American Journal of Health Research
2022; 10(6): 225-256
http://www.sciencepublishinggroup.com/j/ajhr
doi: 10.11648/j.ajhr.20221006.14
ISSN: 2330-8788 (Print); ISSN: 2330-8796 (Online)

An Intelligence Computation of Genetic Algorithm and Its Application in Healthcare Systems: *Algorithms, Methods, and Predictions*

Abdul Joseph Fofanah[1, *], Tesyon Korjo Hwase[2]

[1]Department of Mathematics and Computer Science, Faculty of Environmental Sciences, Milton Margai Technical University, Freetown, Sierra Leone

[2]Software Engineer, MTT Consulting Architects and Engineers PLC, Addis Ababa, Ethiopia

Email address:

abduljoseph.fofanah@mmtu.edu.sl (Abdul Joseph Fofanah), abduljoseph.fofanah@gmail.com (Abdul Joseph Fofanah), tsihondegefu45@gmail.com (Tesyon Korjo Hwase)

[*]Corresponding author

To cite this article:

Abdul Joseph Fofanah, Tesyon Korjo Hwase. An Intelligence Computation of Genetic Algorithm and Its Application in Healthcare Systems: Algorithms, Methods, and Predictions. *American Journal of Health Research*. Vol. 10, No. 6, 2022, pp. 225-256.
doi: 10.11648/j.ajhr.20221006.14

Received: October 21, 2022; **Accepted**: November 12, 2022; **Published**: December 27, 2022

Abstract: In this paper we present a cardiovascular diseases prediction which is referred to as heart diseases. A detail review and application of genetic algorithms in healthcare systems including machine learning algorithms were evaluated. Areas in health systems reviewed and, in this research, includes radiology, oncology, cardiology, obstetrics and gynaecology, surgery, and infectious diseases. We conducted a healthcare management with recent reviewed papers and the application of GA in various health systems using its key parameter evaluation metrics; genetic operator, mutation operators, real coded GA, pareto-based multi-objective genetic algorithm and parallel genetic algorithms. The authors also proposed an architecture of a hybrid genetic algorithm and machine learning techniques implemented in MATLAB setting. One of the leading causes of morbidity and mortality in the global population, cardiovascular disease is characterized by restricted or blocked blood vessels that can cause heart attacks, angina, strokes, and other heart failures such muscle, valve, or rhythm problems. According to our analysis and findings, between 85 and 89 percent of people over the age of 40 were significantly affected by cardiovascular diseases. This result is crucial in light of the 2014–2016 Ebola outbreak in West Africa and the ongoing COVID-19 pandemic, both of which disproportionately affected the elderly population. Our findings also suggest that the algorithm gets more complicated and performs better the higher the generation. To forecast the results from the available data, however, and to compare the probability computation with the dataset for cardiovascular disorders, GA and ML techniques are helpful.

Keywords: Cardiovascular Diseases, Genetic Algorithms, Machine Learning Algorithms, Genetic Programming, Computational Intelligence

1. Introduction

Description and background information

Parkinson Disease (PD), a central nervous system degenerative illness, is an area of the mid-brain that develops as a result of the loss of dopamine-producing cells in the substantia nigra [1]. Over the age of 55, Parkinson's disease affects about 1% of the global population, according to Betarbet et al. [1].

In severe stages of the illness, dementia and dysautonomia are frequently present, along with nonmotor symptoms. When two or more of the hallmarks of Parkinson's disease are present, such as rest tremor, bradykinesia, or stiffness [2]. Functional neuroimaging holds the possibility of better diagnosis and facilitated assessment in early disease. Parkinson's disease primarily manifests as bradykinesia,

tremor, rigidity, and postural instability. Doctors can make the diagnosis of Parkinson's disease when all of these symptoms are present. Dysphonia is one of the most difficult symptoms of Parkinson's disease, according to many patients and their families. Nearly 90% of those who have Parkinson's disease experience speech or vocal problems. Many different vocal tests can be used to identify the signs of dysphonic exaggerated vocal tremor, which include breathiness, reduced loudness, roughness, and diminished energy in the higher harmonic spectrum [3]. Voice data is more suited to the building of an automatic Parkinson's disease diagnosis system due to the most prevalent or common symptoms. Numerous research was carried out to identify the speech signals, which are then captured and recognized using various techniques based on certain characteristics of the signals [4-6]. Parkinson's disease is identified in individuals using a classifier based on specific signal properties. The classifier is necessary for the automatic diagnosis system. Despite the fact that there are a lot of unrestricted variances, the highest precision is attainable.

The findings make it quite evident that computers have transformed daily life. including fields like engineering, archaeology, astronomy, and many others. The foundation of imaging, diagnostic, monitoring, and therapeutic devices, which have made significant contributions to medicine, is made up of electronic chips and computers. Devices are maintained and controlled by software applications and consequently dependent on algorithms. These devices are made up of numerous different physical components. An algorithm is a well-described set of rules and instructions that define or describe a series of actions or functions.

Metaheuristic techniques, according to Osman et al. [7], are algorithms that can more quickly solve complex problems or, alternatively, provide an approximate answer when traditional techniques are unable to locate an accurate one. Researchers have created a large number of metaheuristic algorithms to optimize existing solutions that are motivated by real biological patterns and behaviour. These algorithms include: the ant colony algorithm, which was inspired by ant behaviour [8], the artificial bee colony algorithm, which was inspired by bee behaviour [9], the Grey Wolf optimizer, which was inspired by the behaviour of grey wolves [10, 11], the simulated annealing algorithm, which was inspired by the dynamics of river formation [12], the artificial immune system algorithm, which was inspired by the functions of the immune system [13], and the genetic algorithm, which was inspired by genetics.

Metaheuristic approaches for decision making have been swiftly adopted in different fields of study to solve complex problems or discover the best course of action. It is hoped that when artificial intelligence and machine learning are fully utilized, these algorithms will solve a lot of problems in various fields, especially the expectation for full implementation of q analytics. The power of these potent algorithms for offering solutions to the untold complex issues, especially physicians encounter every day, which has not been fully exploited in medicine.

We introduce the combined genetic algorithm and machine learning techniques in this study to help anticipate problems that may arise in the healthcare industry. There is also a case study for the application of Darwinian theory to the evaluation of the proposed algorithm for successive generations. Some of its applications in medicine are reviewed by the metaheuristic and machine learning.

Significance of the research

The main objective of the research is to provide an intuitive understanding of the computational intelligence and application of the GA in healthcare systems, enhancement through existing algorithms and optimization techniques, predictive pattern of chromosomes or generations over another and determine which generation performance can adequately adapt based on its attributes. The research will examine the performance of the parameter set such as selection, operators to determine its learning rates over each set of generation base on the dataset supplied into the training model of the GA.

The GA has been implemented in various healthcare systems or healthcare domain including radiology- use magnetic resonance imaging (MRI), compute tomography scan (CT), and ultrasound; oncology-for early cancer detection which made possible by the screening tests when combined with appropriate treatment and increase patient survival rates; cardiology-different areas of cardiovascular medicine have used Gas. The majority of strokes and myocardial infarctions are characterized by atherosclerotic plaques. Medical professionals would be better able to identify and map unstable or fragile plaques if the mechanical properties of the plaque, such as its elasticity, were known; obstetrics and gynecology- predicting fetal weight before to delivery might lessen the risks connected to low-birth-weight babies; pediatrics- the fetal heart rate and uterine contractions can be measured using the inexpensive, non-invasive cardiotocography method to determine the health of the fetus; and surgery- a GA-based ANN (GANN) was created to predict the outcomes following surgery for patients with non-small cell lung cancer due to the predictive potential of ANNs (NSCLC). We presented a detailed literature for the applications of GA in various healthcare domain.

Concepts of Genetic Algorithms

In comparison to previous algorithms based on metaheuristic techniques, GA is a metaheuristic technique influenced by the Darwinian laws of genetics, guaranteeing the discovery of useful answers to challenging issues. In this method, certain random solutions, which we named people, are formed. Each individual contains a large number of features, which we refer to as chromosomes. Crossover and mutations in chromosomes (features), inspired by the rule of genetics, result in a second generation of people with more varied traits.

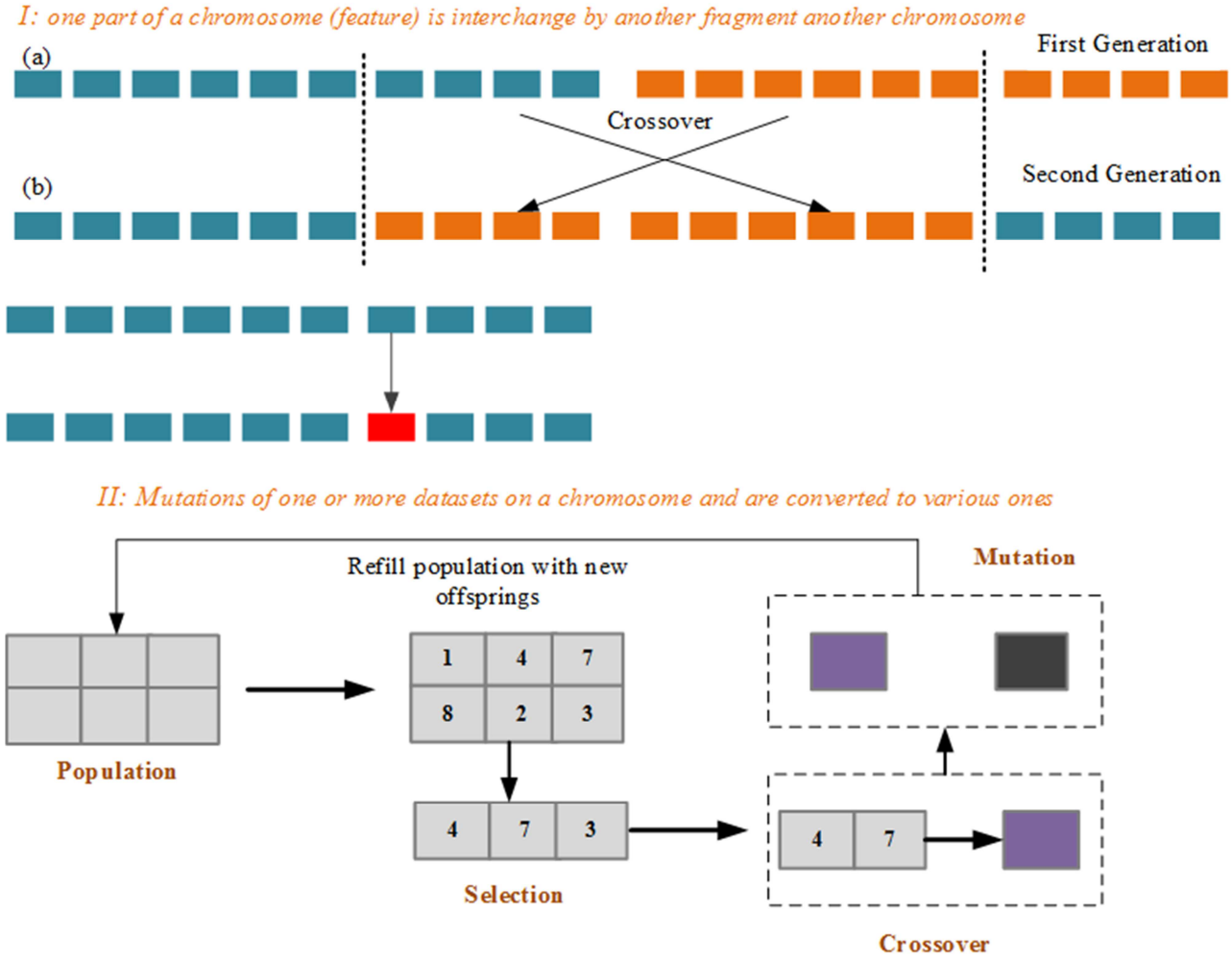

Figure 1. *Crossover, one part of a chromosome is interchange by another fragment another chromosome and II: Mutations one or more datasets on a chromosome are converted to various ones.*

Crossover and mutation are the two main methods for achieving individual diversity. Two chromosomes are chosen for crossing. The two chromosomes are then exchanged after choosing crossover locations along each chromosome (features). Figure 1 depicts a typical scenario of the phenomenon. The method of persuasion used to persuade a diverse group of people is known as a candidate solution. The diagram (a) shows that during crossover, one chromosome feature is swapped out for another segment from a different chromosome, and the diagram (b) shows that during mutations, one or more features on a chromosome are changed into different ones.

The more beneficial or fitter chromosomes (features) that new individuals produced as a result of these modifications or changes have, the more probable it is that they will be selected for breeding the next generation and the greater the likelihood that the succeeding generation will be reproduced. The mutation then creates novel configurations by making random alterations to several chromosomes [14]. Two examples of the basic mutation strategies are shown in Figure 1. There are various methods of selection; therefore, the goal of each is to map out the fitness values to individuals based on a fitness function. The genetic changes in chromosomes or traits of the selected fittest offspring will happen through crossover and mutations to form another generation. Due to the fact that the iterative process will continue until the fittest person is created or the maximum number of generations is reached, an optimal solution would need to be established [15, 16]. Additionally, GA is crucial for predicting specific parameters for a given generation and identifying the variable that will perform at the highest level, particularly when using a combinatory algorithm of GA and ML to predict diseases in the healthcare sector.

Determine a given dataset and with variable settings in the GA for each generation and compare their performance matrices are the main contributions of this paper. When compared to other optimization techniques, GA is noteworthy since it is based on derivatives. In the first place, Gas searches a population of points in the solution space in each iteration, whereas traditional derivative-based techniques search a population using probabilistic transition rules and random number generators, and on the other hand, derivative-based algorithms use deterministic transition rules for choosing the nest point in the other or arrangement. Additionally, by forecasting how successive generations would behave in terms of how well they perform and how

accurately their data can be used to solve biological and business-related issues. This makes it easier to create research-based GA problems, particularly during pandemic or COVID-19 crises.

Genetic algorithms are designed to solve problems through an evolutionary process [17]. An individual or set of solutions serves as the starting point for a genetic algorithm. A population is the collective of solutions. Each population is a solution set, and new solutions are chosen based on fitness values. Furthermore, iteration occurs again in genetic algorithms as long as the new population outperforms the previous one. For instance, a 20 generation computational recurrent is preferable to 10 generation recurrent values. For the following population, it is more likely that an individual will be reproduced the higher its fitness values (functions) are. When certain requirements, such as the population's size, are met, the iterative process is said to be finished [18]. The genetic algorithm contains eight steps, which are highlighted below:

Phase 1: There is established an n-person population random variable. These people are a good way to solve the issue. For experimental purposes, the value of n in this instance is 50.

Phase 2: Each individual (x) is represented by the fitness function f(x), which is calculated in the population [19]. Every member of the population in our experimental examples and research is created at random.

Phase 3: Two parents are chosen from the group of individuals that are chosen. These people have the population's highest fitness value. Then crossover operator realizes this parental individual.

Phase 4: In this stage, a crossover probability estimation is used for the crossover operator that creates the new individuals. If the crossover is reversed, the person will be a perfect replica of their parents.

Phase 5: In this phase, each new individual is created by mutating with a mutation chance. Utilizing any one or more bits from the individual generational computation, this mutation process is realized.

Phase 6: The new individuals are obtained from the new population in this phase.

Phase 7: In this situation, if specific circumstances are met and the end conditions are met, the GA is terminated. In this phase, the best answer within the present population is used.

Phase 8: In this phase, it is returned to phase 2. Then, the new generated population is used for further algorithm.

Research problem and GA Limitations

A population of potential solutions to an optimization issue (also known as people, animals, organisms, or phenotypes) develops toward better answers in a genetic algorithm. Although other encodings are also possible, solutions are typically expressed in binary as strings of 0s and 1s. Each potential solution has a set of adjustable characteristics (referred to as its genotype or chromosomes). The population in each iteration of the evolution, which normally starts with a population of randomly created individuals, is referred to as a generation. Every generation, every member of the population has their fitness evaluated; the fitness is ordinarily the value of the objective function in the optimization issue under consideration. The fittest members of the existing population are stochastically chosen, their genomes are combined, and possibly random mutations are introduced to generate a new generation. The new generation of candidate solutions is used in the next algorithm iteration. When the population reaches a desired level of fitness or the maximum number of generations has been generated, the process normally ends [20].

An ordinary genetic algorithm requires the following:

a) A genetic model of the problem domain and a fitness function to evaluate the solution domain.
b) Typically, each possible response is represented as an array of bits. Fundamentally, using arrays of different types and structures works the same way. These genetic representations have the essential advantage that their parts are simple to align and allow for straightforward crossover operations due to their constant size. The use of representations with varying lengths is also an option, but crossover implementation is more challenging in this case. A combination of both linear chromosomes and trees are investigated in gene expression programming. While evolutionary programming studies representations as graphs, genetic programming explores representations as trees.
c) After specifying the genetic representation and the fitness function, a GA starts a population of solutions and then refines it by repeatedly performing the mutation, crossover, inversion, and selection operations [21].

Initialization

The population of potential solutions might range from a few hundred to thousands, depending on the nature of the problem. All feasible resolutions are possible because the starting population is frequently generated at random. Sometimes, the finest solutions may be "planted" in areas where they are most likely to be discovered.

Selection

Each subsequent generation chooses a portion of the present population to breed a new generation. Individual answers are picked using a fitness-based strategy, with fitter solutions (as defined by a fitness function) frequently having a larger chance of being chosen. Some selection techniques assess each solution's fitness and favor the best ones. Due of the first process' potential length, other methods only rate a representative sample of the population. The efficacy of the represented solution is evaluated by the fitness function, which is specified over the genetic representation. The problem always determines how the fitness function works. For instance, in the knapsack problem, the objective is to maximize the total value of the objects that can fit within a rucksack with a certain amount of capacity. A solution might be represented as an array of bits, where each bit would stand for a different object, and its value (0 or 1) would indicate whether the thing is in the knapsack. These depictions aren't always correct because sometimes an item's size exceeds the knapsack's storage capacity. If the representation is, then the

total value of everything in the knapsack constitutes the fitness of the solution. The fitness of the solution equals the total value of all the items in the knapsack if the representation is accurate; else, it is 0.

