Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 3, March 2019, pg.335 – 344

A Study of Genetic Algorithm and Crossover Techniques

Ashima Malik
Assistant Professor, Computer Science Dept., Dronacharya College of Engineering, Gurugram, India
E-mail: ashima.13malik@gmail.com

ABSTRACT: Genetic algorithms are inspired by Darwin's theory of natural evolution. In the natural world, organisms who are poorly suited for the environment die off, while those well suited, prosper. Genetic algorithms search the space of individuals for good candidates. The chance of particular individual being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. Genetic algorithms are ways of solving problems by mimicking processes nature uses; i.e. Selection, Crossover, Mutation and accepting, to evolve a solution to a problem. Many crossover techniques exist for organism which uses different data structures to store themselves. Genetic algorithm which is one of the most well-known heuristic approaches, crossover components and crossover techniques, which are the most important property of the Genetic algorithms performance, has been discussed.

Keywords: Genetic algorithm; encoding; crossover; mutation.

I. INTRODUCTION

Genetics is the study of heredity, the process in which a parent passes certain genes onto their offspring. It is the process of trait inheritance from parents to children including the molecular structure and function of genes, such as some physical characteristics, natural talents, and genetic disorders. The modern science of genetics, seeking to understand this genetic process began with the work of Gregor Mendel in the middle of 19th century [3]. Mendel, a German-Czech Augustinian monk and scientist who studied the nature of inheritance in plants. The traits of an organism are expressed by genes which are small sections of DNA that are coded for specific traits. Genes are found on chromosomes. Each organism has two alleles that make up the genotype for a specific given trait. In sexual reproduction, each parent contributes only one pair of their alleles to its offspring. In each genotype there is a dominant allele and if in an organism it exists then the phenotype is determined by that allele.

II. GENETIC ALGORITHM

Genetic algorithm is basically inspired by Darwin's theory of natural evolution - "the survival of the fittest". Genetic algorithms are adaptive heuristic search algorithm based on the evolutionary ideas natural selection and genetics. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and genetics to imitate living beings for solving difficult problems with high complexity and undesirable structure. Genetic algorithms are able to evolve solutions to real world problems, if they have been suitably encoded. For example, Genetic algorithms can be used to design bridge structures, for online process control, such as in a chemical plant, or load balancing on a multi-processor computer system. The basic principles of Genetic algorithms were first laid down by Holland [1]. General structure of genetic algorithm

Procedure of Genetic algorithm:

A. [Start] Generate random population of n chromosomes.

B. [Fitness] Evaluate the fitness f(x) of each of the chromosome x in the population.

C. [New population] Create new population by repeating the following steps until new population is complete.

    1) [Selection] Select two parent chromosomes from a population according to their fitness.

    2) [Crossover] With a crossover probability, crossover the parents to form new offspring. If no crossover is performed offspring is the exact copy of the parent.

    3) [Mutation] With a mutation probability, mutate new offspring at each position.

    4) [Accepting] Place this new offspring in the population.

D. [Replace] Use this new generated population for further processing of the algorithm.

E. [Test] If the end condition is satisfied, stop and return the best solution in current population.

F. [Loop] go to step2.

Mutation and crossover are the two most important operators of Genetic algorithm. The performance of algorithm is largely influenced by these two operators.

Termination of loop is when we reach some fitness, reaching some maximum number of generations, reaching some minimum level of diversity or reaching some generations without fitness improvement.

ADVANTAGES OF GENETIC ALGORITHM

A. Genetic algorithms do not have much mathematical requirements about the optimization problems [5].

B. NP-Complete problem can be solved with it in an efficient way.

C. It is an effective way to find a reasonable solution to a complex problem easily and quickly.

D. Parallelism and easy implementation is also one advantage.

DISADVANTAGE OF GENETIC ALGORITHM

1. The only disadvantage is that they give a poor performance on some problems as might be expected from knowledge poor approaches.

FLOWCHART OF GENETIC ALGORITHM

Flowchart of genetic algorithm is shown in Figure1.

