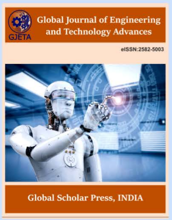
Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/



(RESEARCH ARTICLE)

Genetic algorithm encoding and problems solving for enthusiasts

Whyte Asukwo Akpan [1, *], Godwin Daniel Essien [2], Chikodili Martha Orazulume [3], Francis Etebong Edet [4], Nsikakabasi Nsedu Inyang [5] and Ndifreke Ikemesit Effeng [6]

[1] Department of Mechanical Engineering, School of Engineering and Engineering Technology, Federal University of Technology, Ikot Abasi, Akwa Ibom State, Nigeria.
[2] Department of Computer Engineering, Faculty of Engineering, TopFaith University Mkpatak, Akwa Ibom State, Nigeria.
[3] Department of Electrical and Electronics Engineering, Faculty of Engineering, TopFaith University Mkpatak, Akwa Ibom State, Nigeria.
[4] Department of Mechanical Engineering, Faculty of Engineering, University of Uyo, Akwa Ibom State, Nigeria.
[5] Department of Electrical and Electronics Engineering, School of Engineering and Engineering Technology, Federal University of Technology, Ikot Abasi, Akwa Ibom State, Nigeria.
[6] Department of Mechanical Engineering, Faculty of Engineering, University of Uyo, Akwa Ibom State, Nigeria.

Global Journal of Engineering and Technology Advances, 2025, 24(01), 087-096

Publication history: Received on 07 May 2025; revised on 08 July 2025; accepted on 11 July 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.24.1.0203

Abstract

Genetic algorithm has emerged as one of the evolutionary computation techniques to solve complex problems that are not continuous, lacking linearity, derivatives and non-deterministic polynomial time problem(NP-Hard).It is a heuristic search and optimization technique that mimics the biological natural selection and survival of the fittest theory and terminates when either a maximum number of generations or a satisfactory level of fitness level has been reached for the population. The algorithm utilizes probability techniques and when structured satisfactory results could be achieved. Care is needed to eliminate convergence to local optimum by applying proper codes and encoding technique, selection and mutation.

Keywords: Genetic algorithm; Gene; Mutation; Crossover; Genotype; Fitness function

1. Introduction

Genetic Algorithms also known as GAs are one of the most popular evolutionary algorithms invented by John Holland in 1975. The inventor used binary numbers in the initial operations. In recent times the algorithm has been modified to use other coding methods including decimal numbers.

GAs are heuristic search, done for the purpose of optimization. It derives its strength from biological genetics and natural selection. It covers such areas such as evolutionary programming and evolution strategies that are governed by biological inheritance, selection and cross over and mutation. GAs are efficient tools to provide optimal or near optimal solution in the shortest time possible to difficult problems which are NP-hard (none deterministic polynomial time problem); problems where other optimization methods cannot handle due to lack of continuity, derivatives, linearity, or other features[1].

It is a suitable approach in solving optimization with respect to global optimization that involves several local maxima or minima. It searches, computes and compares several local maxima or minima and established the global solution. GAs may also be used in other problems that are not stochastic. GAs are used where the elements of the variables are real, discrete or complex valued. It is based on the principle of natural selection and survival of the fittest. In GAs, a pool

* Corresponding author: Whyte Asukwo Akpan

is generated for a given problem. The solution will undergo recombination and mutation to produce offsprings and the process is repeated over various generations. In each generation, each individual or candidate is assigned a fitness function value. The fitter individuals are given a higher chance to mate and yield more fitter individuals as theorized by Charles Darwin in his theory of survival of the fittest.

The values of the decision variables in the population are called the chromosomes. The main idea in GA is to navigate a set of chromosomes from an initial collection of values to a point where the fitness function is optimized. It starts with a population of randomly generated individuals in generations. In this generation, the fitness of every individual in the population is evaluated. Multiple individuals are selected from the current population based on fitness and modified to form a new population. The resultant population is used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population.