A simulation or even interactive genetic algorithms may be used to estimate the fitness function value of a phenotypic when certain issues make it difficult or even impossible to define the fitness expression (for example, computational fluid dynamics is used to estimate the air resistance of a vehicle whose shape is encoded as the phenotype).

The following phase is to generate a second-generation population of solutions from the initial population by combining the genetic operator's crossover (also known as recombination) and mutation.

Genetic operator

Each new solution is produced by breeding a pair of "parent" solutions from the pool that was previously selected. A new solution is formed by using the crossover and mutation processes to create a "child" solution, which frequently has many characteristics in common with its "parents". New parents are selected for every new child, and this process is continued until a new population of solutions is created that is the appropriate size. Even while reproduction methods based on the use of two parents are more "biology inspired," some research shows that more than two "parents" create greater quality chromosomes.

These processes finally result in a population of chromosomes that is distinct from the first generation in the succeeding generation. The average fitness of the population should have increased as a result of this procedure since only the best animals from the first generation are picked for breeding, along with a small percentage of less fit solutions. These less efficient methods ensure genetic diversity in the next generation of offspring by ensuring genetic variance within the parental genetic pool.

It is worthwhile to adjust factors like the mutation probability, crossover probability, and population size in order to find the best settings for the issue class under consideration. A relatively low mutation rate may lead to genetic drift. If the recombination rate is too high, the genetic algorithm might not fully converge. In the absence of elitist selection, a high mutation rate could lead to the loss of useful solutions. A suitable population size guarantees that there is sufficient genetic diversity for the problem at hand, but if it is set to a value higher than necessary, it can waste computational resources.

Heuristics

In addition to the major operators listed above, other heuristics may be employed to improve or accelerate the calculation. The speciation heuristic penalizes crossover between candidate solutions that are too similar, which inhibits population homogeneity and delays convergence to a less perfect solution.

Termination

This generational procedure is repeated until a termination condition is satisfied. Typical grounds for termination include:

a) Coming up with a solution that satisfies the essential criteria.
b) A certain number of generations was reached.
c) The computation-related time and financial resources have been consumed.
d) The fitness of the top-ranked solution is getting close to or has passed the point at which further iterations are ineffective.
e) Manual inspection and a combination of the aforementioned methods.

Limitation of GA

There are certain disadvantages of employing a genetic algorithm in comparison to other optimization algorithms:

Why Repeated fitness function evaluation for complex tasks is typically the most prohibitive and limited part of artificial evolutionary algorithms. Finding the optimum solution to complex, multimodal, high-dimensional problems typically necessitate expensive fitness function evaluations. It may take many hours to several days of thorough simulation to evaluate a single function in practical problems like structural optimization problems. Standard optimization methods cannot resolve such issues. In this case, it could be necessary to forgo an exact assessment in favor of an approximative fitness measurement that is computationally efficient. It is obvious that one of the most promising approaches for successfully applying GA to challenging real-world problems is the combination of approximation models [22].

Genetic algorithms' complexity does not scale effectively. That is, in areas with a large number of elements prone to mutation, the size of the search space frequently increases exponentially. Because of this, using the method to solve problems like building a house, an airplane, or an engine is quite difficult. In order for evolutionary search to be able to solve such problems, they must first be reduced to their most elementary representation. As a result, we commonly observe evolutionary algorithms encoding plans for fan blades instead of engines, building forms instead of exact construction blueprints, and airfoils instead of whole aircraft designs. The second complexity challenge is figuring out how to keep elements that have evolved to be effective solutions from undergoing more harmful mutations, especially when their fitness assessment requires them to function well in conjunction with other elements [23].

Rather than the overall solution to the problem, GAs frequently converges towards local optimums or even arbitrary locations. This suggests that it lacks the "know-how" to prioritize long-term fitness over immediate fitness. The likelihood of this relies on the form of the fitness landscape; certain challenges may make it easy to attain a global optimum, while others may make it easier for the function to find local optimal points. Despite the fact that the No Free Lunch theorem proves that no one solution exists for this issue, it can be resolved by using a different fitness function, accelerating mutation, or using selection techniques that protect a diverse population of solutions.

One common tactic for conserving diversity is the application of a "niche penalty," which lowers the representation of any group of people with a high enough degree of similarity (niche radius) in future generations while preserving other (less similar) people. However, this strategy could not be successful depending on the nature of the problem. Another tactic is to simply replace a section of the population with randomly produced individuals when the majority of the population is too similar to one another [24]. Diversity is necessary for genetic algorithms (and genetic programming), as a population that is homogeneous does not offer innovative solutions when it is crossed. Because mutation is more frequently utilized than diversity, evolutionary programming and methodologies do not require it.

Working with dynamic data sets can be difficult because genes begin to converge early on toward conclusions that may not hold true for subsequent data. It has been proposed that increasing genetic variety and avoiding early convergence can cure this. These techniques include either irregularly injecting whole new, randomly produced elements into the gene pool or raising the probability of mutation when the quality of the solution decreases (a process known as triggered hypermutation). Evolutionary programming and strategies can once again be used with the so-called "comma strategy," in which new parents are only selected from offspring and existing parents are not preserved. This might perform better on motion-related issues.

One type of difficulty that generic algorithms (GAs) cannot effectively solve is decision issues since there is no mechanism to converge on a solution. A random search might yield a solution just as quickly in some circumstances. However, if the circumstances allow for a success/failure experiment to be repeated with (possibly) different outcomes, the ratio of successes to failures is an adequate fitness metric [25].

For some optimization problems and issue scenarios, other optimization techniques may be faster at convergent than genetic algorithms. Integer linear programming, ant colony optimization, evolutionary programming, simulated annealing, Gaussian adaptation, hill climbing, and swarm intelligence are a few more and alternative approaches (e.g., ant colony optimization, particle swarm optimization). Genetic algorithms may or may not be appropriate depending on how well-understood the problem is; well-known problems typically have superior, more specialized solutions.

2. Machine Learning and Genetic Algorithms Applied in Healthcare Systems

In this section, we look at genetic algorithms core applications in the field of healthcare and medicine. We present application of GA and concepts on radiology, oncology, cardiology, obstetrics and gynaecology, surgery, infectious disease, and healthcare management. A review of some of these implementations are addressed in this paper.

Radiology

Radiology generates a large amount of data that requires to be analysed and interpreted by radiologists in a relatively short time through imaging techniques. Recently, a number of tools and software programs have been created to combat the detection and diagnosis of expanding interdisciplinary technologies that aim to aid radiologists in more rapid and accurate image analysis through the segmentation, detection, and classification of normal and pathological patterns found on various imaging modalities. Ultrasound, CT scan (compute tomography), magnetic resonance imaging (MRI), and X-rays are typical examples [22].

An image of a scene, such as human body organs in radiology images, is obtained, processed, and interpreted in machine vision. The boundaries of shape and size of the objects within the photographs must be established in order to access the objects in detail. As a result, one of the crucial components of automatic image processing techniques is edge detection [26]. Many researchers obtain and use Gas for image edge detection for a variety of imaging modalities, including MRI, CT, and ultrasound.

Researchers have tried to employ computational tools to enhance the sensitivity of the system through screening mammography which is the gold standard for the detection of breast cancer despite its failure rate [27]. Furthermore, most of the applications of Gas in radiology were performed or employed on breast cancer screening primarily using mammography.

In another research conducted by Zebari et al. [28], to detect micro calculations in mammograms proposed breast cancer, the border of the breast and the nipple position were detected by the genetic algorithm (GA) and machine learning (ML), in their technique after enhancement and normalization of the mammograms. The mammogram images were aligned and deducted from each other to determine the asymmetry image purposive of breast cancer utilizing the border and the nipple position of the right and left breasts as a reference. The area under the receiver operating characteristics (ROC) which is the A2 value has been utilized as a necessary measure for evaluating the diagnostic performance of a system [26]. The proposed algorithm of the A2 value was estimated to 0.9 (19).

Similarly, Pereira et al. [29] mammogram segmentation applied as a set of computation tools to improve the detection of breast cancer. In order to eliminate the artifacts followed by denoising and image enhancement, an algorithm was first designed. Consequently, hybrid wavelet analysis and the GA enable the detection and segmentation of suspicious areas with 95% sensitivity. In terms of application and success stories, Gas has consequently been utilized for the classification and detection of clustered microcalcification in digital mammograms ([30, 31]).

Feature selection is used in machine learning which is the process of selecting a subset of features of abstraction to construct a model and eliminating variables with normal or no analytical value. The importance of feature selection is

essential since selecting irrelevant features would avoid the time, cost, and complexity of computation and minimize the accuracy of the model [32]. Minimizing the number of features would avoid the issues of over-fitting, minimize the chance of failure upon missing data, and enable a better presentation and generalization of the model [32]. Conversely, Gas has been employed for feature selection in research aiming to detect a region of interest in mammograms as normal or containing a mass [33] and to distinguish benign and malignant breast tumors in ultrasound images [34].

Prediction of tumor staying is a significant part of modeling a treatment plan. Because it is relevant for tumor staying, accurate tumor size and volume determination using non-invasive imaging studies become relevant. According to Zhou et al. [35], a system for extraction of tongue carcinoma from head and neck MRIs, GA was applied for segmentation of images in addition to an artificial neural network based (ANN-based) symmetry recognition algorithm to minimize the number of false-positive results. This model approach was able to extract tongue carcinoma from an MRI with high precision and small user dependency.

Oncology

To differentiate between a normal and dysplastic cervix, GA was utilized to examine the biomolecular data produced by Raman spectroscopy using a partial least square discriminant analysis technique. Finding a linear regression model between a dependent variable and a few predictor variables is the aim of this statistical technique (partial least squares). The system's results were 72 and 90 percent sensitive and specific at differentiating dysplasia from a normal cervix, respectively [36].

The massive gene expression profiling has paved the way that could revolutionized the field of molecular diagnostic and prognosis has been initiated through DNA microarrays. Nevertheless, the generation of large dataset poses statistical and analytical problems and has some requiring and need to find the predictive genes [37]. To find a minimum set of biomarkers with maximum classification and prognostication values in breast cancer patients Dolled-Filhart et al [38] generated microarray data through training breast cancer tissues and with many antibodies specific for various markers. The analysis revealed more than 95% five-year survival rate using Gas that three markers with available antibodies could define a population of patients.

Cardiology

Atherosclerotic plaques are a common cause of myocardial infarctions and strokes. Gas has been used in several cardiovascular medicine domains to identify mechanical characteristics of the plague, such as elasticity, that would help doctors detect and map vulnerable plaque or unstable situations more accurately [39]. For parameter estimation, a system involving gas was utilized, which is crucial for forecasting precise elasticity quantification and figuring out tissue elasticity. This system is superior than gradient-based strategies for inhomogeneous solution spaces with several local minima and the demand for significant computation time limiting their application. the quickening pace of medical diagnosis, prognosis, and illness monitoring, particularly in the areas of clinical proteomics and biomarker discovery. One cutting-edge technology that can produce readouts for thousands of patients from patient samples is mass spectrometry. Nevertheless, choosing a small number of highly relevant markers is necessary for the advancement of clinical research due to the complexity and expense of each procedure as well as computational and statistical tools for analysis. Zhou et al. [35] used an enhanced version of GA that supports the local float augmentation method to forecast the likelihood of a major adverse cardiac event (MACE) through recursiveness. A panel of seven proteins, which included myeloperoxidase, was chosen because it surpassed multiple existing approaches in accurately predicting the likelihood of MACE by 77 percent. Similar study was conducted by Fofanah A.J. et al [40] to determine the GA performance suing healthcare data to predict it the status of the patient affected by cardiovascular diseases.

Logistic regression models have been frequently used in diagnosing disease. A genetic algorithm has been utilized to select the best variables for a logistic regression system whose objective is to model the presence of myocardial infractions in patients with chest pain. This method of GA was superior in variable selection to other traditional techniques [33].

Obstetrics and gynaecology

Obstetricians can choose the best time to intervene during labour (if necessary) by comparing normal and delayed deliveries. One of the parameters that can serve forecast goals is the time to reach full cervical dilatation, delivery time, and segregate usually against protracted labour. In a study, Hoh et al. [41] used a three-parameter logistic model to predict the time it would take to reach full cervical dilation using either the Newtons Raphson (NR) approach or GA. Based on the GA algorithm, more cervical dilation was used, outperforming the NR method.

Genetic algorithms have also been applied in prenatal diagnosis Fetal macrosomia is one of the fetal features that can complicate delivery. The difference between the large for gestational age (LGA) from the appropriate for gestational age (AGA) infants and amniotic fluid from the second trimester was assessed by capillary electrophoresis. Data analysis related to Bayesian statistical theory was applied. In order to minimize the computation time required for the Bayesian computation, a GA was used to select the appropriate wavelets or variables of the electropherogram. The system distinguishes LGA from AGA using only two wavelets, one of the albumins and the other of a negatively charged unknown small molecule with sensitivity and specificity of 100 percent and 98 percent, respectively [42]. The potential problems related to low-birth-weight infants can be minimized with a prediction of fetal weight before delivery. According to Yu et al. [43], fuzzy logic with support vector regression (FSVR) to approximate the fetal weights and to determine the optimal features for the FSVR system, Gas was used to generate an evolutionary FSVR. A 6.6 percent mean absolute percent error and 0.902 highest correlation coefficient between the estimated and the actual fetal birth weight was attained which outperformed a

backpropagation neural network (BNN).

Surgery

One of the potent mathematical algorithms that can forecast how systems would behave is the ANN. Due to the predictive power of ANNs, a GA-based on ANN (GANN) was created to forecast the results of surgery for patients with non-small cell lung cancer (NSCLC). In order to optimize, the GA was used to avoid hitting local minima. When the GANN model was used to predict the outcome for NSCLC patients, it performed much better than logistic regression. However, adding tumor size to the calculation considerably enhanced the accuracy of the predictions [44].

As the population ages, there are more geriatric individuals who require heart procedures. Preventing overestimation of risk and denial of surgery for individuals who deserve it, which could happen with some prediction models due to the high occurrence of comorbid illnesses in an elderly person, would be helpful if postoperative morbidity and death were accurately predicted. Younger age was connected with a short length of stay following cardiac surgery, no preoperative beta-blocker use, a shorter cross-clamp period, and no congestive heart failure demonstrated that GA by employing GA can generate greater predictability in medicine [45].

Infectious diseases

Not only in developing nations but also in developed nations, tuberculosis is a possible lethal infectious disease after the emergence of the human immunodeficiency virus (HIV). Tuberculosis versus non-tuberculosis patients' prediction of the diagnosis, 38 parameters composed of examination parameters and laboratory data were used to develop an ANN trained by a GA algorithm. The results indicated that through classification accuracy of the system was approximately 95 percent, thus higher than the results obtained by other algorithms [46].

An integral part of the treatment modalities against HIV is composed of a hybrid of several antiretroviral medications with the objective to decrease the replication of the virus, which is highly active antiretroviral therapy (HAART) has been proposed to minimize not only side effects, but also the selection pressure on the virus that could lead to the emergence of resistant partially since long term HAART treatment requires patient compliance and possibly be related with some side effects. Consequently, Carlisle et al. [47] devised a GA-based system to select the best HAART treatment schedule to control HIV and a model of the immune system was utilized to evaluate the effects of anti-HIV drugs on virtual patients.

Healthcare management

Efficient management of monetary resources and personnel is an integral part of the health system across the globe. In order to improve patient service satisfaction, and cost-effectiveness ratios are efficient scheduling of patient admission is one of the significant elements of hospital management or healthcare management. To improve patient scheduling in an ophthalmology hospital, a mathematical model was designed and optimized using GA. In a comparison of the new algorithm from the traditional, "first come first server" by shortening the waiting list, lowering the vacancy rate of hospital beds, minimizing preoperation ability waiting time for patients, and increasing the number of patients discharged from the hospital [48]. In another study, GA and Particle Swam Optimization (PSO) were combined as another metaheuristic algorithm and it improved patient scheduling, minimize time wastage, and increase patient satisfaction [49].

Additionally, in clinical laboratories, regular rotation of staff based on their skills through various facilities is fundamental for maintaining job skills and competence. The application of Gas has improved staff rotation scheduling in clinical laboratories. The application of Gas has improved the staff rotation schedule in the clinical laboratory. The GA-based application was capable of planning the rotation of staff effectively, by ensuring maintenance of techniques and skills saving time and cost necessary for the scheduling process.

Evolution of GA Enhancement

Improvement of one makes performance limit evaluation into a universal and ideal problem, which is expressed as:

$$\min_{O_{pt} \in R} b(O_{pt}) \tag{1}$$

Where O_{pt} is the objective function that determines the intended performance; where $b(O_{pt})$ are the optimization variables that are for any simulation exercise; and where R is the set made up of all potential scenarios. Equation (1) serves as the performance limit in this scenario and is regarded as the objective function's minimal value. Additionally, equation 1 is an analogous representation of the optimization issue.

Evolution of Desired Setting

The evolution test by the enhanced GA consists of five fundamental steps, similar to the standard evaluation, as shown in Figure 2 [50], where the crossover or mutation operators are modified to increase the evaluation efficiency. In such magnitude, the new ones are referred to as crossover and manifold mutation. In relation to the performance limit, which is more likely to be activated with complicated simulations, the full crossover and manifold mutation operators are designed to improve the likelihood of developing more complex situations. The initial population, say $\mathbb{x}_1 = \{X_1, X_2 \ldots\ldots X_{2n}\}$ evaluation situations, is generated randomly. The assess situation, X_i is composed of values chosen randomly from the $m - simulations$ elements. A chromosome containing L-genes is used to depict each simulation as an individual that is related with it. An objective function measures the performance. When it repeats and reaches the desired value, it indicates that the simulation procedure has ended and the performance limit has been reached. The sort-based fitness function is employed in natural selection [51] to avoid premature selection due to over-selection of individuals and to guarantee that each person has a greater than zero chance of being chosen. This is denoted by the following mathematical function:

$$f(X_i) = \frac{P-2(P-1)(Q_i-1)}{(2n-1)(2n)} \tag{2}$$

Where $f(X_i)$ is the fitness function determining the

probability of the selection of $X_i, P \in [1, 2n]$ is the ranking of X_i number. As part of an elitist selection method to assure worldwide convergence, the best person discovered throughout the exam is recorded before natural selection [52, 53].