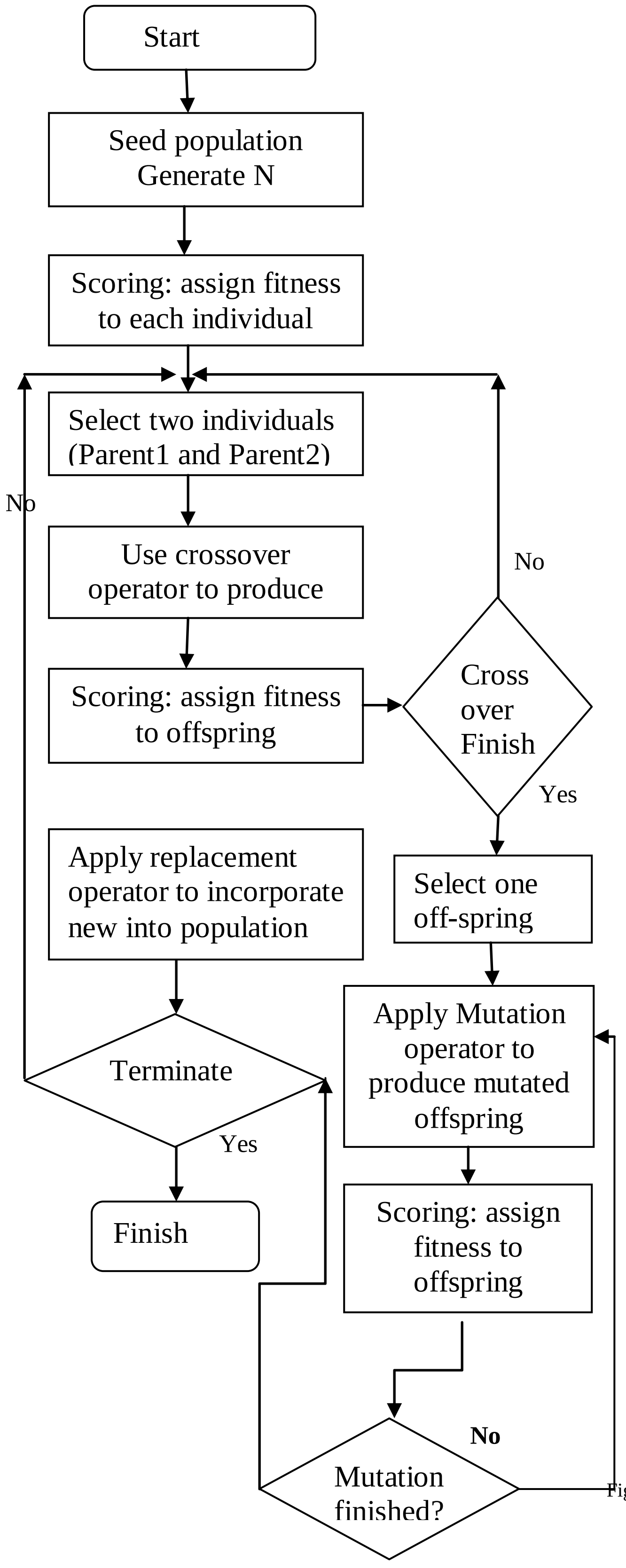
Start
Seed population Generate N
Scoring: assign fitness to each individual
Select two individuals (Parent1 and Parent2)
No
Use crossover operator to produce
Scoring: assign fitness to offspring
No
Cross over Finish
Yes
Select one off-spring
Apply replacement operator to incorporate new into population
Apply Mutation operator to produce mutated offspring
Terminate
Yes
Finish
Scoring: assign fitness to offspring
No
Mutation finished?

Figure1. Flowchart of Genetic Algorithm

III. ENCODING/DECODING

Before a genetic algorithm can be applied to solve any problem, a method is needed to encode potential solutions to that problem in a form so that computer can process.

A. Binary encoding

In genetic algorithms binary encoding is the most common way to represent information contained. It was the first used encoding because of its simplicity. In binary encoding each chromosome is a string of bits: 0 or 1.

B. Value encoding

The value encoding is used in problems where values such as real numbers are used. In value encoding, every chromosome is a sequence of some values. The values can be anything such as: real numbers, characters or objects.

C. Permutation encoding

Permutation encoding can be used in ordering problems such as travelling salesman problem. In permutation encoding, every chromosome is a string of numbers that represent a position in a sequence.

D. Tree encoding

Tree encoding is used mainly for evolving programs or expressions. In tree encoding, every chromosome is a tree of some objects, such as functions or commands. Tree encoding is useful for evolving programs that can be encoded in trees.

IV. OPERATORS OF GENETIC ALGORITHM

Genetic operators used in genetic algorithms maintain genetic diversity. Genetic operators are analogous to those occur in the natural world. Genetic operators are:

A. Reproduction

Reproduction is usually the first operator applied on the population. From the population, the chromosomes are selected to become parents to perform crossover on them and produce offspring. The problem is how to select these chromosomes?