According to [1], all living organisms are made of cells and each cell contains the same set of one or more chromosomes-strings of deoxyribonucleic acid (DNA).The DNA serves as a blueprint for the organism. It s a molecule that carries genetic instruction for the development and functioning of an organism. A chromosome can be conceptually divided into genes, each of which encodes a particular protein. One can think of it as encoding a trait, such as a eye colour. The different possible setting for a trait (e.g blue, brown,or haze) forms alleles. Each gene is located at a particular position in the chromosome.

Many organisms have multiple chromosomes in each cell. The complete collection of genetic material is called the organisms' genome. Genotype describes a particular set of genes contained in a genome. Two individuals that have identical genomes are said to have the same genotype. The genotype gives rise under fetal and later development to the organism's phenotype representing its physical and mental well being, such as eye colour, height, brain size and intelligence.

When the chromosomes of the organisms are arranged in pairs they are called diploid and those not arranged in pairs are called haploid. Most sexually reproducing organisms are diploid , like human being with 23 pairs of chromosomes in each somatic (non-germ) in the body. During sexual reproduction , recombination or crossover occurs. In each parent, genes are exchanged between each pair of the chromosomes to form a gametes (single chromosomes), and then gametes from the two parents pair up to create a full set of diploid chromosomes. In haploid sexual reproduction, genes are exchanged between the two parents 'singled-strand chromosomes.

Offspring are subject to mutation, in which single nucleotides are changed from parent to offspring, the changes often resulting from copying errors. The fitness of an organism is typically defined as a probability that the organism will live to reproduce or as a function of the number of offspring the organism has fertility.

In genetic algorithms, chromosome refers to a candidate solution to a problem, encoded as a bit string. The genes are either single bits, short block of adjacent bits that encode a particular element of the candidate's solution (e.g in the context of multiple parameter function optimization the bits encoding a particular parameter might be considered to be a gene). An allele in a bit string is either 0 or 1, for larger alphabets more alleles are possible at each locus. Crossover typically consists of exchanging genetic material between two single - chromosomes haploid parents. Mutation consists of flipping the bit at a randomly chosen locus (for larger alphabets, replacing the symbol at a randomly chosen locus with a randomly chosen new symbol).

2. Practical Implication

Evolutionary computation has offered several opportunities for the solution of complex engineering problems that were so difficult and almost impossible in some cases to solve. Genetic algorithm is one of such branches of evolutionary computation that need to be explored to solve complex engineering problems. Such problems include control systems, maintenance scheduling, inventory control, workforce scheduling, machine replacement, productivity analysis. For these problems to be properly tackled and solved, there is the need to understand the intricate procedures involved in natural selections and survival of the fittest theory by Charles Darwin and also be able to navigate the problem terrain and identify suitable methods that will promote satisfactory results.Present encoding schemes need be examined and evaluated. Future encoding schemes are not out of place, provided they are able to show and provide better and efficient solutions to these problems. There is also they need to explore where genetic algorithm offers potential advantages compared to other computational methods.

2.1. Background to the Problem

There many problems in engineering that are very complex that analytical or numerical solutions may be cumbersome or impossible to solve. Such real problems need unique methods to address them. To enable the results of the problem to be meaningful and realistic, a systematic procedure must be followed. Genetic algorithm falls into this nature and proper attention is needed to achieve good results in its applications.Proper understanding of the technique is required which this research is poised to address.

3. Genetic Algorithm

The basic steps in genetic algorithm include:

Starting with a large ‘population’ of randomly generated or ‘attempted solutions’to a given problem

Repeatedly:

- Evaluate each of the attempted solutions
- Probabilistically, keep a subset of the best solution
- Use these solutions to generate a new population (offspring)
- Quit when you have a satisfactory solution is reached.