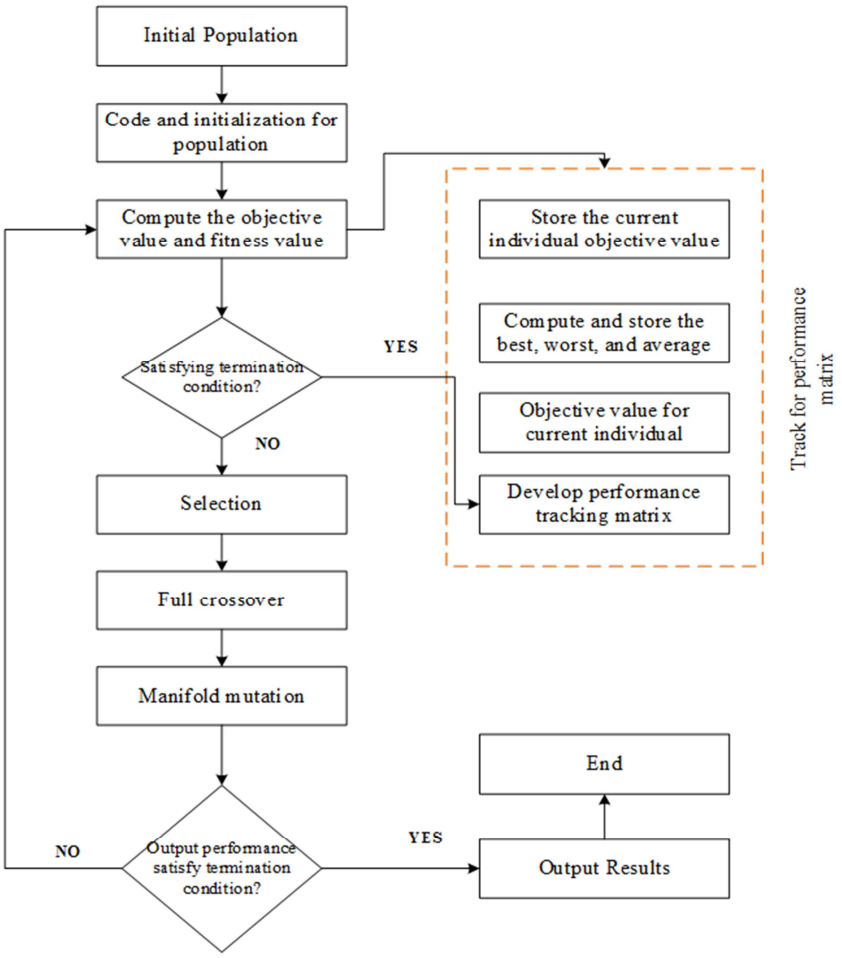

Figure 2. *Evolutionary algorithm performance evaluation.*

Complexity of GA Architecture

Crossover and mutation operators are the GA's primary inputs. In order to determine the effectiveness of the reviewed situation without carrying out the evaluation with the goal of improving them, the chance of developing more complicated simulations must increase. To set up an analytical function in various circumstances, such as between the performance of an automated driver system (ADS) and evaluation scenarios, is challenging. The situation complexity index is developed using the analytical hierarchy process (AHP) [54, 55]. Compared to the goal function obtained from equation [1] this is different.

Any complex problem has a number of different, intricate leverage variables based on the various assessment scenarios being run. When structuring the evaluation situation numerically, it is highly challenging to ascertain the importance of each aspect. This tree structure necessitates comparative study of the variables that are related to the same parent node. The relevance level of the node in the bottom layer is how the AHP is determined [56, 57]. The following mathematical examples show this:

$$T_k = \prod_{(i,j)\in\mathbb{R}} R_{i,j} \tag{3}$$

Where T_k is the significance degree, $R_{i,j}$ is the relative significance of node, $b_{i,j}$ and $\mathbb{R}$ is set composed of the index of the path from $b_{f,k}$ to the root.

An evaluation scenario, X, is denoted by the values of situation elements, that is $\mathbb{x}_1 = \{X_1, X_2 \ldots\ldots X_{2n}\}$, where X_i denotes the value of the $i-th$ situation element. To determine the exact performance limit, X_i is generated randomly in it continues range, consequently $X_i$is not equal to the discrete value, $b_{f,j}$. The linear interpolation is used to calculate the significance degree of $X_i \epsilon (b_{f,k-1}, b_{f,k})$ and the situation complexity index is obtained by the full function of:

$$C(X) = \sum_{i=1}^{m} T_i = \frac{T_{k-1}+X_i-V_{f,k-1}}{V_{f,k}-V_{f,k-1}} * (T_k - T_{k-1}) \tag{4}$$

Crossover Operator

The traditional GA has no prior knowledge regarding the effectiveness of the offspring because the crossover point is chosen at random. The likelihood of missing the best one is great; as a result, the good offspring exist in the candidate ones without a doubt. Figure 3 develops the entire crossover operator (for singleton and manifold operators). In each position, this does the single point crossover. If all candidate

kids are chosen to mutate, the number of offspring will eventually improve or rise by the geometric progression. To select the ultimate offspring pair based on their scenario complexity index, which has a positive correlation with selection likelihood, the neighbouring competition mechanism is improved. The conventional operator only creates one population for the manifold mutation operator, and there is a slim chance that population will contain any decent people. To expand the possibilities, the manifold mutation operators are designed as shown in Figure 3. It performs conventional canonical mutation on the population for [58]. Using the overall simulations' complexity index provided by the neighbouring competition process, the final progeny is selected. $X_i^m = Mut(X^C, W_m)$. By randomly selecting a candidate mutation offspring population as the final on (X^m) with the probability that:

$$W_i^m = \frac{e^{\alpha x C(X_i^m)}}{\sum_{k=1}^{N} e^{\alpha x C(X_k^m)}} \quad (5)$$

where $Mut(X^C, W_m)$ donotes the canonical mutation on X^C with the mutation probability, W_m [56].

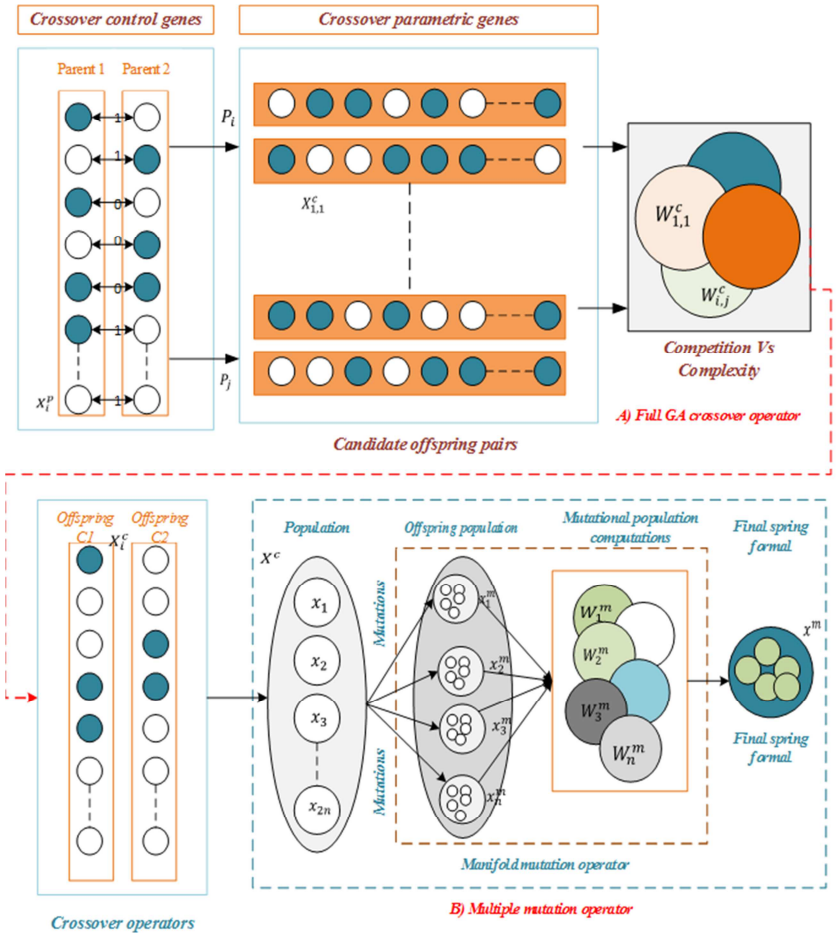

Figure 3. *Full GA crossover operator (top) and multiple mutation operator (bottom).*

Classical GA

An optimization technique called a genetic algorithm borrows the idea of natural selection from biology. It uses the Darwinian theory of evolution's premise of survival of the fittest [59] because it is a population-based search algorithm. Utilizing an iterative process, new populations are created by applying genetic operators to individuals already existing in the population. Important components including chromosome representation, selection, crossover, mutation, and fitness function serve to represent them. The GA algorithm looks like this: The initialization of an n-chromosome population (P) is random. Each P chromosome's fitness is determined. According to the fitness value, two chromosomes (examples C1 and C2) are chosen from the population P. To create an offspring, let's say Os, we apply the single-point crossover operator with crossover probability (Cp) to C1 and C2. The uniform mutation operator is then used to create O's by creating an offspring (Os) with a probability of mutation (Mp) (new offspring). Once the new population is complete, the new offspring Os are introduced. The following pseudocode is a typical classical GA (algorithm).

Algorithm 1: Classical Genetic Algorithm

Input:

Population size, n

Maximum number of iterations, Max

Output:

Global Solution, P_{bt}

Begin

Generate initial population of n chromosomes $P_i(i = 1, 2, n)$
Set iteration counter $t = 0$
Compute the fitness value of each chromosome
While $t < Max)$
Select a pair of chromosomes from initial population based on fitness
Apply crossover operation on selected pair with crossover probability
Apply mutation on the offspring with mutation probability
Replace old population with newly generated population
Increase the current iteration t by 1
End While
Return the best solution, P_{bt}
End

3. Genetic Operator

Machine learning algorithms or hybrids of GA and ML are developed using a range of gas operators. Encoding techniques, crossover, mutation, and selection processes are some of these operators. The GAs operator utilized in several health systems is depicted in Figure 4.

The encoding technique converts data into a certain format and is important in the majority of computing issues. The supplied data must be encoded using a certain bit string [60]. Depending on the issue domain, several encoding strategies are used.

The encoding systems that are currently most well-understood are binary, octal, hexadecimal, permutation, value-based, and tree. Each gene or chromosome is represented as a string of [61] in the widely used encoding system known as binary encoding. Crossover and mutation operators can be implemented more quickly since each bit represents a feature of the solution. However, binary conversion costs additional work, and algorithm accuracy depends on binary conversion. Due to epistasis and natural representation caused by the bit stream changing depending on the task, the binary encoding approach is insufficient for various engineering design challenges. Chromosomes are represented by octal numbers in the octal encoding system. The chromosome is also represented by hexadecimal numbers in the hexadecimal encoding technique [61]. In this encoding system, the string of integers that designates the place in a sequence denotes the chromosome. This is more important while trying to solve more challenging issues. It is employed in neural networks to determine the ideal weights.

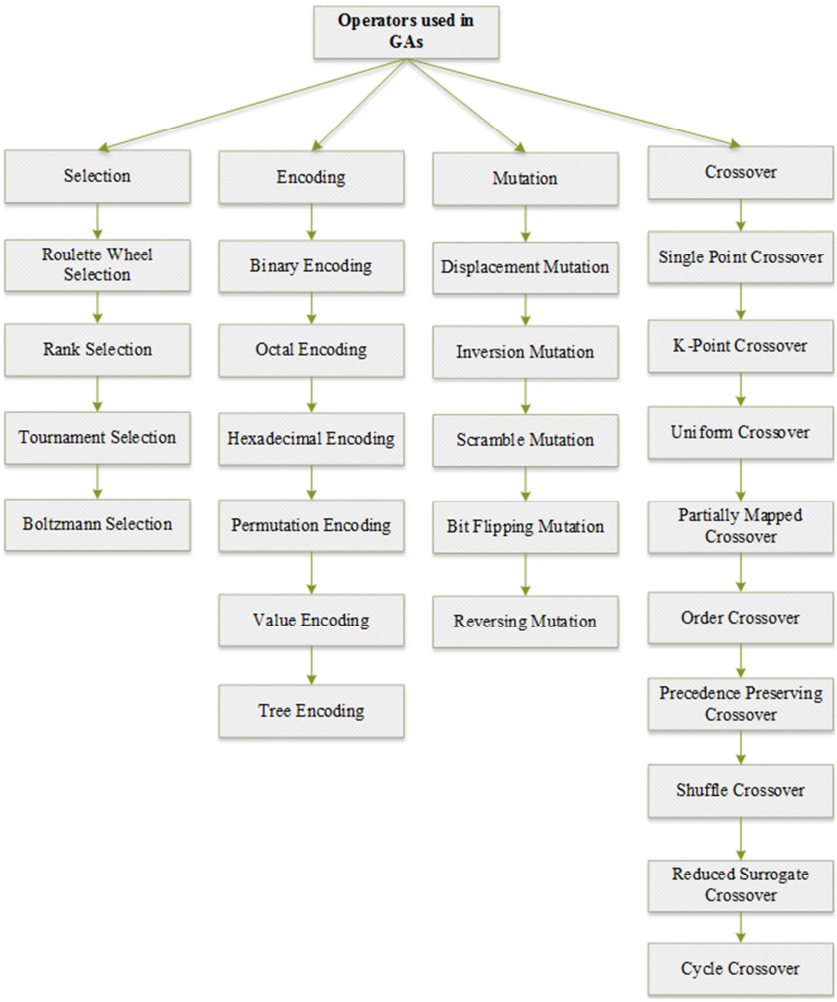

Figure 4. *Operators used in genetic algorithms.*

To determine whether the particular string will contribute in the production process or not, the selection technique is important step in genetic algorithms. The selection phase is occasionally known as the production operator. The convergence rate of GA depends upon the selection pressure. Roulette wheel, rank, tournament, boltzman, and stochastic universal sampling are the eminent examples of selection methods. Roulette wheel selection maps all the possible strings onto a wheel with a portion of the possible strings onto a wheel with a portion of the wheel allocated to them according to their fitness value. According to Agarkar et al. [61], to select specific solutions that will participate in the formation of the next generation the wheel is then rotated randomly. Nevertheless, it has some shortcomings such as error introduced by its stochastic nature. Ranks are provided according to their fitness value to enable individual gets an opportunity of getting selected in relation to their ranks and the selection technique minimizes the chances of prematurely converging the solution to a local minimum. In regards to tournament select technique proposed since 1983, the individuals are chosen according to their fitness values from a stochastic roulette wheel in pairs. The individuals with higher fitness value are added to the pool of next generation, after the method of selection, and each individual is compared against all $n-1$ other individuals provided it reaches the final population of solutions.

When using crossover operators, the genetic information of two or more parents is combined to produce the offspring. One-point, two-point, k-point, uniform, partially matched, order precedence preserving crossover, shuffle, reduced surrogate, and cycle are examples of well-known crossover operators. In a single point crossover, a random crossover point is selected, and the genetic information of two parents that is beyond that point will be exchanged. However, in a two point and k-point crossover, the genetic information is switched based on the produced segments at two or more randomly selected crossover locations. The genetic data after being swapped for single- and two-point crossover is shown in Figure 5. While in two-point crossover the middle segment of the parents is replaced to produce the new offspring, it replaced the tail array bits of both the parents.

Table 1. *Comparing various encoding algorithms.*

Encoding Algorithm	Advantages and Application	Disadvantages
Binary	a. Easy to implement b. Faster execution and can be used on problems that support binary encoding	No provision for inversion operator
Octal	Easy to implement and can be used on limited application	No provision for inversion operators
Hexadecimal	Easy to implement	No provision for inversion operators
Permutation	Provision of inversion operator and can be used on task ordering problem	No provision for binary operators
Value	Value conversion not needed and can be used on neural network problems	Requires specific crossover and mutation
Tree	Operator can easily be applied and can be applied in evolving programs	Difficult to design tree for some problems

Table 2. *Comparing different selection methods.*

Selection Method	Advantages	Disadvantages
Elitism	Preserve the best individual in a population	The best individual can be lost due to crossover and mutation operators
Sampling stochastic universal	a. Free from bias b. Fast method	Premature convergence
Boltzmann	Global optimum can be achieved	Computationally expensive
Roulette Wheel	a. Easy to implement b. Simple c. Free from bias	1) Risk of premature convergence 2) Depends upon variance present in the fitness function
Rank	a. Preserve diversity b. Free from bias	1) Slow convergence 2) Sorting required

Figure 5. *A typical scenario of a single point crossover (top) and two-point crossover (bottom).*

Mutation Operators

Mutation is an operator that keeps the genetic variety from one population to the next. Displacement, simple inversion, and scramble mutation are the three most important mutation operators. Displacement mutation refers to the operator that moves a portion of a specific individual solution inside of

itself. In exchange mutation and inversive mutation operators, a portion of an individual solution is either exchanged with another portion or inserted in a different position. The simple inversion mutation operators (SIM) in a single solution reverse the substring between any two given sites. The SIM is an inversion operator that reverses the randomly chosen string and inserts it in a random spot. The scramble mutation operator determines whether or not the fitness value of the currently created solution is improved by placing the elements in a given range of the individual solution in a random order.