According to Darwin's evolution theory "survival of the fittest"- the best ones should survive and produce offspring. The Reproduction operators are also called Selection operators. Selection means extract a subset of genes from an existing population. Every gene has a meaning, so one can drive from the gene a kind of quality measurement called fitness function [4]. The most commonly used methods of selecting chromosomes from parents to crossover are: Roulette wheel selection, Boltzmann selection, Tournament selection, Rank selection and Steady state selection.

Example of selection

Evolutionary algorithm is to maximize the function $f(x) = x^2$ with the x in the integer interval [0, 31], i.e. x=0, 1, 2…31

The first step is encoding of chromosomes by using binary representation for integers; 5 bits are used to represent up to 31. Assume population size is 4. Generate initial population at random. They are chromosomes or genotypes as 01101, 11000, 01000, and 10011.

Calculate fitness value for each individual.

- Decode the individual into an integer. 01101 $\leftarrow$ 13; 11000 $\leftarrow$ 24; 01000 $\leftarrow$ 8; 10011 $\leftarrow$ 19
- Evaluate the fitness according to $f(x)= x^2$ 13 $\leftarrow$ 169; 24 $\leftarrow$ 576; 8 $\leftarrow$ 64; 19 $\leftarrow$ 361

Select two individuals called parents for crossover based on their fitness in pi. If roulette wheel selection is used then the probability of the ith string in the population is:

$$p_i = F_i \,/\, \left(\sum_{j=1}^{n} F_j\right)$$

where

Fi is fitness for the string I in the population expressed as f(x)

pi is the probability of the string I being selected

n is the number of individuals in the population, is population size, n=4

n*pi is the expected count

Table 2

String no	Initial population	X value	Fitness Fi $F(x)=x^2$	pi	Expected count n*prob i
1	01101	13	169	0.14	0.58
2	11000	24	576	0.49	1.97
3	01000	8	64	0.06	0.22
4	10011	19	361	0.31	1.23
Sum			1170	1.00	4.00
Average			293	0.25	1.00
Max			576	0.49	1.97

The string no. 2 has maximum chance of selection.

B. Crossover

Crossover is a genetic operator that combines two chromosomes to produce a new chromosome. The idea behind it is that the new chromosome may be better than the parent chromosome if it inherits the best characteristics from each of the chromosomes. The crossover operators are of many types:

One simple is One-point crossover

Other types are Uniform, Arithmetic and Heuristic crossovers.

1) One-point crossover

One-point crossover operator randomly selects one crossover point and then copy everything before this point from the first parent and then everything after the crossover point from the second parent.

Consider the two parents selected for crossover

Parent1 11011||00100110111

Parent2 11011||11000011111

Interchanging the parent chromosomes after the crossover points. The offspring produced are:

Offspring1 1101111000011111

Offspring2 1101100100110111

Two crossover operator randomly selects two crossover point

Consider the two parents selected for the crossover

Parent 1 11011||0010011||0111

Parent2   11011||1100001||1111

Interchanging the parent chromosomes between the crossover points

Offspring1 11011||0010011||0111

Offspring2 11011||0010011||0111

2) Uniform crossover

Uniform Crossover operator decides which parent will contribute how the gene value in the offspring chromosomes. The crossover operator allows the parent chromosomes to be mixed at the gene level rather than the segment level.

Consider the two parents selected for crossover

Parent1   1101100100110110

Parent 2 1101111000011110

If the mixing ratio is 0.5% approximately, then half of the genes in the offspring will come from parent1 and half from parent 2.

The possible off springs after crossover is

Offspring1 1101111000011110

Offspring2 1101100100110110

3) Arithmetic crossover

Arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new offspring according to the equations:

Offspring1 = a*Parent1 + (1-a)* Parent2

Offspring2 = (1-a)*Parent1+a* Parent2

Where a is a random weighting factor chosen before each crossover operation.

4) Heuristic crossover

Heuristic crossover operator uses the fitness values of the two parent chromosomes to determine the direction of the search. The offspring are created according to the equations:

Offspring1 = Best parent + r *(Best parent- worst parent)

Offspring2 = Best parent

where r is a random number between 0 and 1.

C. MUTATION

Mutation takes place after the crossover is performed. Mutation occurs during evolution according to a user definable mutation probability, usually set to a low value, say 0.01a good first choice. Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With the new gene values the genetic algorithm may be able to arrive at better solution than what was previously possible [10]. The mutation operators are of many types:

1) *Flip-bit Mutation*

The Flip bit Mutation operator simply inverts the value of the chosen gene. i.e. 0 goes to 1 and 1 goes to 0.