3.1. Encoding Problem

A population of possible solutions to a specific problem in genetic algorithm is represented by the chromosomes. This is achieved by choosing and applying a suitable encoding scheme. The encoding may be classified either as one or two dimensional encoding. Examples of one dimensional encoding are : binary, octal, hexadecimal, permutation and value encoding. Similarly, example of two dimensional encoding is tree encoding. In binary encoding, the encoding scheme depends on value and order. In permutation encoding, the scheme depends on order only. Value encoding has its scheme depending on value only.

4. Binary Encoding

In binary encoding each chromosome is represented using a binary string. Every chromosome is a string of bits of 0 or 1.In binary encoding, each string represents a value. With smaller number of alleles, many posible chromosomes can be represented. This encoding is not often natural for many problems. In some cases, corrections will be made after crossover and or mutation. Possible crossover operations in binary encoding are 1-point cossover, N-pont crossover, uniform crossover and arithemetic crossove. The mutation operator possible is flip mutation. In flip mutation, bit changes from 0 to 1 or 1 to 0 based on generated mutation chromosomes. Table 1 shows the binary encoding scheme.

Table 1 Binary encoding Scheme

Individual Chromosome Number	**String (genotype)**	**Decimal Value (Phenotype)**
Chomosome 1	01101	13
Chromosome 2	11000	24

In octal encoding a chromosome is represented using octal numbers (0-7) as shown in Figure 2.

Table 2 Octal Encoding

Individual Chromosome Number	**String (genetype)**	**Decimal Value(phenotype)**
Chromosome 1	06354524	1,694,036
Chromosome 2	63726525	13,610,325

Table 3 shows hexadecimal encoding scheme.

Table 3 Hexadecimal Encoding

Individual Chromosome Number	String(genotype)	Decimal Value (phenotype)
Chromosome 1	97AA	38,826
Chromosome 2	A2B6	41,854

Hexadecimal encoding uses hexadecimal numbers (0-9, A-F) to represent a chromosome.

In permutation encoding scheme, each chromosome is a string of numbers , representing numbers in a sequence. Each chromosome is taken to be an arrangement or a permutation of a set of genes. The search space is the total number of possible permutations, while the population is the current set of permutations. Values must not be repeated in a single chromosome. Examples given is a set of genes (ABCDEFG) or (12345678).The possible chromosomes are shown in Table 4.

Table 4 Permutation Encoding

Chromosomes	Alphabets	Numbers
Chromosomes 1	ABDCEGF	14368572
Chromosome 2	BFGDCAE	32147586

Permutation encoding is used in ordering problems, such as the classical salesman travelling problem, where the string of numbers represents the sequence of cities vsited by salesman. In this example, each chromosome represents position in a sequence. Sometimes corrections have to be done after genetic operations is completed. Crossover and mutation operation of this type of encoding must take care not to alter the chromosome in an inconsistent manner.Crossover operates on permutation encoding are patially mapped crossover (PMX) , cycle crossover (OCX) and oder crossover (OX) . Permutation encoding uses inversion mutation operator applied on ordered chromosome. It simply changes the location of characters.

5. Value Encoding

In value encoding, each chromosome is represented as the string of some value. The value can be integer, real number, character or some object. Value encoding in particular is useful when the problem values are difficult to capture in a binary scheme. Fo integer values, the crossover operator applied are the same as that applied in binary encoding. Values can be anything connected to the problem, from numbers, real numbers, chars to some complicated objects. Figure 5 shows the value encoding scheme.

Table 5 Value Encoding

Individual Chromosome Number	String(genotype)	Type
Chromosome 1	1.23,2.12,3.1.34,0.34,4.62	Real numbers
Chromosome 2	ABDJEIFJDHDDLDFLFEGT	Alphabets
Chromosome 3	assad, word, odef, odef, went	Words
Chromosome 4	SSNSN,NNSW,WESNE	Directions

Value encoding can be used in neural networks[2]. This encoding is generally used in finding weights for neural network. Chromosome’s value represents weights for inputs. Often, this encoding scheme will require some unique crossover and mutation methods, specific to particular values in the problem.