Table 3. *Comparing different crossover methods.*

Methods	Advantages	Disadvantages
Uniform	a. Unbiased exploration b. Applicable on large subsets c. Better recombination potential	Less diverse solution
Single-point	a. Easy to implement b. Simple	Less diverse solution
Tow and K-point	Easy to implement	1) Less diverse solution 2) Applicable on small subsets
Reduced surrogate	Better performance over small optimization problems	Premature convergence
Precedence preservative (PPX)	Better offspring generation	Redundancy problem
Order crossover (OX)	Better exploration	Less of information from previous individual
Cycle crossover	Unbiased exploration	Premature convergence
Partially mapped (PMX)	a. Better convergence rate b. Superior than the other crossovers	NA

Table 4. *Mutation operators.*

Operator	Advantages	Disadvantages
Displacement mutation	a. Easy to implement b. Applicable on small problem instances	Risk of premature convergence
Simple inversion mutation	Easy to implement	Premature convergence
Scramble mutation	a. Affects large number of genes b. Applicable on large problem instances	1) Disturbance in the population 2) Deterioration of solution quality in some problems

Real Coded GA

The design and development of real coded genetic algorithms (RCGAs) hseas been widely utilized in different real-lie applications. The representation of chromosomes is closely related with real-life challenges. Robustness, efficiency, and precision are the major advantages of RCGAs. Nevertheless, RCGAs suffer from premature convergence. The improvement to their performance matrixes is the current tasks for researchers. By modifying the crossover, mutation, and selection operators are developed by RCGAs. The searching capability operators are not satisfactory for continuous search space. The advancements in crossover operators have been implemented to enable their performance in real settings or infrastructure. A heuristics crossover that was applied on parents to produce offspring was presented by Wright [62]. Arithmetical crossover operators for RCGAs were proposed by Jennings et al. [63] and consequently, Sato and Oyama [64] developed real-coded crossover operators, which is based on features of single-point crossover in binary GA. A novel mutation operator based on power law and named as power mutation was presented by Das and Pratihar [65]. A novel mutation operator for enhancing the performance of RCGA was presented by Tang and Tseng [66] of which both approaches were fast and reliable.

$$P_1 = \frac{1}{2}[(1-\beta)x_i + (1+\beta)y_i] \; and \; Q_i = \frac{1}{2}[(1+\beta)x_i + (1-\beta)y_i] \quad (6)$$

where P and Q are two off
− springs generated. x and y are individuals and β is a variable

whose value lies in the interval of $[0, \infty]$

Blend crossover, with mathematical formulation where offspring P is generated from parents $x \; and \; y$ from interval $[Min - (Max - Min)\delta), Max + ((Max - Min)\delta)]$ where $Min = Min(x, y) \; and \; Max = Max(x_i, y_i), \delta$ is a variable whose value lies in the interval of $[0, 1]$ [21].

The mathematical formulation of genetic operators in RCGAs such as arithmetic crossover and geometric crossover are presented as follows:

Arithmetic crossover: $P_i = \delta x_i + (1-\delta)y_i$ and $Q_i = \delta y_i + (1-\delta)x_i$

Geometric crossover: $P_i = y_i^{\delta} * y_i^{(1-\delta)}$ and $Q_i = y_i^{\delta} * x_i^{(1-\delta)}$

Similarly, the operator unimodal normal distribution crossover operator with mathematical formulation given by the following functions:

$$P_i = X_p + \mu V + \sum_{k=1}^{n-1} \psi_k \, L_{ek} \quad (7)$$

$$Q_i = X_p - \mu V - \sum_{k=1}^{n-1} \psi_k \, L_{ek} \quad (8)$$

$where \; e_k, k = 1, \ldots \ldots n-1$ are orthogonal bases that perpendicular to $V * X_p$ in the midpoint and V is the difference vector, μ is a random value taken from normal distribution and ψ_k and $n-1$ random values follow a normal distribution. L is the length from parent 3 to perpendicular line [67].

Furthermore, Deep et al. [68] presented a mathematical formulation of Laplace crossover genetic operators as stated below:

$P_i = X_i + \beta|X_i - Y_i|$ and $Y_i + \beta|X_i - Y_i|$ thus,

$$\beta = \begin{cases} a - blog_e(u), u \leq \frac{1}{2} \\ a + blog_e(u), u\frac{1}{2} \end{cases} \quad (9)$$

where a and b are variables.The default values of a and b are 0 and 1, respectively, u is random variables.

Multi-objective GAs

The modified version of simple GA is called multi-objective. The multi-objective GA differs from GA in regards to its fitness function that is assigned to it. The other remaining phases are similar to GA. The generated optimal Pareto Front in the objective space in such a manner that no further enhancement in any fitness function besides distributing the other fitness functions are the main objective of multi-objective GA [69]. The major goal of multi-objective Gas are convergence, diversity, and coverage. Pareto-based and decomposition-based multi-objective Gas are the two broadly categorized of multi-objective Gas.

Pareto-based Multi-objective GA

The idea of Pareto dominance was proposed in multi-objective Gas and the first multi-objective GA was developed by Fonseca and Fleming [70]. Nevertheless, multi-objective GA has some shortcomings in regards to parameter tuning problem and degree of selection pressure. The proposed method was the niche and decision maker concepts to resolve the multimodal challenges. Furthermore, a niched pareto genetic algorithm that used the concept of tournament selection and pareto dominance was proposed by Horn et al. [71] development of a non-dominated sorting genetic algorithm, fast elitist non-dominated sorting genetic algorithm, dynamic crowing distance in non-dominated sorting genetic algorithm sorting genetic algorithm (NSGA-II) [21], a multi-objective micro-based techniques may be deteriorated in many problems developed by Coello and Pulido [72].

Parallel GAs

In order to improve the computational time and quality of solutions through distributed individuals, the motivation behind is the design and implementation of GAs Master-slave parallel GAs, fine grained parallel GAs, and multi-population course gained parallel GAs are the major broad categories of parallel GAs [73]. The computation of fitness functions is distributed over the various processors and consequently referred to as master-slave parallel GA. On another hand as parallel computers are used to solve the real-life problems thus referred to as gained GA. The genetic operators are bounded to their neighbourhoods. Nevertheless, the exchange of individual among sup-populations is performed in a course gained by GA and the interaction is enhanced among the individuals. The control parameters are also transferred during migration. To maximize memory bandwidth and arrange threads for utilizing the power of GPUs are the major challenges in parallel GAs. The comparative analysis of parallel GAs in regard to hardware and software are indicated in Table 5.

Table 5. *Analysis of parallel GAs in regard to hardware and software.*

Hardware	No. of Processors	Programming Language used	API	Main Application of GA
Cluster	130	Java		Data mining
Multicore CPU	8	Java		Path finding
Cluster	30	Fortran	MPI	Road traffic
Cluster	48	JavaScript	Node. JS	Building structure
Multicore CPU	8, 3	Java	Java. util. component	Land planning Job scheduling
Cloud	300		MPI	Internet of things
Cluster	100		MPI	Wireless Network
GPU	448		CUDA	Scheduling
GPU	512		CUDA	Electronics

4. Architecture of the Proposed Hybrid GA and ML Approaches

MATLAB is currently a widely used programming environment that supports numerous computer platforms. It is the fastest and most powerful mathematical calculus because of its simplicity, ease of use, and learning capacity. Its rich and simple data visualization features also make it a very enticing programming environment. The MATLAB environment offers a strong base upon which to construct and optimize toolbox collections, as well as application-specific functionalities. The MATLAB toolbox is known as GPLAB (genetic programming).

The proposed hybrid genetic algorithm and machine learning algorithms (HGAMLA) approaches include dynamic functionalities such as reset *parameter's function, set parameters function, set operators function, desired obtained function (variables, indices, x parameters, black and white, size of x and y variables), accuracy and complexity function (variables, offsets, black and white, and size x*y), pareto function), and GP tree function.*

Algorithm 2: Parameters Variables for GPLAB Algorithm

Switch

Nargin

Case

2

Determine the type

Case
3
Equate the type and variable nargin
Otherwise
Determine the error input arguments
End
What will recognized or not as parameter state variables
What will be the defaults values of the define parameters
If
Parameters and list are defaults
Determine all variables
For all required variables (i=1)
Evaluates the default values
Evaluate the parameters
End
End
If
Structural parameter-list are defaults
Setting separators allowed
Assignment symbols allowed
For i=1 (number of parameters)
If number of pieces =2
If field and available values
Sum tests the text parameters in which there is a cell array of valid strings
If sum>0 and available values
If Iscell values ≠ 0
Evaluate the parameters
Else
Evaluate values
End
Else
Warning (invalid parameters values
End
Else
Warning (set parameters: unknown parameters
End
Else
Warning (set parameters: invalid parameter settings)
End
End
End

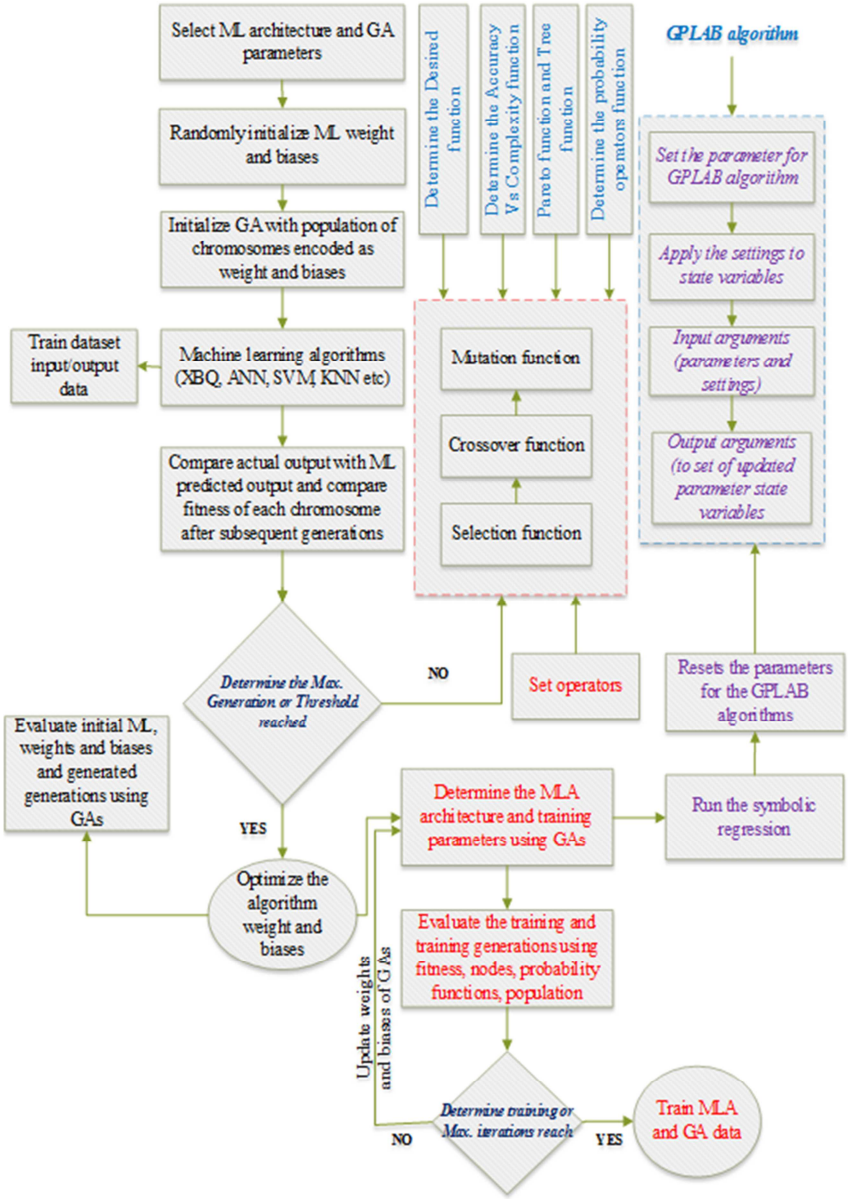

Figure 6. *A block-diagram for the proposed HGAMLA architecture.*

A working GPLAB toolbox that executes the symbolic regression problem known as the polynomial quartic serves as an illustration of this architecture. The proposed HGAMLA architecture of the GA and ML algorithms for health-related problems as well as general predictions and estimate criterion settings is depicted in Figure 6 by a block diagram. In our experimental simulation activities, 100 individuals were used for 40 or more generations with automatic operator probability adaptation, drawing a number of plots during runtime, and two more plots at the end of the run. The experimental outcomes return all the algorithmic variables required to plot charts, carry out more runs, draw trees, etc. They also return the top candidates discovered throughout the simulation exercises.

Table 6 shows the block diagram that was utilized to create the proposed HGAMLA architecture of the GA and ML algorithms for general predictions and estimate criteria settings and health-related challenges.

Set Operators Function

Sets the parameter variables for the GPLAB algorithm. The set parameters incudes *Params and Setting* which applies the settings to the parameters stored in *Params*, and returns the updated set of the parameters. If the *Params* is empty, all the parameters not included in *Settings* are set with the default values. The *Setparams* includes three parameters: *Params, Settings, and State*, which applies the settings to the state variables instead of the parameter variables. This is not advisable for the user to do as it may put the algorithm into an inconsistent state. The set operators have two arguments called input argument and output arguments. The set of parameter or state variables called structure defines the input arguments and the set of updated parameter or state variables construct the output arguments.

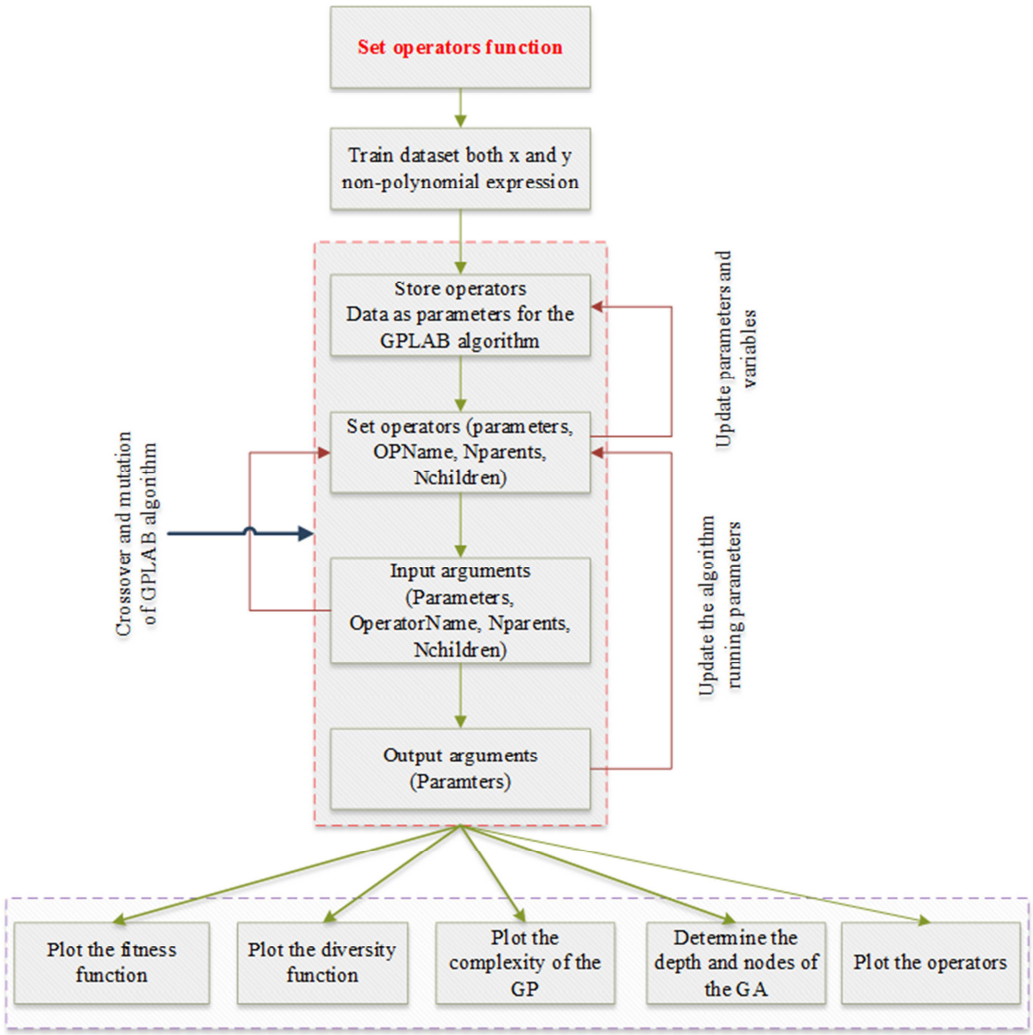

Figure 7. *Set operators function for the GPLAB algorithm.*

The *Setoperator* functions stores information as parameters for the GPLAB algorithm. The set operators include four parameters: *Params- the algorithm running parameter structure, OPName-the name of the operator to use (string), NParents- the number of parents required by the operator (integer), and NChildren- the number of children produced by the operator (integer) has been set with the data provided in the function arguments (OPName, NParents,*

NChildren). Several operators can be set to the same time by adding several triplets *(OPName, NParents, NChildren)* to the list of arguments. The functional set operator algorithm is represented in Figure 7 of the GPLAB algorithm.

Desired Obtained Function

The desired obtained function contains four parameters: *VARS, INDICEBEST, INPUTVAR, BLACKWHITE, and SIZEPLOT*. It is called the plots desired functions with the GPLAB. The desired functions draw a plot with the desired function to approximate and the approximations obtained by the best individuals in indices best. Only one input variable x is represented in each plot. The plot can be sized by the user with '*SIZEPLOT'* and be drawn in black and white by using the flag '*BLACKWHITE'*. If *INDICESBEST* is empty, all the available best individuals will be used, until a certain limit. If the size plot is empty, the default plot size will be adopted, if any of size plot dimensions is null, the default size for that dimension will be adopted. The input arguments: *VARS*- all the variables of the algorithm, *INDICESBEST*- the best individuals to plot for all $1xn$ matrix, *INPUTVAR*- the input variable to plot (integer), *BLACKWHITE*- the flag to draw a black and white or colour plot (Boolean), SIZEPLOT- the $x\ and\ y$ size of plot for default $1x2$ matrix.

The desired obtained function algorithm is explained below: The symbline variables are initialized if the draw black and white flag colour plot (Boolean) is not empty. The *SIZEPLOT* that contains the x and y size of plot are initialize and equated to zero. The algorithm also checks if indices and best size is equated to one (1) and determine the best history. If the black and white (BW) and length called indices is greater than length (symblline) the algorithm provides a warning of the desired obtained and cannot use so many indices and consequently, discard the last ones. If the indices are greater than the size and lesser than 1, the algorithm provide a warning of the desired obtained and some indices not available. This enables the algorithm to get the desired results and obtained results by determining the length and compute the individual population. The architecture of the functional desired and obtained function is represented in Figure 8.