2) *Binary Mutation*

Boundary Mutation operator replaces the value of the chosen gene with either the upper or lower bound for that gene. This mutation operator can only be used for integer and float genes.

3) *Non-uniform Mutation*

Non-uniform Mutation operator increases the probability such that the amount of the mutation will be close to 0 as the generation number increases. This mutation operator can only be used for integer and float genes.

4) *Uniform Mutation*

Uniform Mutation operator replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene. This mutation operator can only be used for integer and float genes.

5) *Guassian Mutation*

The Guassian Mutation operator adds a unit Guassian distributed random value to the chosen gene [2]. The new gene value is clipped if it falls outside of the user specified upper or lower bounds for that gene.

V. DETAILED STUDY OF CROSSOVER TECHNIQUES

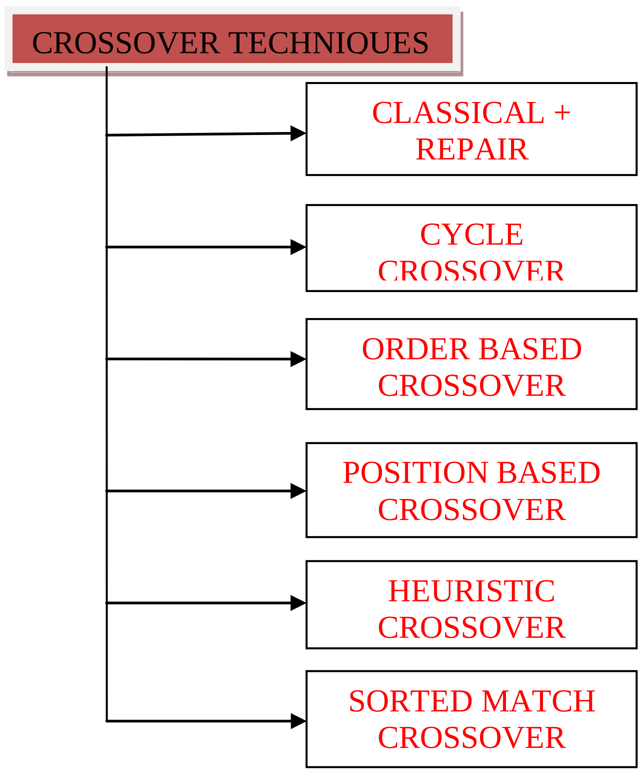

Figure2: Crossover Techniques

A. *Classical Crossover Operator:*

It was proposed by Holland in year 1975. Here strings are broken into separate parts by selecting a crossover point randomly as shown in example.

Example: Consider two strings :

001 100 110 000 111 010

110 001 101 010 101 001

Select a crossover point as:

001 100 110||000 111 010

110 001 101||010 101 001

Then resulted offsprings will be generated by recombining different parts as:

001 100 110 010 101 001

110 001 101 000 111 010

Disadvantage: This method results in illegal tours and repair algorithm can be used to generate legal tours.

B. *Cycle Crossover:*

It was suggested by Oliver et. Al. in the year 1987. It creates an offspring from the parent by occupying every position by a corresponding element from one of the parent.

Example: Consider two parent strings:

(1 2 3 4 5 6 7 8) and (2 4 6 8 7 5 3 1)

⇓

Now for making first part of offspring either first element of first parent tour will be chosen or first element of second parent tour.

(1 * * * * * * *)

Last element is also chosen from one of the parents:

(1 * * * * * * 8)

It is found that the fourth and second element of offspring is selected from the first parent:

(1 2 * 4 * * * 8)

These positions chosen upto now are said to be cycle.

Third element can be chosen from any of the parents. Fifth, sixth, seventh elements of offsprings has to be chosen from second parent as they form second cycle. Hence following offspring is generated:

(1 2 6 4 7 5 3 8)

Advantage:

It gives better results than PMX for TSP.

C. *Order based Crossover (OX2):*

It was proposed by Sysverda in 1991. It selects randomly several positions in a parent tour and the order of cities in selected position of this parent are imposed on other parent. This can be explained with the help of following example:

Consider again the same strings:

(1 2 3 4 5 6 7 8) and (2 4 6 8 7 5 3 1)

Suppose we select position third fourth and fifth of first parent. These positions contain digits 3,4,5. Now for generating offspring we will keep all position digits of string two other than 3, 4, 5 same as:

(2 * 6 8 7 * * 1)

Now the remaining three digits will be written in same order as they are present in first string as:

(2 3 6 8 7 4 5 1)

D. *Position based crossover*:

It was suggested by Sysverda in 1991. Here random positions in parent tour are selected, and positions of selected cities are imposed on the corresponding cities of second parent.