Tree Encoding is used mainly for evolving programmes or expressions for genetic programming In this scheme, every chromosome is a tree of some objects, such as functions or commands in programming language. The performamce of genetic algorithm depends on the method used to encode the candidate solutions into chromosomes and what the fitness function is ascribed to do [3].

6. Generation of Initial Population

The initial population can be randomly selected, as in tossing a balanced coin , a given number of times or it can be generated randomly with respect to a certain probability distribution(normally uniform) or it can be generated using a known heuristic method for the problem. The random process gives the initial population the desired truly random nature. This method is trivial particularly, when working with binary strings [4]. The method of heuristic has been observed to lead to population having similar solutions and less diversity.

In GAs, there are two population models widely employed in practice. These are steady state and generational model. In the steady state model, one or more offsprings are generated in each iteration (generation), and they replace one or more individuals from the population. The steady state GA is also known as incremental GA. In generational model, n-offsprings are generated, where n is the population size and the entire population is replaced by the new one at the end of the iteration.

6.1. Fitness Evaluation

After generating the initial population, each string is evaluated using the evaluation function and assigned a fitness value, using the value function. Each string is referred to as a chromosome. The evaluation function or the objective function provides a measure of performance with respect to a particular set of parameters. Fitness value is determined by transforming, using the fitness function, the measure of performance from the fitness function into allocation of productive opportunities.

The evaluation of a particular string of parameters is independent of the evaluation of any other string. However, the fitness of that sting is always evaluated with respect to other strings in the current population. In most cases, fitness function and the objective function are the same , as the objective is to maximize or minimize the given objective function. Nevertheless, for more complex problems with multiple objectives and constraints, different fitness functions might be incorporated into the algorithm. The evaluation function is a function that returns an absolute measure of the individual, while the fitness function is a function that measures the value of the individual relative to the rest of the population. In practice, genetic algorithms require the population to be much higher and it is not a good idea to have negative fitness values. Care should be taken with this method when dealing with negative numbers. In this case, adjustment of numbers should be made to make sure no fitness value are negative.

6.2. Selection

Parent selection is the process of selecting parents which will mate and recombine to create off-springs for the next generation. This is important in that the convergence rate of GA is that of good control of parents to drive individuals to better and fitter solutions. There are numerous selection methods: Fitness proportional selection, Stochastic Universal Sampling (SUS),Tournament selection, Rank selection, Random selection, and Steady State selection.

Fitness proportional selection , also known as roulette wheel selection is one of the most popular ways of parent selection. In this selection, every individual becomes a parent with the probability proportional to its fitness. Fitter individual have a higher chance of mating and propagating their features in the next generations. Thus, the strategies help to apply pressure to select more fitter individuals in the population towards evolving better individuals over time. Stochastic Universal Sampling is quite similar to Roulette wheel selection, but unlike the Roulette wheel which has one fixed point on the wheel, it has two fixed points on the wheel. By this design, all the parents are selected in just one spin of the wheel. This arrangement encourages the highly fitted individuals to be selected at last once. Fitness proportional method is not applicable where the fitness can take negative value.In tournament selection, several turnaments are played among a few individuals and the individual are chosen at random from the population. In each tournament, the winner of the tournament is selected for next generations. Selection pressure can be adjusted by changing the tournament size. Weak individuals have a smaller chance to be selected if tournament is large. Tournament selection is very popular in many literatures as it can even work with negative fitness values.

Rank selection also works with negative fitness values and is mostly used when the individuals in the population have very close fitness values. In the version proposed by [5], the individual in the population are ranked according to fitness, and the expected values of each individual depends on its rank rather on its absolute fitness. The linear ranking method proposed by [2] is as follows: Each individual in the population is ranked with N, according to some equation. In many cases, the increased preservation of diversity that results from ranking leads to more successful search than the quick convergence that can result from fitness proportionate selection .A variety of other ranking schemes such as exponential has also been tried.