To determine the correct x variable, the algorithm checks if variable x is less than one and variable x is greater than length (data), the algorithm send a desired obtained error called '*input variable not available*'. The algorithm search for required dataset size and sort them in rows by computing the $size\ x\ and\ y\ variable$ of which one is less than and equal to zero and equal to 400 and equal 360 to obtain the desired versus obtained. The black and white flag chart is drawn by determine the list of variables and function and finally build the length or indices to approximate base on generation of GA.

Accuracy and Complexity, Pareto Function, GP Tree Function

The accuracy versus complexity function incapsulate four main parameters: *VARS, OFFSETS, BW (black and white), SIZEXY)*. The plots accuracy and complexity measures with GPLAB. The accuracy and complexity draw a plot with the evolution of fitness and complexity measures level nodes of the best individuals. The offsets can be used to improve the visibility of the several lines in the plot (fitness, level, nodes). The plot can be sized by the white by using the flag *BLACKWHITE*. If the offsets are empty, no offsets will be considered. If *SIZEPLOT* is empty, the default plot size will be adopted, if any of *OFFSPLOT* dimensions are null, the default size for the dimension will be adopted. There are four input arguments in the algorithms: VAR-all the variables of the algorithm (struct), *OFFSETS*-the offsets for each line and empty for no offset (1x3 matrix), BLACKWHITE-the flog to draw a black and white or colour plot called Boolean, and *SIZEPLOT*- the $x\ and\ y$ size plot with empty for default ($1x2\ matrix$). Figure 8 shows the flowchart for the accuracy and complexity of the algorithms (right).

The Pareto front is plotted using the GPLAB. The plot Pareto variables draw a plot number of nodes in blue (see analysis and results). The red graph is the Pareto front that is the set of solutions for which no other solution was found of which both has smaller free and better fitness. The sizes and fitness of the current population are plotted in green. The inputs arguments contain *VARS*- all the variables of the algorithm are structured. The Pareto front has no output arguments and it perform the following functions: It ensure the number of nodes are computed for current population, it builds the Pareto front variable, it performs the cross validation, collect fitness and number of nodes, and it compute and plot the Pareto front.

The GA Tree function algorithm translate a GPLAB tree into a string. The tree (*TREE2STR*) returns the string represented by the tree, in valid MATLAB notation, ready for evaluation. The input arguments contain the *TREE* parameter (the tree to translate the structure) and the output arguments contain string parameter (the string represented by the tree). The *TREELEVEL* counts the number of levels of a GPLAB algorithm tree. The GPLAB algorithm that represent individual representation tree with input and output arguments that contain *TREE* parameter -the tree of measure and *NLEVELS* the depth level of the tree, respectively. The *TREESIZE* returns the tree size of a GPLAB individual. The tree size includes *INDIVIDUAL, PARAMS, DATA, TERMINALS, and VARSVALS* that returns the tree size of individual measured as the number of nodes. The input arguments include: *INDIVIDUAL*- the individual whose fitness is to measure (struct), DATA- the dataset on which to measure the fitness struct), *TERMINALS*-the variables to set with the input dataset (cell array), *VARSVARS*- the string of the variables of the fitness cases (string). The output arguments include *INDIVIDUAL*-the individual whose tree size was measured. The number of nodes counts of a GPLAB algorithm tree can be determine by returns of *NODES* (functions and terminal) of a GPLAB algorithm individual representation tree (see Figure 7 left).

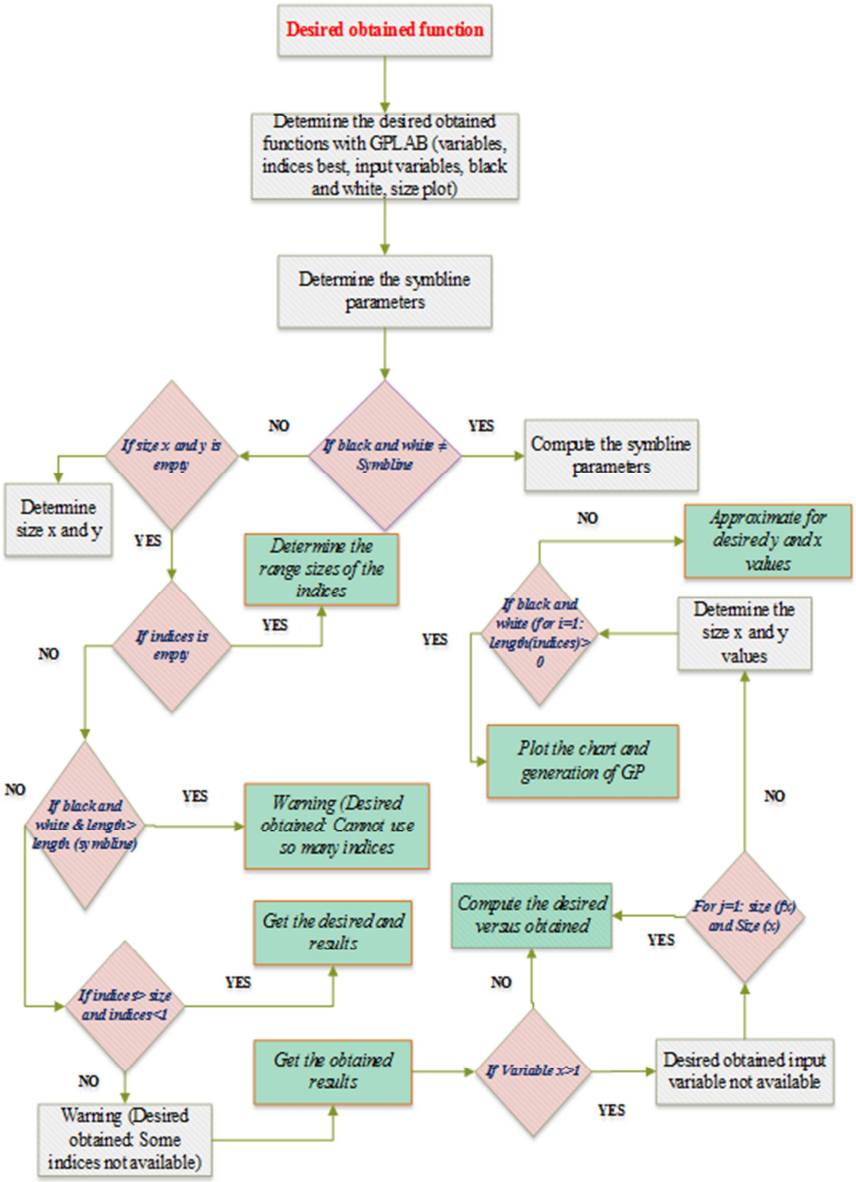

Figure 8. *Desired and obtained function flowchart.*

Descriptions of Dataset

The dataset employed in this experimental analysis was divided into three sections: the probability expressions, the heart disease dataset, and the 11-multiplexer represented in x and y variables. In order to compare the block and thread intertwined spirals (x variable) against the variable, the multiplexer was tested using a training set size of 2048. The y-axis determines the probability expression in the and plot, which is represented between [1.1052, 2.7183], and and between and. The dataset for cardiac disease is displayed as follows: A dataset is created by taking into account some of the data from 779 people. The challenge is to determine whether each person will get heart disease based on the information that has been provided about them. The dataset has 14 parameters that are effective for learning GA and ML, and they indicate variable x, which includes the following parameters: Age: displays the person's age; Sex- uses the following format to display a person's gender: 0 = female, 1 = male; Chest-pain Type- uses the following structure to show the kind of chest pain the person is feeling: 1 denotes asymptotic angina, 2 typical angina, 3 non-anginal pain, and 4 atypical angina; Resting Blood Pressure: Displays a person's resting blood pressure in millimetres of mercury (mmHg); The serum cholesterol is displayed as mg/dl in Serum Cholesterol (unit), Fasting Blood Sugar: Compares a person's fasting blood sugar level to 120 mg/dl. 0 (false) if fasting blood sugar is less than 120 mg/dl, 1 (true) otherwise; A normal resting ECG is 0; an irregular ST-T wave is 1, and a left ventricular hypertrophy is 2; Max heart rate achieved—displays a person's maximum heart rate; 0 = nil, 1 = exercise-induced angina; Exercise-induced ST depression compared to rest: reveals whether a value is an integer or a float; Peak exercise ST segment: 1, 2, and 3 are upslopes, respectively; Fluoroscopy-based main vascular count (0–3): value is shown as an integer or float; Thalassemia: 3 indicates normal, 6 indicates a fixed defect, and 7 indicates a reversible defect;

Heart disease diagnosis: Determines whether a person has heart disease or not. 0, 1, 2, 3, and 4 indicate presence.

Model Training and Prediction

Because we already know whether each patient has heart disease, probability expressions, and an 11-multiplexer, we can train our prediction model by analyzing the data that is currently available. This will help us to assess the effectiveness and accuracy of the GA model. This method is often referred to as learning and supervised modularization. The trained model is then applied to analyze the complexity, variety, and performance of the GA model, as well as forecast whether users have heart disease.

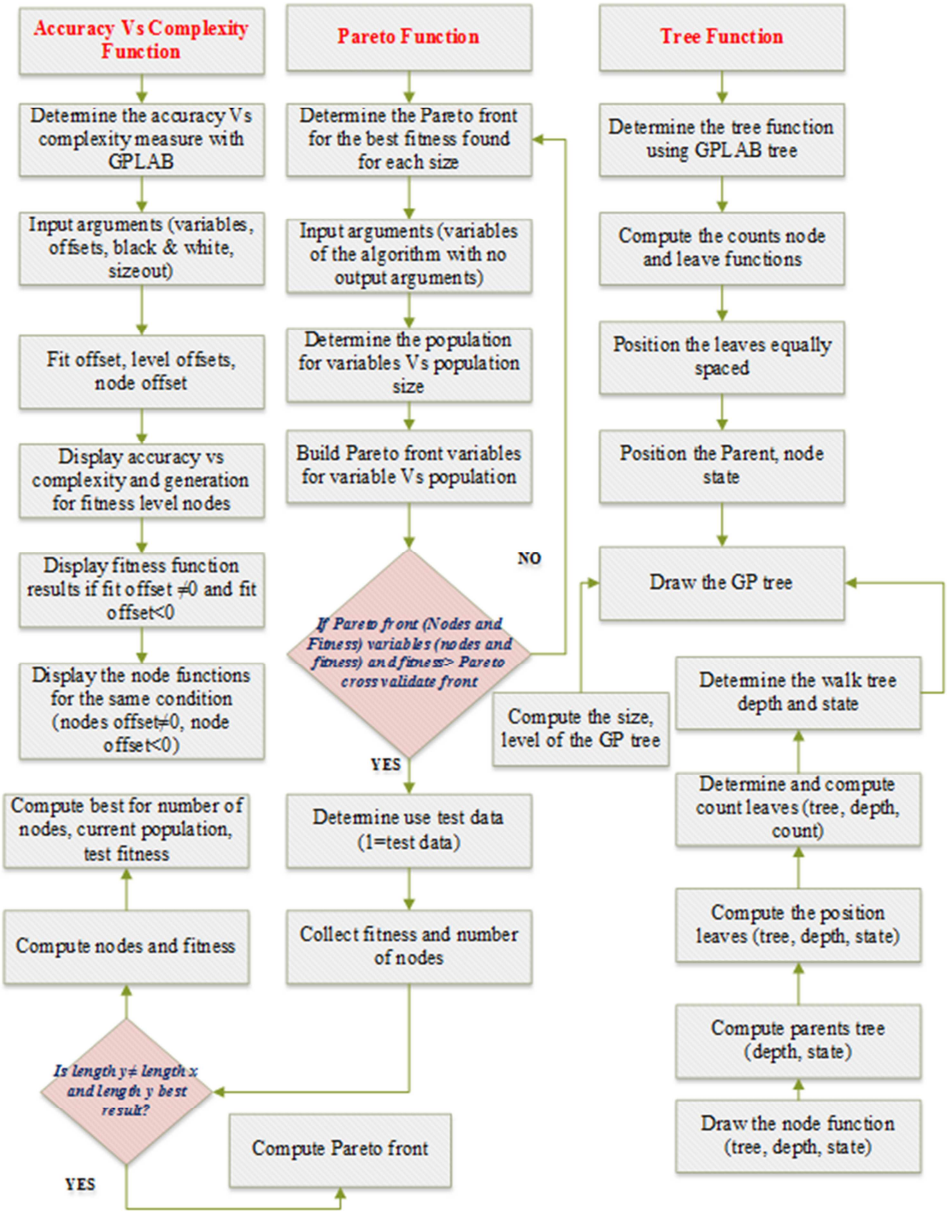

Figure 9. *Model training and prediction.*

5. Experimental Results and Analysis

The experiment was carried out using the MATLAB programming environment, an x64-based processor, an Intel (R) Core (TM) i7-7500U CPU running at 2.70GHz and 2.90GHz, and 16GB of RAM. In order to determine the following GPLAB functions—accuracy vs complexity, pareto front, GPLAB tree functionality, desired achieved function, GA operators, fitness, structural complexity, and population diversity—the findings were presented in tables and figures. For 25 generations, 100 individuals were inserted in accordance with the analysis, and the results for each generation were shown in Table 6. Surprisingly, the GA provides varied values depending on the generation type, consumed resources, fitness, test fitness, depth, and number of nodes for any set of computation (25 generations and 100 individual).

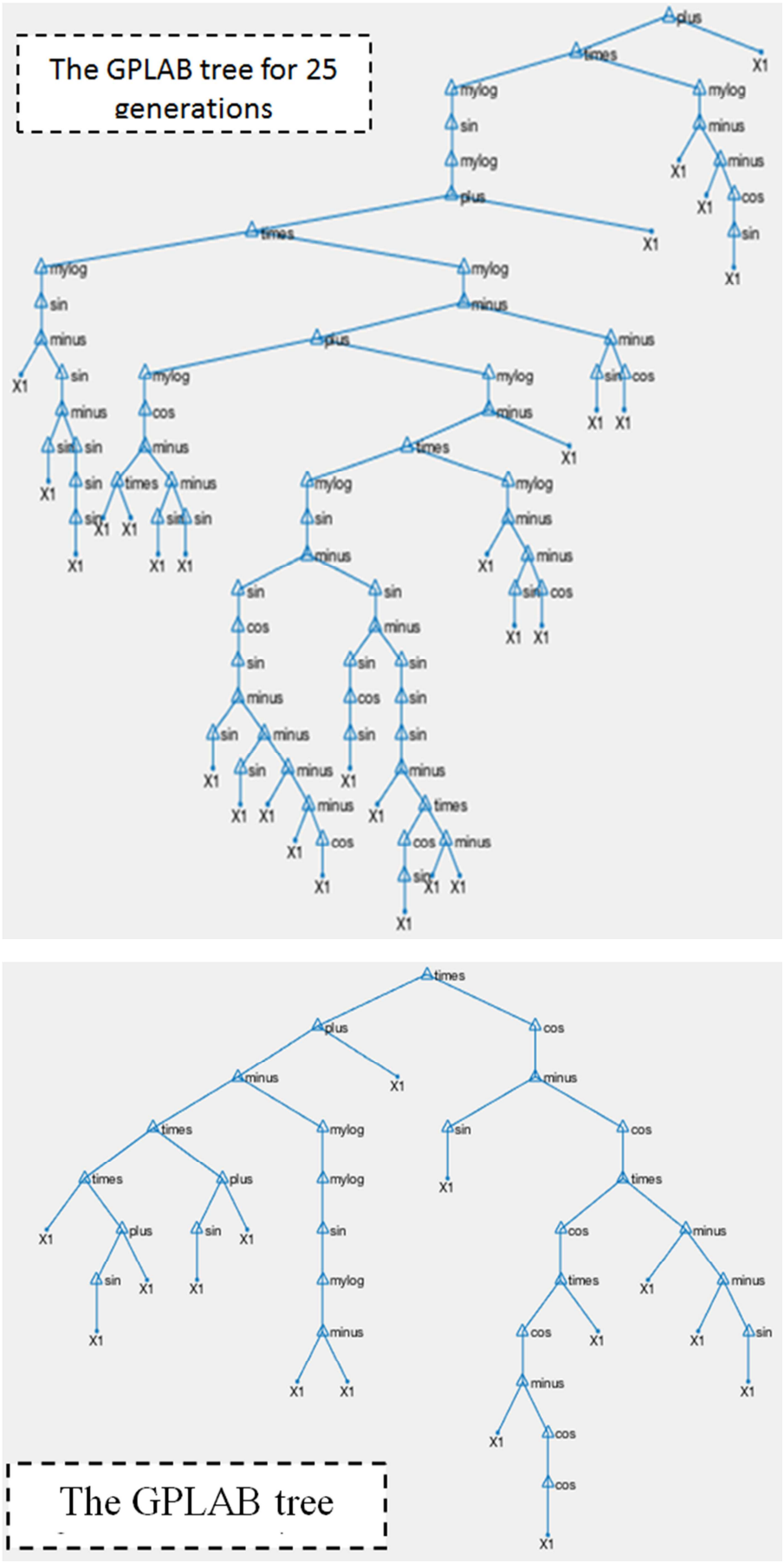

Figure 10. *Results of the GA tree produced by the GPLAB algorithm for 100 individual populations over 25 generations (top) and 40 generations (bottom).*