Example:

(1 2 3 4 5 6 7 8) and (2 4 5 6 8 7 5 3 1)

By selecting second, third and sixth positions, following offsprings are generated:

(1 4 6 2 3 5 7 8) and (4 2 3 8 7 6 5 1)

E. Heuristic Crossover:

It was proposed by Grenfenstette in the year of 1987b. he emphasized on edges by developing a class of heuristic crossover operators which creates offsprings in following way:

1. Firstly a city is selected at random to be the current city of offspring.
2. Now four undirected edges incident to the current city are considered. A probability distribution is defined based on cost over these edges.
3. An edge is selected on this distribution (If none of parental edges leads to an unvisited city a random edge is selected).
4. Step 3 and 4 are repeated until a complete tour has been constructed.

F. Sorted match crossover:

It was proposed by Brady in 1985. Here both the parents are searched for those sub tours which have same length, which start in the same city, which end in the same city, and which contain the same set of cities. Then the cost of substring of these sub tours if determined. The offspring of constructed from the parent which contains the sub tour with the highest cost by substituting this sub tour for the sub tour with the lowest cost.

Example: Consider two parent tours:

(1 2 3 4 5 6 7 8) and (3 4 6 5 7 2 8 1)

By applying sorted match crossover, following offspring will be generated:

(1 2 3 4 6 5 7 8)

VI. CONCLUSION

In the computer science field of artificial intelligence, a genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection and crossover. Crossover and mutation are two basic operators of Genetic algorithm. Performance of genetic algorithm very depends on them. Type and implementation of operators depends on encoding. There are many ways how to do crossover and mutation. Genetic algorithms find applications in bioinformatics, computational science, mathematics, physics and other fields.

REFERENCES

1. J. H. Holland. Adaption in Natural and artificial systems. MIT Press, 1975.
2. David E. Goldberg, (1989), Addison-Wesley, “Genetic algorithm in Search, Optimization, and Machine Learning.”
3. Weiling, F (1991). “Historical study: Johann Gregor Mendel 1822-1884.” American journal of medical genetics.
4. K. F. Man, K. S. and Tang, S. Kwong, (201), Springer, “Genetic Algorithm Concepts and Designs.”
5. http://www.ie.ncsu.edu/fangroup/ga.html
6. Mrs. Geetha Ramani.R, Nishaa Bouvanasilan,Vasumathy Seenuvasan, “ A perspective view on Travelling Salesman Problem using Genetic Algorithm”
7. Antonio Augusto Chaves, Luiz Antonio Nogueira Lorena, “Hybrid Algorithms with Detection of Promising Areas for the Prize Collecting Travelling Salesman Problem”
8. Shakeel Arshad, and Shengxiang Yang, Member, IEEE, “A Hybrid Genetic Algorithm and Inver Over Approach for the Travelling Salesman Problem”

9. P. Laranagga, C.M.H Kuijpers, R.H. Murga I Inja and S. Dizderevic," Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators.
10. S. Rajasekaran and G. A. Vijayalaksmi Pai, "Neural Networks, Fuzzy Logic, and Genetic algorithms- Synthesis and Applications"
11. Arindam Chaudhuri and Kajal De, "Fuzzy multi-objective linear programming for traveling salesman problem"
12. Fozia Hanif Khan, Nasiruddin Khan, Syed Inayatulllah, Shaikh Tajuddin Nizami and Muhammad Imtiaz, "Hybrid approach for solving TSP by using DPX Cross-over operator" Pelagia Research Library ISSN: 0976-8610
13. Petrică C. Pop, Serban Iordache, "A Hybrid Heuristic Approach for Solving the Generalized Traveling Salesman Problem"
14. Devasenatipathi N.Mudaliar, Dr. Nilesh K.Modi , " Unraveling Travelling Salesperson Problem By Genetic Algorithm Using Mcrossover -Operator" 2013 International conference on Signal Processing, Image Processing and Pattern Recognition [ICSIPR].
15. Khalid Jebari, Abdelaziz El moujahid1, Abdelaziz Bouroumi, and Aziz Ettouhami, "Unsupervised Fuzzy Clustering-based Genetic Algorithms to Traveling Salesman Problem ".