Random selection chooses parents from existing population. Here there is no selection pressure in favour of fitter individuals. For this reason, this strategy is usually avoided.

Most GAs described in the literature has been generational.At each generation, the new population consists entirely of offspring formed by parents in the previous generation (though some of these offspings may be identical to their parent). In some schemes, such as the elitist schemes described above, successive generations overlap to some degree-some portion of the previous generation is retained in the new population. The fraction of new individuals at each generation has been called the ‘ generation gap’[6]. In steady state selection, only a few individuals are replaced in each generation: usually a small number of the least fit individuals are replaced by offspring resulting from crossover and mutation of the fittest individuals. Steady state GAs are often used in evolving rule-based systems (e.g classifier systems)[7] ,in which incremental learning is important and in which members of the population collectively solve the problem at hand.

6.3. Crossover-Breeding

Cross over or combination is a genetic operator used to recombine the genetic information of two parents to generate the offspring. Crossover operator will stochastically generate new solutions from an existing population.Newly generated solutions are usually mutated before being added to the population. In traditional genetic algorithms, the algorithm store genetic information in a chromosome represened by bit array. The methods of crossover for bit arrays are popular. Some known cross over methods are discussed forthwith.

7. Single Point Crossover

In the single point crossover, a point on the parent’s chromosome is picked randomly as crossover point. Bits to the right of this point are swapped between the two parent chromosomes, to produce two offsprings, each with some genetic information from the two parents. Simple illustration is shown in Table 6.

Table 6 Single point crossover cycle

Parents	Offspring(step 1)	Offspring (step 2	Offspring(step 3)	Offspring(step 4)
415│3│26	4152│2│6	4│1│5216	│4│45216	345216
346│2│15	3463│1│5	3│4│6325	│3│16325	416325

7.1. Two Point Crossover

In two point crossover, two points rather a point are taken randomly to arrive at the offspring.

7.2. Arithmetic Crossover

Arithmetic cross over is commonly used for integer representation, and the crossover operator linearly combines two parents chromosomes vector to produce two new offspring. This is achieved by choosing a random weight between 0 and 1. For chromosome from parent1[1,2,3] and from parent 2 [4,5,6], if the weight taken as 0.5, the new offspring 1 is obtained as:

Offspring 1= a x parent 1_gene + (1-a) x parent 2 _gene………………..1

Offspring 1 =0.5x[1, 2 ,5]+(1-0.5)[4,5,6]

= [2,5, 3.5, 4.5]

Offspring 2= a x parent 2_gene+ (1-a) x parent 1_gene ……………….2

= 0.5[4,5,6] + (1-0.5)[1,2,3]

= [2.5,3.5,4.5]

7.3. Davis' Order Crossover(OXI)

OX1 is used for permutation based crossover, the purpose is to transmit information about relative ordering to the offsprings. The generation of the offspring is shown below:

Parent 1

1	2	3	4	5	6	7	8	9

↓ ↓ ↓ ↓

offspring

7	9	3	4	5	6	1	2	8

Parent 2 \ \ / ↑ ↑

5	7	4	9	1	3	6	2	8

Chromosomes 3, 4, 5 and 6 from parent 1 moved down and 7 and 9 from parent 2 take up first and second positions; while 1, 2 and 8 from the same parent 2 moved up to form the offspring.

7.4. Heuristic Crossover

Heuristic crossover operator uses the fitness values of the two chromosomes to determine the direction of the search. Offspring are created according to the following equation.

Offspring 1= Bestparent+ r x(bestparent - worstparent) ……………3

Offspring 2= Best parent ……………4

where r is a random number between 0 and 1.

Where offspring 1 is not feasible, which can happen if r is chosen such that one or more of its genes fall outside of the allowable upper and lower limits, then the heuristic crossover takes on a user defined parameter n. This parameter represents the number of times to try and find the r that results in a feasible chromosome. If after n trials the feasible chromosome is not obtained, the worst parent is returned as offspring.