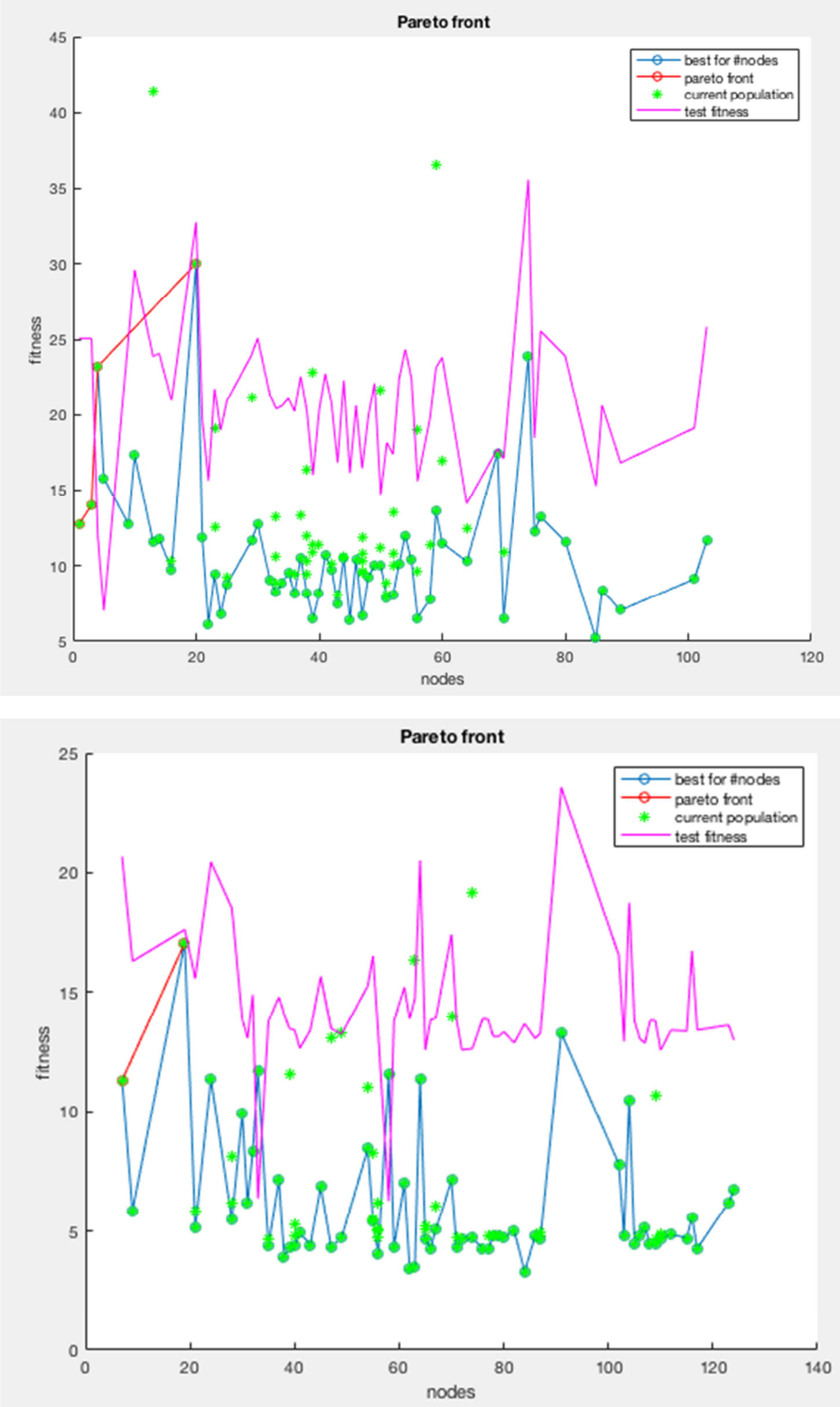

Figure 11. *The pareto front was developed for a population of 100 individuals, 40 generations, and 25 generations, respectively.*

The term "generations" refers to the number of cycles that have passed before the GA has ended. Depending on the complexity and problem kind, we may just need a few hundred loops in some circumstances, while we may need many more. Additionally, depending on the GA architecture, this parameter may not always be used, especially if deterministic criteria are used to determine when the GA should be terminated. The GA parameters are rigidly defined regardless of whether it discovers an efficient, more or less efficient, or optimal solution. The GA result is affected positively or negatively by changes (increases or decreases) in the value of these parameters, thus choosing the appropriate parameters is a difficult process. To ascertain the

depth and nodes of the GA, the test was run for 25 generations (top), 40 generations (bottom), and 100 individuals during the simulated activities. According to the results, which are shown in Figure 10, there were dynamic generation of various depths and nodes utilizing the same generation and individual population for the first, second, and so on.

According to the findings shown in Figure 11, a Pareto front algorithm (functional) was computed for 40 generations and 100 individual populations using over 100 nodes and 40 fitness (top). The green data point represents the current population, the red line the pareto front, the blue line the best for the number of nodes, and the pink line the fitness test. As a result, test fitness performance improves and the resultant prediction becomes better as the number of nodes increases. Additionally, fitness functions are better and heart disease experiments against the GA/GP in our simulation exercises are more accurate where the current population of data points is large. It should be emphasized, nevertheless, that as the algorithm is run more frequently, a more diversified pareto front is obtained, which enhances the program's forecast.

Table 6. *Experimental results showing 25 and 50 generations with 100 individual population.*

Experimental Computation	Number of Generation	Number of Individuals	Used Resources	Best obtained so far	Fitness	Test Fitness	Depth	Number of Nodes
1st Results	Initial Gen.		1024	67	10.103316	10.808240	6	16
2nd Results			1024	27	12.766600	25.066600	1	1
1st Results	1		1321	67	10.103316	10.808240	6	16
2nd Results			1116	146	10.842765	12.108143	5	6
1st Results	2		1220	232	8.895924	18.178717	4	7
2nd Results			988	243	10.368153	18.178717	6	15
1st Results	3		958	232	8.895924	18.178717	4	7
2nd Results			979	243	10.368153	18.178717	6	15
1st Results	4		856	448	6.897281	19.165462	10	24
2nd Results			1134	450	9.4726114	19.907328	14	39
1st Results	5		793	448	6.897281	19.165462	10	24
2nd Results			942	530	6.050000	17.366600	3	5
1st Results	6		582	581	6.667338	18.446664	6	11
2nd Results			976	530	6.050000	17.366600	3	5
1st Results	7		528	581	6.667338	18.446464	6	8
2nd Results			1489	530	6.050000	17.366600	3	14
1st Results	8		672	753	6.387716	12.654220	4	8
2nd Results			1930	817	4.799050	13.788685	6	14
1st Results	9		766	753	6.387716	12.654220	4	8
2nd Results			2877	817	4.799050	13.788685	6	14
1st Results	10		954	753	6.387716	12.387716	4	8
2nd Results			3096	1015	4.576328	15.511013	18	47
1st Results	11		889	753	6.387716	12.654220	4	8
2nd Results			3319	1059	4.335451	15.273556	18	43
1st Results	12		998	753	6.387716	12.654220	4	8
2nd Results		100 individual	3608	1153	3.419405	15.799956	14	41
1st Results	13	population	1180	1304	5.901219	14.531697	6	15
2nd Results			3511	1153	3.419405	15.799956	14	41
1st Results	14		1470	1304	5.901219	14.531697	6	15
2nd Results			3511	1153	3.419405	15.799956	14	41
1st Results	15		1430	1444	3.521069	14.474716	8	17
2nd Results			3483	1153	3.419405	15.799956	14	41
1st Results	16		1530	1444	3.521069	14.474716	8	17
2nd Results			3451	1416	3.302804	13.774998	14	31
1st Results	17		1721	1647	2.468834	14.523477	12	23
2nd Results			3616	1519	2.712197	14.537375	14	34
1st Results	18		1657	1647	2.468834	14.523477	12	23
2nd Results			3848	1631	2.646642	14.537375	18	55
1st Results	19		1787	1647	2.468834	14.523477	12	23
2nd Results			3787	1631	2.646642	14.537375	18	55
1st Results	20		2060	1647	2.468834	14.523477	12	23
2nd Results			3598	1631	2.646442	14.537375	18	55
1st Results	21		2275	2064	1.902202	16.358613	11	22
2nd Results			3533	1631	2.646642	14.537375	18	55
1st Results	22		2352	2064	1.902202	16.358613	11	22
2nd Results			3478	1631	2.646642	14.537325	18	55
1st Results	23		2368	2064	1.902202	16.358613	11	22
2nd Results			3697	1631	2.646642	14.537378	18	55
1st Results	24		2368	2064	1.902202	16.358613	11	22
2nd Results			3908	1631	2.646642	14.537375	18	55
1st Results	25		2212	2064	1.902202	16.358613	11	22
2nd Results			3903	1631	2.646642	14.537375	18	55

For 25 generations (on the left) and 40 generations (on the right), respectively, of heart disease sufferers with ages ranging from 0 to 90 years, the GA or the GPLAB algorithm was employed to make the heart prediction. The fitness line is coloured blue, the number of nodes is yellow (showing the ages vs generations), and the red line shows the degree of intricacy between the two factors (fitness and nodes). According to the data, heart disease affects more people between the ages of 10 and 35 in generation 5 than it does in generation 15. (less than 10 years old). However, cardiac illnesses affect those in generations 20 and 25, who are 40 to 90 years old. On the other side, as more generations are added to the algorithm, the complexity of the algorithm rises, making it intuitively predictable that an aging population will be more susceptible to heart disease and make it much harder to survive infectious diseases like the COVID-19 Pandemic. In contrast to generation 40, where the aging population declines for ages greater than 40 years and so on, we discovered that the findings are similar when we compared the top and bottom statistics (Figure 12).

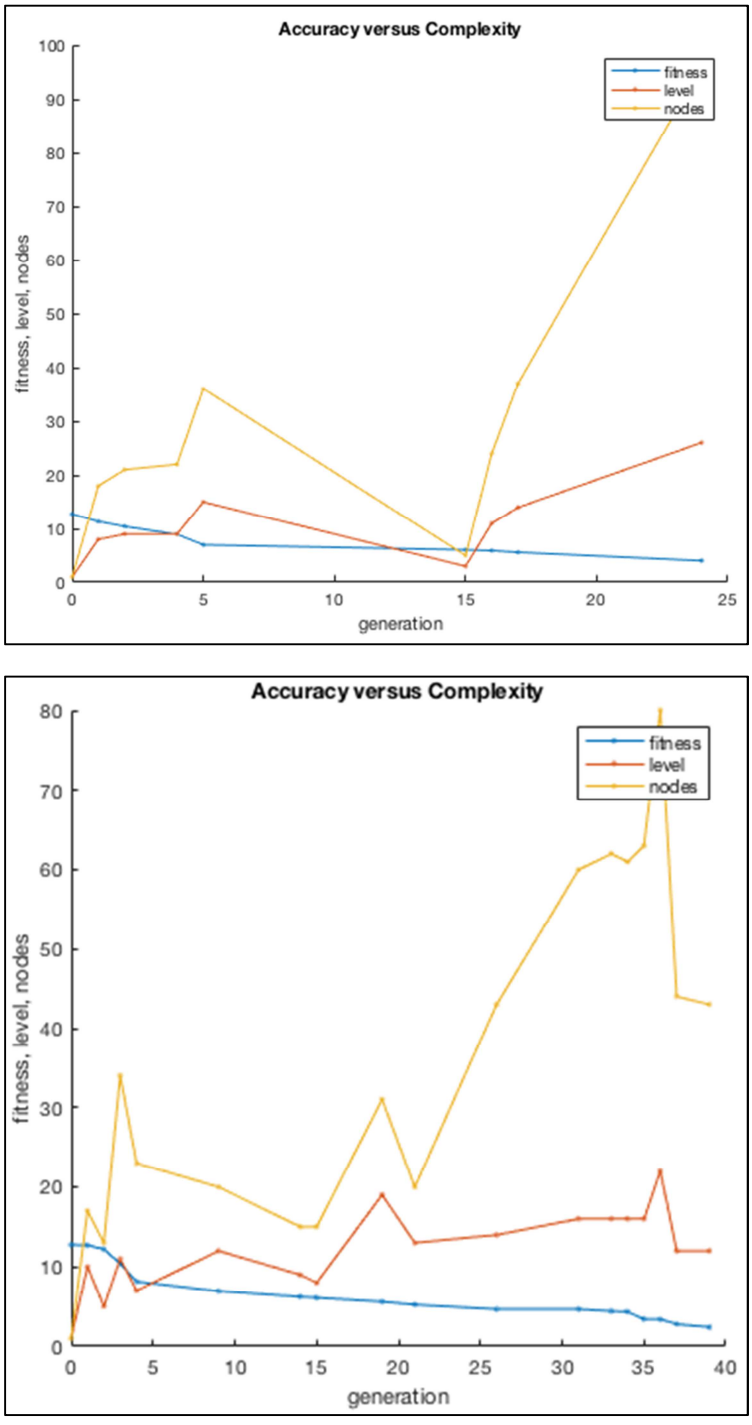

Figure 12. *Accuracy versus complexity for 100 distinct populations over 25 generations.*

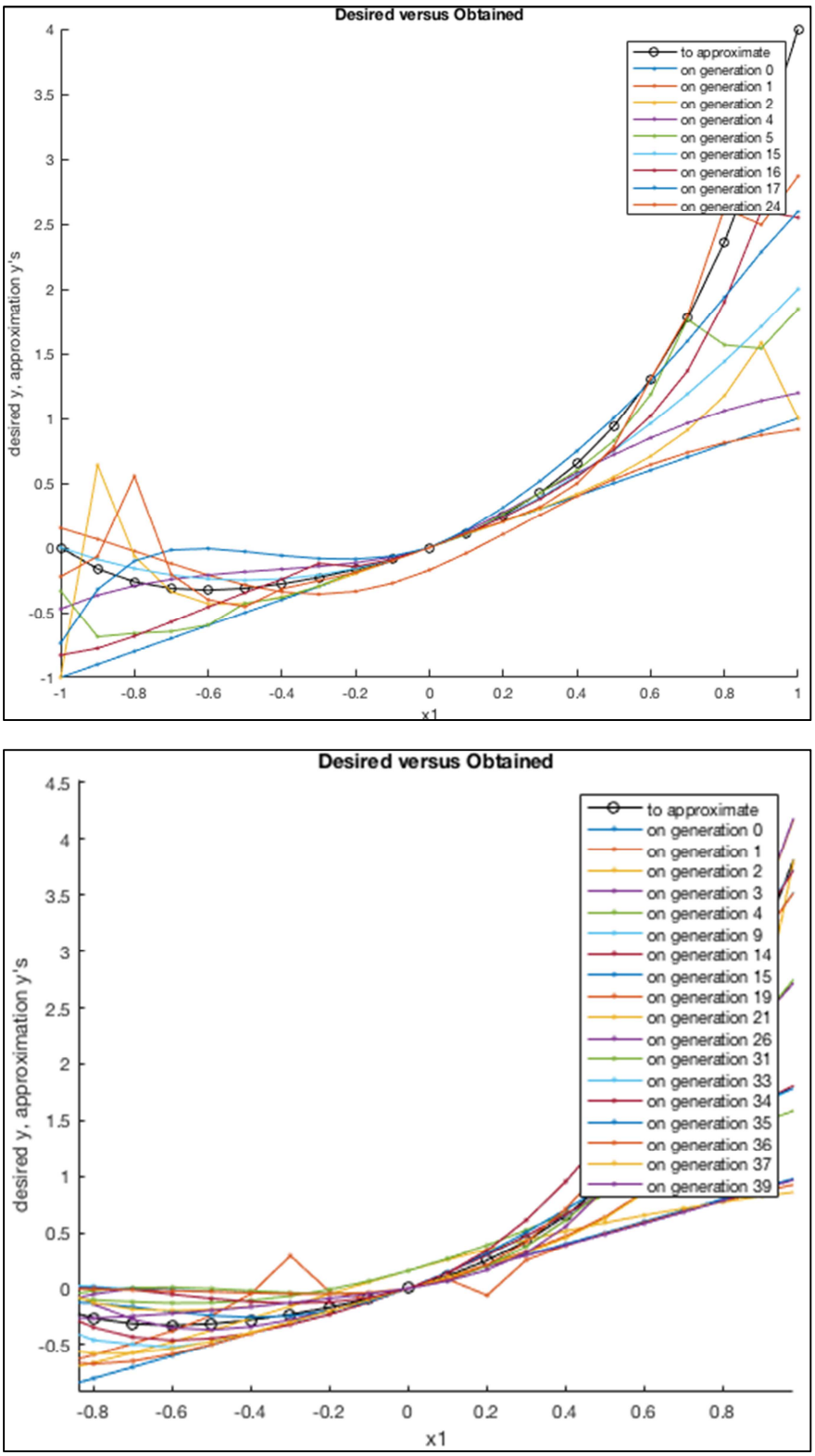

Figure 13. *The GPLAB algorithms for 25 generation (top) and 40 generation (bottom), each with 100 individual population, are intended vs obtained.*

This function was programmed to calculate the probabilities between -1 and +1 for generations 25 and 40. (Figure 13). We discovered that the better the generation, the better the desired obtained algorithm by comparing generation from other generations. This provided more explanation and demonstrated the Darwinian theory's inescapable conclusion. The likelihood of having descendants that can perform with greater accuracy and performance is higher at generations 24 (top) and 39 (bottom) as compared to lower generations for 100 individuals.

Using 25 and 40 generations, the probability calculation was carried out as follows: the light blue line represents the

maximum fitness (5.1958 and 3.2896, respectively), the green line represents the median (10.5403 and 4.9774, respectively), the orange line represents the average (11.979 and 6.9774, respectively), the negative broken orange line represents the average standard deviation (6.2613 and 6.4129, respectively), and the positive broken orange line represents the average standard deviation (17.6967 and 9.6493, respectively) (15.0958 and 17.2177, respectively). When measured against test fitness, standard deviation, maximum depth, and greatest probability estimation to date, the algorithm performs better at generation 40 than it did at generation 25. (Figure 14). In contrast, the data are more evenly distributed or spread out at a certain point at 25 generations than they are at 40 generations, where they are clustered around the mean (3.1764 and 9.6493). (6.2614 and 17.6967).

