Selection:

"Elitism," first introduced by Kenneth DeJong (1975)[8]. Retaining the best individuals in a generation unchanged in the next generation, is called elitism or elitist selection [7].Elitism selection is a strategy which guarantees that the solution quality obtained by genetic algorithm will not decreases from one generation to next. In this method [6], the chromosome are arranged in the decreasing order. After that selection is applied with every two chromosomes in arrange set. In elitism selection we can copy the first best chromosomes or the few best chromosomes to the new population. There is no need of any modification we can pass the best individual to next generation. Elitism significantly improves the GA's performance.

1) Advantages:

1) This significantly improves the GA's performance because elitism generates very fit population generations.
2) Time complexity: O (MN^2)

D. Boltzmann Selection:

"Boltzmann selection" (an approach similar to simulated annealing), in which a continuously varying "temperature" controls the rate of selection according to a preset schedule. The temperature starts out high, which means that selection pressure is low (i.e., every individual has some reasonable probability of reproducing). [8] The temperature is gradually lowered, which gradually increases the selection pressure, thereby allowing the GA to narrow in ever more closely to the best part of the search space while maintaining the "appropriate" degree of diversity.

E. Tournament Selection:

In tournament selection various tour of a few individual is selected at randomly from population and selected best individual as a parent. We select best fitness values with winner of each tournament. When the tournament size is changed we can easily adjusted selection pressure. Tournament selection is similar to rank selection in terms of selection pressure, but it is computationally more efficient and more amenable to parallel implementation. [8]Two individuals are chosen at random from the population. A random number r is then chosen between 0 and 1. If $r < k$ (where k is a parameter, for example 0.75), the fitter of the two individuals is selected to be a parent; otherwise the less fit individual is selected.

1) Advantages:

1) When implementing in parallel it has efficient time complexity.
2) Fitness scaling or sorting not required.

2) Disadvantages:

1) This type of selection does not give guarantee to reproduction of best solution.
2) Time complexity: O (n)

F. 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. Steady−state GAs are often used in evolving rule−based systems (e.g., classifier systems; see Holland 1986) in which incremental learning (and remembering what has already been learned) is important and in which members of the population collectively (rather than individually) solve the problem at hand.[8]

G. Truncation Selection:

In truncation selection individuals are sorted according to their fitness. Only the best individual are selected for parents. These selected parents produce uniform at random offsprings. The parameters for truncation selection is the theroshold Trunc. Trunc indicates the proportion of the population to be selected as parents and takes values ranging from 50% to 10%. Individuals below the truncation theroshold do not produce offsprings. Taking the best half, third or another proportion of the individuals is truncation selection .

H. Hybrid Selection Method:

Our idea lies in hybrid selection by embedding local search operations into the simple selection methods. We can use local search in Roulette Wheel Selection, Rank Selection, Tournament Selection, Elitism Selection etc to form hybrid selection. [10] The goal of hybrid selection is to improve the performance of genetic algorithm. In hybrid selection, first parent is selected by any suitable selection method and the second by common means. It means that the selection method always picks one member of the class of best ranked chromosomes from the population, which gives the best solutions and second parent is chosen from whole population (possibly but not necessarily also belonging to the class of best ranked individuals).The local search operations gives a solution towards a local optimum, and these improvments over the entire generations result in a performance improvement in the final. In a hybrid selection quality of selected chromosomes are improved.

V. CONCLUSION

Genetic algorithms can be applied to areas in which insufficient knowledge of the system and high complexity is there. Genetic algorithms can 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. This paper presents the different type of selection methods used in the genetic algorithm. In Roulette wheel selection the area covered by the entire chromosome in a population as per the fitness value. Now this process is repeated until the desired number of individual is selected. In rank selection first of all rank the population and after that from the ranking every genes receives the fitness. In Tournament Selection We select best fitness values with winner of each tournament. When the tournament size is changed we can easily adjusted selection pressure. In elitism selection we can copy the first best chromosomes or the few best chromosomes to the new population. Hybrid selection is used to get optimal solutions. A perfect blending of genetic algorithm with local search techniques helps in exploitation as well as exploration.

REFERENCES

[1] Khalid Jebari, Mohammed Madiafi , "Selection Methods for Genetic Algorithms", Int. J. Emerg. Sci., vol. 3, No 4, pp 333-344, Issue: December 2013, ISSN: 2222-4254.
[2] Rahul Malhotra, Narinder Singh & Yaduvir Singh,"Genetic Algorithms: Concepts, Design for Optimization of Process Controllers", Computer and Information Science, Vol. 4, No. 2, Issue : March 2011,ISSN 1913-8989 E-ISSN 1913-8997
[3] Rakesh Kumar, Jyotishree, "Novel Encoding Scheme in Genetic Algorithms for Better Fitness", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-6, August 2012.4)
[4] Poonam Sharma, Amit Wadhwa, Komal,"Analysis of Selection Schemes for Solving an Optimization Problem in Genetic Algorithm",International Journal of Computer Applications (0975 – 8887) Volume 93 – No.11, Issue- May 2014.
[5] Ying-Hong Liao, and Chuen-Tsai Sun, "An Educational Genetic Algorithms Learning Tool", Member, IEEE.
[6] Firas Alabsi, Reyadh Naoum, " Comparison of Selection Methods and Crossover Operations using Steady State Genetic Based Intrusion Detection System", Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO.7, Issue-July 2012, ISSN 2079-8407.
[7] ://en.m.wikipedia.org/wiki/Selection_(genetic_algorithm)
[8] Mitchell Melanie," An Introduction to Genetic Algorithms", A Bradford Book The MIT Press, Cambridge, Massachusetts London, England . Fifth printing, 1999, First MIT Press paperback edition, 1998.
[9] Kapil Juneja, Nasib Singh Gill, " Optimization of Dejong Function using GA under Different Selection Algorithms", Department of Computer Science & Applications, International Journal of Computer Applications (0975 – 8887) Volume 64– No.7, February 2013.
[10] I1-Seok Oh, Jin-Seon Lee, Byung-Ro Moon, "Local Search embedded Genetic Algorithms for Feature Selection", Department of Computer Science & Applications , CNU, 10514651/02$17.00(c)2002IEEE.