Other crossover schemes such as partially mapped crossover (PMX), order based crossover (OX2), ring crossover, and shuffle crossover also exist.

7.5. Mutation

Mutation refers to change in the bits DNA.This change is mainly due to error in copying genes from parents. In GA it means changing a random gene in the individual. For example in binary encoding from 0 to 1 or from 1 to 0. After crossover is performed, mutation takes place. Mutation is a genetic operator used to maintain genetic diversity from one generation of population of the chromosomes. Mutation rates[2,3] in GAs are usually very minimal and lie between 0.015 to 0.012. The reason being that, the need to strive to accomplish proper balance between exploration and exploitation ability of the searching and optimizer algorithm. Exploration means searching the search space as much as possible, while exploitation refers to concentrating on one point, such as may be the global optimum.GAs mutation operators are used to provide exploration while, crossover operators are widely used to lead population to converge on one good solution (exploitation). It is important to explore much more in the beginning of the search to prevent convergence to a local minimum. On the other hand, more exploitation at the end of the search process is needed to ensure the convergence of the population of the global optimum.When the populaion converges to a local optimum, care should be taken to increase the population diversity to explore other areas.

Ther are various mutation operators namely: flip bit mutation, boundary mutation, uniform and non-uniform mutation and Guassan mutation. Flip bit mutation operation simply reverses the bit value of the chosen genes. Boundary mutation operator repalces the value of the chosen gene with either the upper or lower bound for that gene, randomly chosen. It is used for integer and float genes. Uniform mutation operator replaces the value of a chosen gene, a uniform random value selected between the user-specified upper and lower bound of the gene. This mutation operation only serves for integer and float genes.

In non-uniform mutation, the mutation operator increases the probability in such a way that the amount of mutation will be close to 0 as the generation number increases. This form of mutation operator prevents the population from stagnating in the early stages of the evolution, while allowing the genetic algorithm refine the solution in the later stages of the evolution. This mutation oprator can be used for interger and float genes.Guassian mutation operator adds a unit Guassian distributed random value to chosen gene. Then the new gene is clipped if it falls outside the user-specified upper and lower bound for the gene. This mutation operation only serves for integer and float genes.

7.6. Illustration Example

Consider the problem of maximizing a function in shown in Equation 5 [8]:

$$f(x) = \frac{-x^2}{10} + 3x \quad , \qquad \ldots\ldots\ldots\ldots\ldots 5$$

where x is allowed to vary between 0 and 31.

Solving this problem using genetic algorithm, the possible values of x must be encoded as chromosomes.The chromosomes for the genetic algorithm will be sequence of 0's and 1's with a length of 5bits, and have a range of (0000) to 31 (1111). This can be generated by tossing a weighted coin (unbaise) for 1 as head and 0 as tail. A result of such a toss is shown in Table 7, culled from [8].

Table 7 Initial Population

Chromosomes	Initial Population	x value	Fitness value $f(x)$	Selection probability
1	01011	11	20.9	0.1416
2	11010	26	10.4	0.0705
3	00010	2	5.6	0.0379
4	01110	14	22.4	0.1518
5	01100	12	21.6	0.1463
6	11110	30	0	0
7	10110	22	17.6	0.1192
8	01001	9	18.9	0.1280
9	00011	3	8.1	0.0549
10	10001	17	22.1	0.1497

Sum of fitness value = 147.6; Average f fitness value = 14.76; Maximum fitness value = 22.4

The selection of chromosomes that will reproduce is based on their fitness values using the probability function shown below:

$$\text{P (Chromosomes I reproduces)} = \frac{f(x_i)}{\sum_{k=1}^{10} f(x_K)} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots 6$$

A random experiment could be carried out to select the mating 5 pairs. Such experiment resulted in the production of the mating pairs in Table 8 culled from [8].