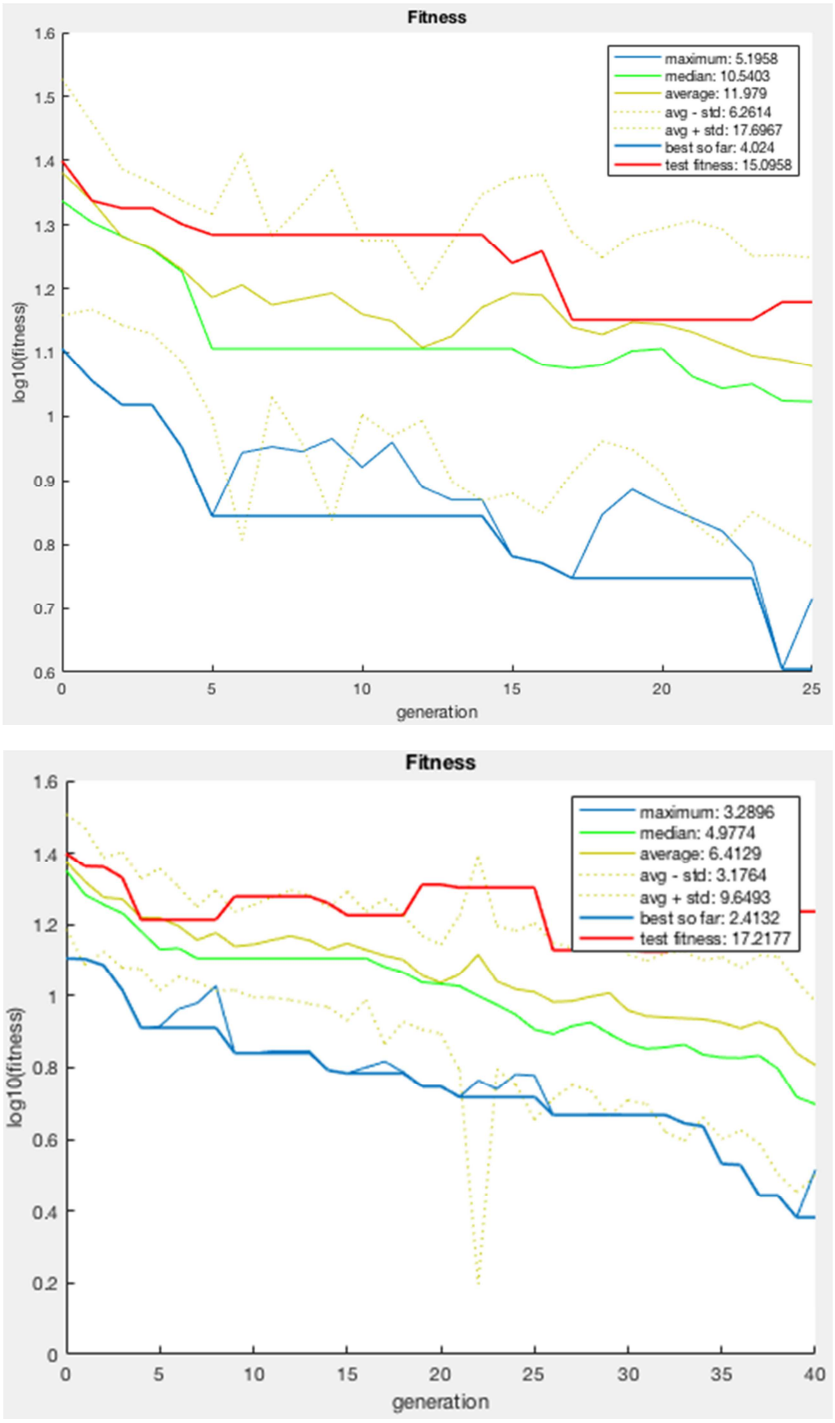

Figure 14. *Fitness function utilizing 100 individual population members for 25 generations (top) and 40 generations (bottom).*

Using 25 and 40 generations, the probability calculation was carried out as follows: the light blue line represents the maximum fitness (5.1958 and 3.2896, respectively), the green line represents the median (10.5403 and 4.9774,

respectively), the orange line represents the average (11.979 and 6.9774, respectively), the negative broken orange line represents the average standard deviation (6.2613 and 6.4129, respectively), and the positive broken orange line represents the average standard deviation (17.6967 and 9.6493, respectively) (15.0958 and 17.2177, respectively). When measured against test fitness, standard deviation, maximum depth, and greatest probability estimation to date, the algorithm performs better at generation 40 than it did at generation 25 (Figure 15). In contrast, the data are more evenly distributed or spread out at a certain point at 25 generations than they are at 40 generations, where they are clustered around the mean (3.1764 and 9.6493). (6.2614 and 17.6967).

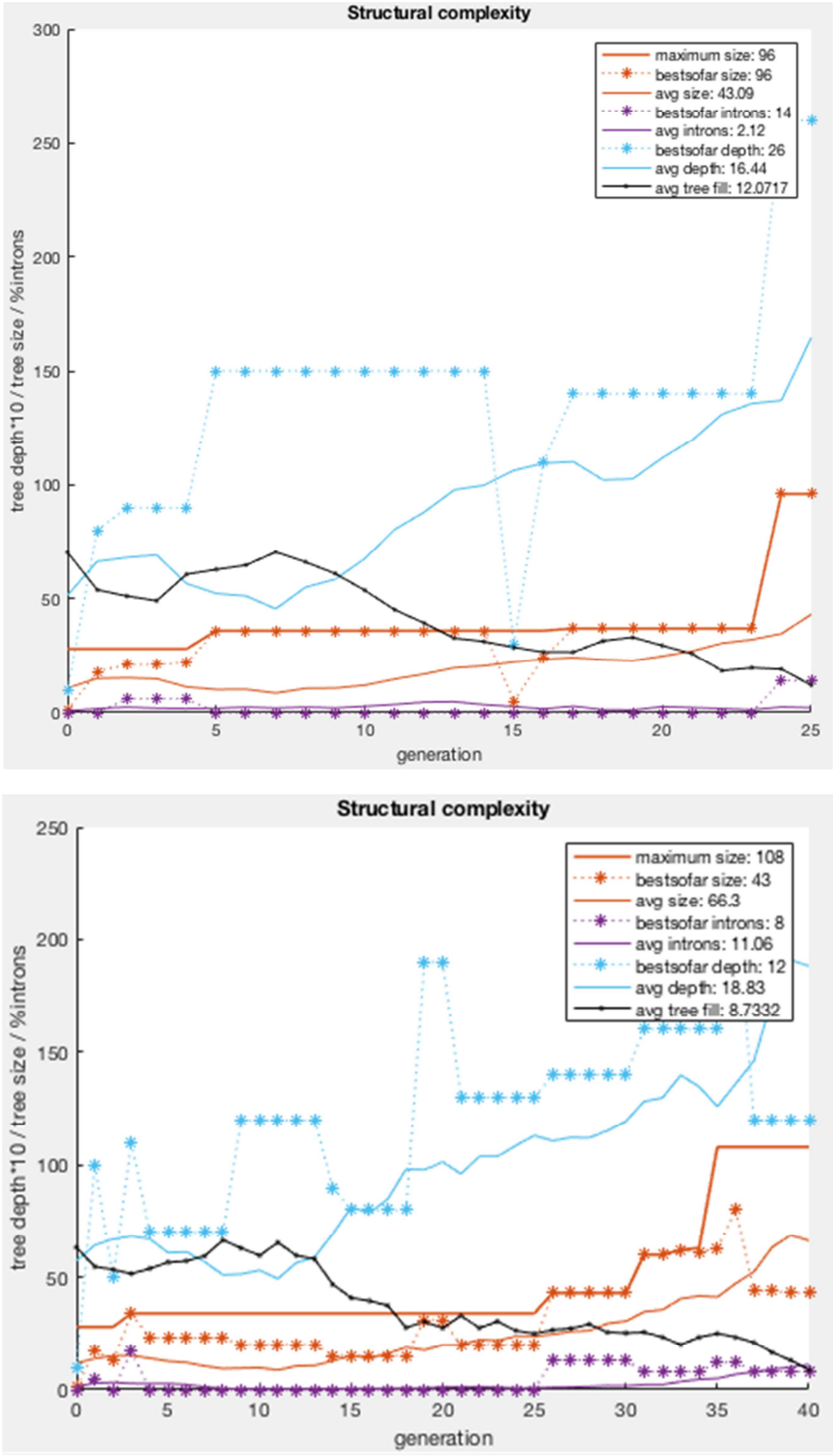

Figure 15. *The GA structural complexity when tested for 25 generations (top) and 40 generations (bottom) using 100 individual population.*

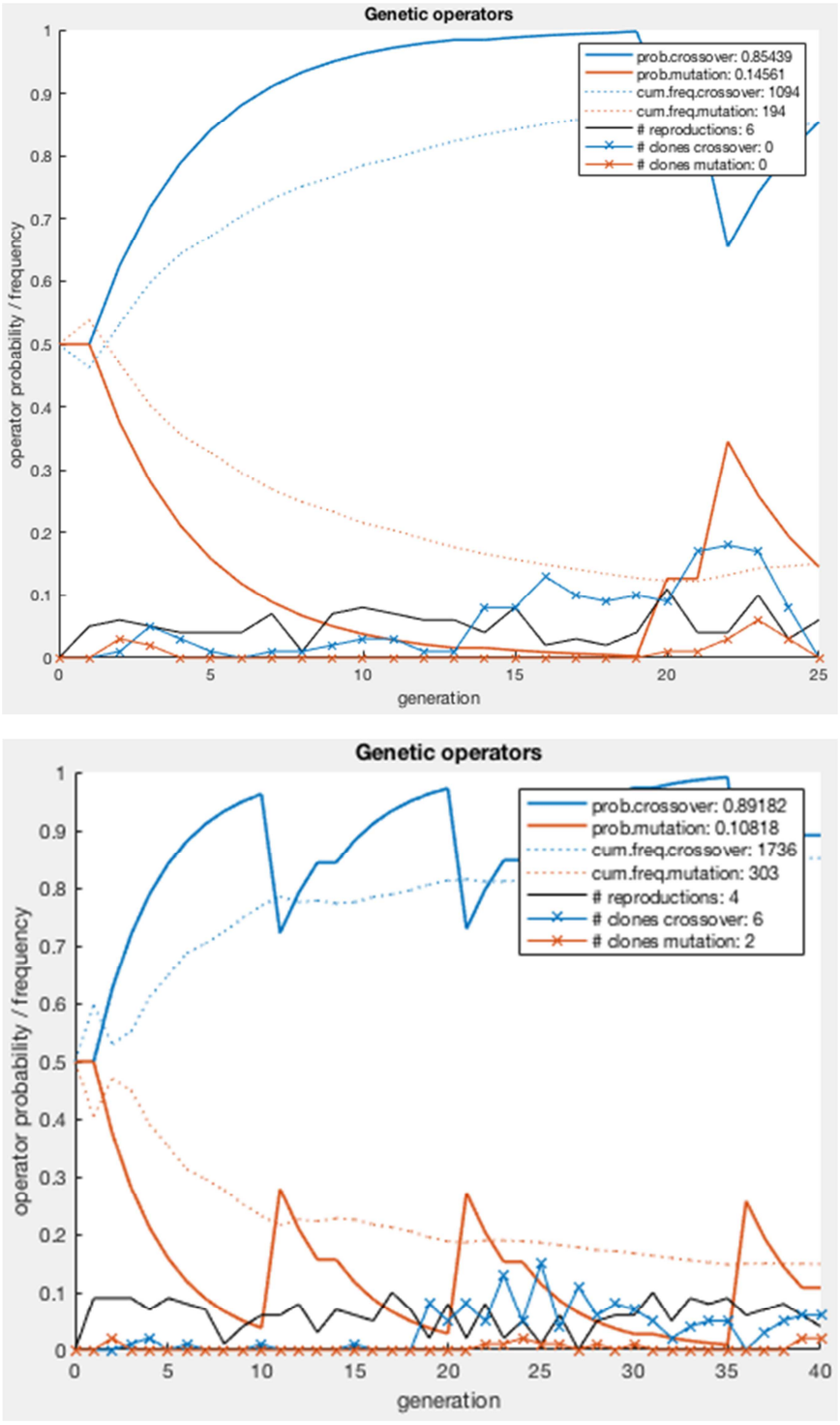

Figure 16. *The genetic operators for 25 generations (top) and 40 generations (bottom) using 100 individual population.*

The simulation's findings show that the architecture of the GA algorithm becomes more complex the greater the generational computation for a certain set of individual population. The experiment involved comparing 25 and 40 generations to mathematical functions represented by different coloured lines: the maximum size was plotted by a thick orange line, the best size result so far was plotted by a dotted orange line with an asterisk, the best result so far for introns was plotted by a mauve line, the best depth result so far was plotted by a blue line with an asterisk, the average depth was plotted by a blue line, and the average tree fill was plotted The greatest size complexity at generation 40 is 108, whereas the maximum size complexity at generation 25 is 96 (Figure 15).

The heart disease dataset is used to execute the genetic operators in this simulation against 25 and 40 generations. The forecasts made at the 40th generation are as follows: The thick green line represents the number of reproductions, the line in blue with x represents the number of done crossovers

(6), and the line in orange with x represents the number of done mutation (2). The line in blue indicates the probability of crossover (89.2%), the line in orange displays the probability of mutation (10.8%), the dotted blue line represents the cumulative frequency of crossover (1736), and the dotted orange line represents the cumulative frequency of mutation (303). This indicates that the majority of offspring are produced by crossover at a rate of (85-89%; [0, 1]), whilst between (10-14%) suggests the likelihood of how many chromosomes should be altered in a single generation (Figure 16). The mutation's main goal is to stop the GA from converging to local optimum.

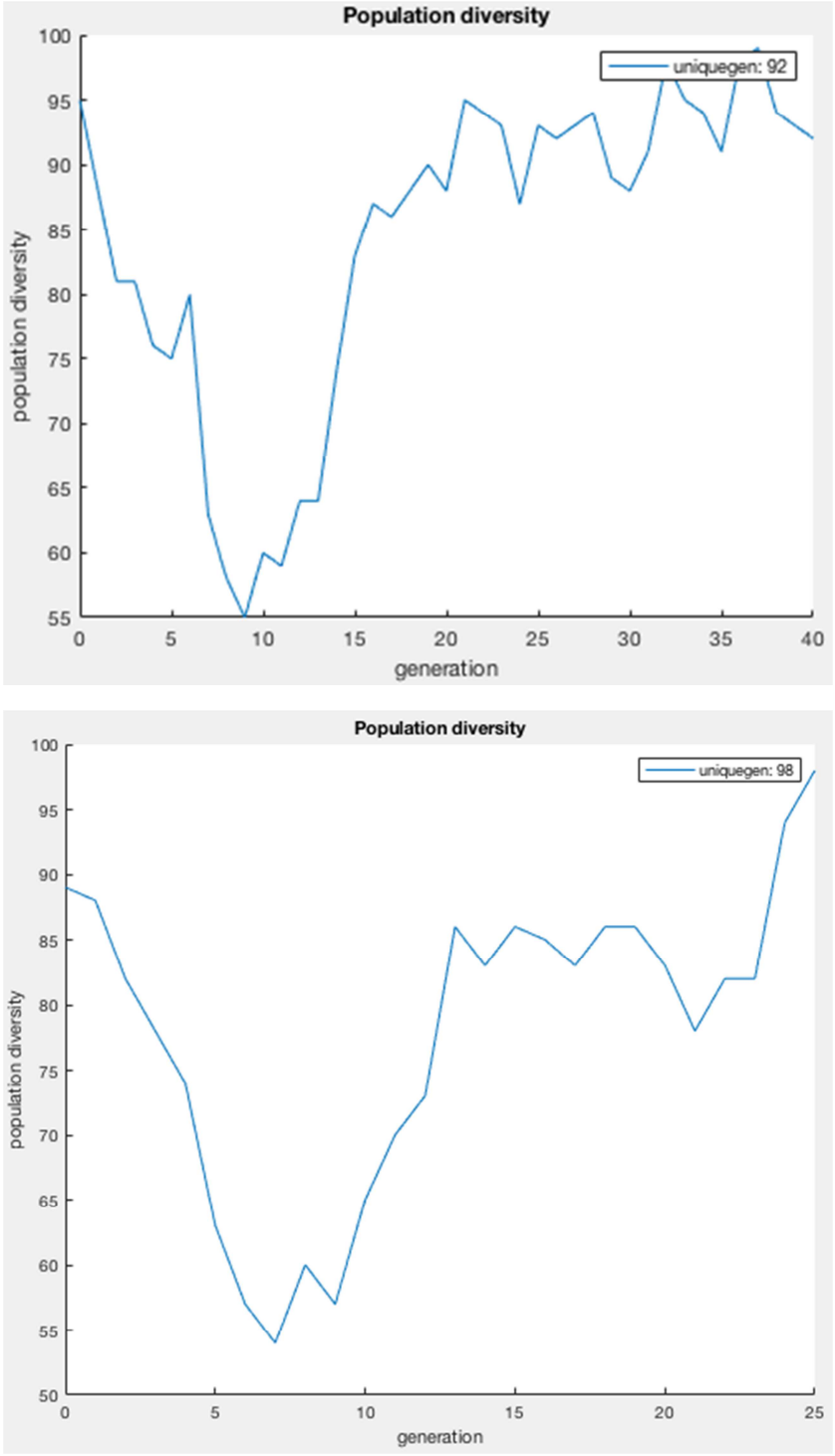

Figure 17. *Population diversity generated by the GA.*

The population diversity displays the population's total number of residents. By selecting the population size and sensitive diversity under the condition that the population search space is small, it is possible to achieve a local optimum. However, if the population is very large, the search area is expanded and the computational load increases,

therefore the population size must be fair (Figure 17). When comparing generations 25 and 40, the analysis reveals that persons who were more impacted by heart problems at generation 40 were also more affected at generation 25 between the ages of 85 and 95. However, in less than 5 generations, the population's age diversity decreases from 88 to 50 to 55 years, as calculated by generation 25 with a unique generation of 98 (bottom figure). Similar to this, the population variety decreases from a population of approximately 95 years to ages between 55 and 60 with a unique generation of 92 between 5 and 10 generations and less (top figure).

6. Conclusion and Research Contributions

The fundamental principles of genetic algorithms, their architecture, the management evolution of GA enhancement, and the evolution of desirable settings have all been thoroughly reviewed in relation to their applications to healthcare systems. To comprehend its architecture and modularization using GA operators, a theoretical and mathematical modeling was also presented. In this study, we provided a hybrid architecture and used the MATLAB programming environment to implement the algorithm. The GPLAB algorithm allows us to suggest a method that may accurately and adequately forecast heart problems (cardiovascular) for a variety of parameters.

These contributions are made by this paper:

1) A system that combines GA and ML algorithms in order to do symbolic regression and ultimately calculate the GPLAB method. Pareto front functionality, structural complexity, GP tree functionality, and desired achieved functionality, accuracy, and complexity are all included in the architecture.
2) To ascertain which generation performs better than the other, tests were conducted on fitness, depth, and the number of nodes.
3) To assess the structural complexity, computational cost, and flexibility of ANN-based approaches to PSO and ANFIS-algorithms.
4) A mathematical model to identify the different types of GA and how they apply to the healthcare system.