Table 8 Second Generation

Chromosome number	Mating Pairs	New Population	x value	Fitness value $f(x$
5	01│100	01010	10	20
2	11│010	11100	28	5.6
4	0111│0	01111	15	22.5
8	0100│1	01000	8	17.6
9	0001│1	00010**(**0**1**010**)**	10	20
2	1101│0	11011	27	8.1
7	10110	10110	22	17.6
4	01110	01110	14	22.4
10	10001	10001	17	22.1
8	01001	01001	9	18.9

Sum of fitness value = 174.8; Average f fitness value = 17.48; Maximum fitness value = 22.5

To create offspring, a crossover point is chosen at random as shown in Table 8 as a vertical line. Sometimes crossover does no occur, which means the offspring are each copies of their parents.Mutation can also take (exchange of bit in a chromosome) place as shown after 9 and 2 mate. After the crossover and mutation have been completed, the generated new population is tested for fitness function as shown in Table 8. This process can be repeated as many times as possible, until no improvement is observed or after a defined number of generations, such as 50 more is reached and the optimum solution is obtained [9,10,11,12].

8. Conclusion

Genetic algorithim has a high potential to solve complex engineering problems and also suitable for other areas of research .Its heuristic nature allows a search mechanism to explore the solution space, identifying local minima and maxima and yet arriving at the global optimum. The success of the genetic algorithm lies on proper problem encoding and fitnes and evaluation functions definitions. Once these are achieved, the results obtained from the algorithm are deemed to be meaningful. Generation of initial population, crossover, crossover length, mutation and number of generation, fitness function definitions should be fashioned out using appropriate techniques and precision. Exploration, expliotation and refinement of this algorithm to solve complex problems in the shortest possible time is the major expectation from many researchers in this field. With its nature, genetic algorithm will continue to attract many researchers in their applications and more techniques with refinement are expected in the future.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Holland H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications in Biology , Control and Artificial Intelligence, University of Michigan Press, Oxford, Uk, 1975

[2] Anit, K. (2013) Encoding Schemes in Genetic Algrithms, International Journal of Advanced Research in IT and Engineerig

[3] Baker, J.E. (1985) .Adaptive Selection Methods for Genetic Algorithms. In J.J.Grefenstette, ed.,Poceedings of the First International Conference n Genetic Alrihms and Their Applications

[4] De Jong. K.A.(1975). An Analysis of the Behaviur of a Class of Genetic Adaptive Systems. Ph,D Thesis , University of Michian, Ann Arbor

[5] Hermawanto, D. Genetic Algorithm for Solving Simple Mathematical Equatity Problem

[6] Holland, J.H.(1986) .Escaping Brittle : The Possibilities of General- Purpose Learning Algorithms Applied to Parallel Rule –Based Systems. In R.S.Michaski, J.G.Caonell, and T.M.Mitchell, eds. Machine Learning II. Morgan Kuufmann

[7] Bodenhofe,(2002). Genetic Algorithms: Theory and Applications, Lecture Notes.

[8] Holland J.H. , Adaptation in Natural and ARTIFICIAL Systems: An Introductory analysis with Applicatins to Biology, Control and Artificial Intelligence, University of Michigan Press, Oxford, Uk

[9] Carr, J. An Introduction to Genetic Algorithms, May 16, 2014

[10] Chakraborty, R.C.(2020). Fundamentals of Genetic Algorithms: AI lecture Note, June 1

[11] Cho, D.(2001).Solving Function Opimization Problem with Genetic Algorithms, September 26

[12] Katsifarakis, K.L and Karpouzos, D.K.(2012) Genetic Algortithms and Water Resources Management An Established , Yt Evolving Relationship, WIT Transactions on State of the Art in Science and Enngineering, Vol. 56.

[13] Mitchell, M(1996) An Introduction to Genetic Algorithms, MIT Press, Cambridge, Massachusetts.