Abbreviation

Parkinson Disease (PD), Genetic Algorithms (GA), Machine Learning (ML), Receiver Operating Characteristics (ROC), Artificial Neural Network (ANN), Major Adverse Cardiac Event (MACE), Newtons Raphson (NR), Large for Gestational Age (LGA), Appropriate for Gestational Age (AGA), Fuzzy Logic with Support Vector Regression (FSVR), Backpropagation Neural Network (BNN), A GA-based on ANN (GANN), Non-Small Cell Lung Cancer (NSCLC), Human Immunodeficiency Virus (HIV), Highly Active Antiretroviral Therapy (HAART), Particle Swam Optimization (PSO), Automate Driver System (ADS), Analytical Hierarchy Process (AHP), Simple Inversion Mutation Operators (SIM), Real Coded Genetic Algorithms (RCGAs), Hybrid Genetic Algorithm and Machine Learning Algorithms (HGAMLA), Adaptive Neuro-fuzzy Inference System (ANFIS), Genetic Programming Toolbox for MATLAB (GPLAB).

Competing Interest

The authors declare that they have no competing interests.

Availability of Data and Material

We obtained the dataset from Kaggle (heart diseases) and social networking sites as described in the section above. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset. The data was experimented with targeted and output to determine the performance of the GP.

Acknowledgements

This work was undertaken independently by the authors and but affiliated with the Milton Margai Technical University, Freetown Sierra Leone and MTT Consulting Architects and Engineers Plc., Addis Ababa, Ethiopia.

References

[1] Betarbet R., Sherer T. B., and Greenamyre J. T., "Animalmodels of Parkinson's disease," BioEssays, vol. 24, no. 4, pp. 308–318, 2002.

[2] Hughes A. J., S. Daniel E., Kilford L. Lees J., "Accuracy of clinical diagnosis of diopathic Parkinson's disease: a clinicopathological study of 100 cases," BritishMedical Journal, vol. 55, pp. 181–184, 1992.

[3] Little M. A., McSharry P. E., Hunter E. J., Spielman J., and Ramig L. O., "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," IEEE Transactions on Biomedical Engineering, vol. 56, no. 4, pp. 1015–1022, 2009.

[4] Boyanov B. and Hadjitodorov S., "Acoustic analysis of pathological voices: a voice analysis system for the screening and laryngeal diseases," IEEE Engineering in Medicine and Biology Magazine, vol. 16, no. 4, pp. 74–82, 1997.

[5] Godino-Llorente J. I. and G´omez-Vilda P., "Automatic detection of voice impairments by means of short-term cepstral parameters and neural network-based detectors," IEEE Transactions on Biomedical Engineering, vol. 51, no. 2, pp. 380–384, 2004.

[6] Huang W., Li N., Lin Z. et al., "Liver tumor detection and segmentation using kernel-based extreme learning machine," in Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '13), pp. 3662–3665, Osaka, Japan, July 2013.

[7] Osman I. H. and Kelly J. P., 1996. Meta-heuristics: an overview. Meta-heuristics, pp. 1-21.

[8] Dorigo M. and Stützle T., 2003. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In Handbook of metaheuristics (pp. 250-285). Springer, Boston, MA.

[9] Alrosan A., Alomoush W., Norwawi N., Alswaitti M. and Makhadmeh S. N., 2021. An improved artificial bee colony algorithm based on mean best-guided approach for continuous optimization problems and real brain MRI images segmentation. Neural Computing and Applications, 33 (5), pp. 1671-1697.

[10] Rodrigues, L. R., 2021. A chaotic grey wolf optimizer for constrained optimization problems. Expert Systems, p. e12719.

[11] Banupriya C. V. and Aruna D., 2021. D. Robust Optimization of electroencephalograph (EEG) Signals for Epilepsy Seizure Prediction by utilizing VSPO Genetic Algorithms with SVM and Machine Learning Methods. Indian J. Sci. Technol, 14, pp. 1250-1260.

[12] Hassani Z. I. M., El Barkany A., Jabri, A. Abbassi I. E. and Darcherif A. M., 2021. A Comparative Analysis of Metaheuristic Approaches (Genetic Algorithm/Hybridization of Genetic Algorithms and Simulated Annealing) for Planning and Scheduling Problem with Energy Aspect. SAE International Journal of Materials & Manufacturing, 14 (4), p. 363.

[13] Game P. S. and Vaze D., 2020. Bio-inspired Optimization: metaheuristic algorithms for optimization. arXiv preprint arXiv:2003.11637.

[14] Hussain A., Muhammad Y. S. and Sajid M. N., 2018. An efficient genetic algorithm for numerical function optimization with two new crossover operators. International Journal of Mathematical Sciences and Computing, 4 (4), pp. 41-55.

[15] Díaz, D., Valledor, P., Ena, B., Iglesias, M. and Menéndez, C., 2020. Improved Method for Parallelization of Evolutionary Metaheuristics. Mathematics, 8 (9), p. 1476.

[16] Sohail A., 2021. Genetic algorithms in the fields of artificial intelligence and data sciences. Annals of Data Science, pp. 1-12.

[17] Melanie M., An Introduction to Genetic Algorithms, A Bradford Book, The MIT Press, Cambridge, Mass, USA, 5th edition, 1999.

[18] D. Whitley, "An executable model of a simple genetic algorithm," Foundations of Genetic Algorithms, vol. 2, no. 1519, pp. 45–62, 2014.

[19] El-Sawy A. A., Hussein M. A., Zaki E. M. and Mousa A. A. An Introduction to Genetic Algorithms: A survey. A practical Issues. Int J Sci Eng Res, 5 (1), pp. 252-262, 2014.

[20] Ivanov, M. V., Talipov, M. R. and Timerghazin, Q. K., 2015. Genetic algorithm optimization of point charges in force field development: challenges and insights. *The Journal of Physical Chemistry A*, *119* (8), pp. 1422-1434.

[21] Katoch, S., Chauhan, S. S. and Kumar, V., 2021. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80* (5), pp. 8091-8126.

[22] Beg, A. H. and Islam, M. Z., 2016, June. Advantages and limitations of genetic algorithms for clustering records. In *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)* (pp. 2478-2483). IEEE.

[23] Yuret, D. and De La Maza, M., 1993, June. Dynamic hill climbing: Overcoming the limitations of optimization techniques. In *The second Turkish symposium on artificial intelligence and neural networks* (pp. 208-212). Citeseer.

[24] Serpik, I. N., Alekseytsev, A. V. and Balabin, P. Y., 2017. Mixed approaches to handle limitations and execute mutation in the genetic algorithm for truss size, shape and topology optimization. *Periodica polytechnica civil engineering*, *61* (3), pp. 471-482.

[25] Elbaz, K., Shen, S. L., Zhou, A., Yuan, D. J. and Xu, Y. S., 2019. Optimization of EPB shield performance with adaptive neuro-fuzzy inference system and genetic algorithm. *Applied Sciences*, *9* (4), p. 780.

[26] Muniyappan S. and Rajendran P., 2019. Contrast enhancement of medical images through adaptive genetic algorithm (AGA) over genetic algorithm (GA) and particle swarm optimization (PSO). Multimedia Tools and Applications, 78 (6), pp. 6487-6511.

[27] Lee C. S., Moy L., Hughes D., Golden D., Bhargavan-Chatfield M., Hemingway J., Geras A., Duszak R. and Rosenkrantz A. B., 2021. Radiologist Characteristics Associated with Interpretive Performance of Screening Mammography: A National Mammography Database (NMD) Study. Radiology, 300 (3), pp. 518-528.

[28] Zebari D. A., Zeebaree D. Q., Abdulazeez A. M., Haron H. and Hamed H. N. A., 2020. Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images. Ieee Access, 8, pp.203097-203116.

[29] Pereira DC, Ramos RP, do Nascimento MZ. Segmentation and detection of breast cancer in mammograms combining wavelet analysis and genetic algorithm. Comput Methods Programs Biomed 2014 Apr; 114 (1): 88-101.

[30] Soulami K. B., Saidi M. N., Honnit B., Anibou C. and Tamtaoui A., 2019. Detection of breast abnormalities in digital mammograms using the electromagnetism-like algorithm. Multimedia Tools and Applications, 78 (10), pp. 12835-12863.

[31] Oza P., Sharma P., Patel S. and Bruno A., 2021. A bottom-up review of image analysis methods for suspicious region detection in mammograms. Journal of Imaging, 7 (9), p. 190.

[32] Acevedo P. and Vazquez M., 2019, December. Classification of tumors in breast echography using a SVM algorithm. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 686-689). IEEE.

[33] Zhang Z., Trevino V., Hoseini S. S., Belciug S., Boopathi A. M., Zhang P., Gorunescu F., Subha V. and Dai S., 2018. Variable selection in Logistic regression model with genetic algorithm. Annals of translational medicine, 6 (3).

[34] Kausar T., Ashraf M. A., Kausar A. and Riaz I., 2021, January. Convolution neural network-based approach for breast cancer type classification. In 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST) (pp. 407-413). IEEE.

[35] Zhou J, Krishnan S, Chong V, Huang J, eds. Extraction of tongue carcinoma using genetic algorithm-induced fuzzy clustering and artificial neural network from MR images. Engineering in Medicine and Biology Society, 2004 IEMBS'04 26th Annual International Conference of the IEEE; 2004: IEEE.

[36] Heng H. P. S., Shu C., Zheng W., Lin, K. and Huang Z., 2021. Advances in real-time fiber-optic Raman spectroscopy for early cancer diagnosis: Pushing the frontier into clinical endoscopic applications. Translational Biophotonics, 3 (1), p. e202000018.

[37] Balakrishnan R. and Karthikeyan T., 2019. Microarray gene expression and multiclass cancer classification using extreme learning machine (ELM) with refined group search optimizer (RGSO). Int Sci J Sci Eng Technol, 18.

[38] Dolled-Filhart M, Rydén L, Cregger M, Jirström K, Harigopal M, Camp RL, et al. Classification of breast cancer using genetic algorithms and tissue microarrays. Clin Cancer Res 2006 Nov; 12 (21): 6459-6468.

[39] Silva E., Parente M., Brandão S., Mascarenhas T. and Natal Jorge R., 2019. Characterizing the biomechanical properties of the pubovisceralis muscle using a genetic algorithm and the finite element method. Journal of Biomechanical Engineering, 141 (1), p. 011009.

[40] Fofanah, A. J., Bundu, H. R. and Kargbo, J. G., A generic heart diseases prediction and application of genetic algorithms in healthcare systems: Genetic algorithm and machine learning algorithm approaches.

[41] Hoh JK, Cha KJ, Park MI, Ting Lee ML, Park YS. Estimating time to full uterine cervical dilation using genetic algorithm. Kaohsiung J Med Sci 2012 Aug; 28 (8): 423-428.

[42] Boisvert MR, Koski KG, Burns DH, Skinner CD. Early prediction of macrosomia based on an analysis of second trimester amniotic fluid by capillary electrophoresis. Biomark Med 2012 Oct; 6 (5): 655-662.

[43] Yu J, Wang Y, Chen P. Fetal weight estimation using the evolutionary fuzzy support vector regression for low-birthweight fetuses. IEEE Trans Inf Technol Biomed 2009 Jan; 13 (1): 57-66.

[44] Gao J., Lyu T., Xiong F., Wang J., Ke W. and Li Z., 2021. Predicting the survival of cancer patients with multimodal graph neural network. IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[45] Rellum S. R., Schuurmans J., van der Ven W. H., Eberl S., Driessen A. H., Vlaar A. P. and Veelo D. P., 2021. Machine learning methods for perioperative anesthetic management in cardiac surgery patients: a scoping review. Journal of Thoracic Disease, 13 (12), p. 6976.

[46] Sahlol A. T., Abd Elaziz M., Tariq Jamal A., Damaševičius R. and Farouk Hassan O., 2020. A novel method for detection of tuberculosis in chest radiographs using artificial ecosystem-based optimisation of deep neural network features. Symmetry, 12 (7), p. 1146.

[47] Carlisle L. A., Turk T., Kusejko K., Metzner K. J., Leemann C., Schenkel C. D., Bachmann N., Posada S., Beerenwinkel N., Böni J. and Yerly S., 2019. Viral diversity based on next-generation sequencing of HIV-1 provides precise estimates of infection recency and time since infection. The Journal of infectious diseases, 220 (2), pp. 254-265.

[48] Alizadeh R., Rezaeian J., Abedi M. and Chiong R., 2020. A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities. Computers & Industrial Engineering, 139, p. 106106.

[49] Abadi M. Q. H., Rahmati S., Sharifi A. and Ahmadi M., 2021. HSSAGA: designation and scheduling of nurses for taking care of COVID-19 patients using novel method of hybrid salp swarm algorithm and genetic algorithm. Applied Soft Computing, 108, p. 107449.

[50] Gao F., Zhang Q., Han Z. and Yang Y., 2021. Evolution test by improved genetic algorithm with application to performance limit evaluation of automatic parallel parking system. IET Intelligent Transport Systems, 15 (6), pp. 754-764.

[51] Di Francescomarino C., Dumas M., Federici M., Ghidini C., Maggi F. M., Rizzi W. and Simonetto L., 2018. Genetic algorithms for hyperparameter optimization in predictive business process monitoring. Information Systems, 74, pp. 67-83.

[52] Jun-hua L. and Ming L., 2013. An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments. Optik, 124 (24), pp. 6780-6785.

[53] Peng Y., Luo X. and Wei W., 2014. A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation. International Journal of Control, Automation and Systems, 12 (3), pp. 670-679.

[54] Xia, Q., et at.: Test scenario design for intelligent driving system ensuring coverage and effectiveness. Int. J. Automot. Technol. 19 (4), 751–758 (2018).

[55] Gao F., et al.: Test scenario automatic generation strategy for intelligent driving systems. Math. Probl. Eng. 2019, 3737486 (2019).

[56] Saaty T. L., 1986. Axiomatic foundation of the analytic hierarchy process. Management science, 32 (7), pp. 841-855.

[57] Ishizaka A. and Labib A., 2011. Review of the main developments in the analytic hierarchy process. Expert systems with applications, 38 (11), pp. 14336-14345.

[58] Rudolph G., 1994. Convergence analysis of canonical genetic algorithms. IEEE transactions on neural networks, 5 (1), pp. 96-101.

[59] Gao F, Zhang Q, Han Z, Yang Y. Evolution test by improved genetic algorithm with application to performance limit evaluation of automatic parallel parking system. IET Intell Transp Syst. 2021; 15:754–764. https://doi.org/10.1049/itr2.12058.

[60] Michalewicz Z., 1999. Genetic Algorithms+ Data Structures= Evolution Programs. Springer-Verlag, 1999. Google Scholar Google Scholar Digital Library Digital Library.

[61] Arifin H. H., Ong H. K. R., Daengdej J. and Novita D., 2019, July. Encoding technique of genetic algorithms for block definition diagram using OMG SysML™ notations. In INCOSE International Symposium (Vol. 29, No. 1, pp. 218-232).

[62] Shanmugasundaram N., Sushita K., Kumar S. P. and Ganesh E. N., 2019. Genetic algorithm-based road network design for optimising the vehicle travel distance. International Journal of Vehicle Information and Communication Systems, 4 (4), pp. 344-354.

[63] Aszemi N. M. and Dominic P. D. D., 2019. Hyperparameter optimization in convolutional neural network using genetic algorithms. Int. J. Adv. Comput. Sci. Appl, 10 (6), pp. 269-278.

[64] Jennings P. C., Lysgaard S., Hummelshøj J. S., Vegge T. and Bligaard T., 2019. Genetic algorithms for computational materials discovery accelerated by machine learning. NPJ Computational Materials, 5 (1), pp. 1-6.

[65] Sato M. and Oyama A., 2021, December. Comparative Study of Crossovers for Decision Space Diversity of Non-Dominated Solutions. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 01-08). IEEE.

[66] Das A. K. and Pratihar D. K., 2021. Solving engineering optimization problems using an improved real-coded genetic algorithm (IRGA) with directional mutation and crossover. Soft Computing, 25 (7), pp. 5455-5481.

[67] Tang PH, Tseng MH (2013): Adaptive directed mutation for real-coded genetic algorithms. Appl Soft Comput 13 (1): 600–614.

[68] Ono I, Kobayashi S (1997): A real-coded genetic algorithm for functional optimization using unimodal normal distribution crossover. In: Back T (ed) Proceedings of the 7th international conference on genetic algorithms, ICGA-7. Morgan Kaufmann, East Lansing, MI, USA, pp 246–253.

[69] Deep K, ThakurM (2007) A new crossover operator for real coded genetic algorithms. Appl Math Comput 188: 895–911.

[70] Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: ICGA, pp 416–423. Morgan Kaufmann.

[71] Horn J, Nafpliotis N, Goldberg DE. (1994) A niched Pareto genetic algorithm for multiobjective optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, vol. 1, Piscataway, NJ: IEEE Service Center, p. 67–72.

[72] Coello CAC, Pulido GT (2001) A micro-genetic algorithm for multiobjective optimization. In: EMO, volume 1993 of lecture notes in computer science, pp 126–140. Springer.

[73] Harada T, Alba E (2020) Parallel genetic algorithms: a useful survey. ACM Computing Survey 53 (4): 1–39.

Biography

Abdul Joseph Fofanah is affiliated with the Department of Mathematics & Computer Science, Faculty of Environmental Sciences, Milton Margai Technical University. He has a double master's degree in computer science and software engineering, and he is currently pursuing his Ph.D. program at the Texila American University. He is a Senior Lecturer at MMTU, consultant for both local and international as data scientist and software developer. He has worked on multiple emergency response situations related to outbreaks and disasters in West Africa, Kenya, Nepal, Afgahistan, and Myanmar.

Tesyon Korjo Hwase is with the MTT Consulting Architects and Engineer's Plc. She has attained a degree in Electrical and Software Engineering and a master's degree in Software Engineering at Haramaya University and Nankai University, respectively. She has worked for 4yrs as a lecturer and researcher at Akaki Polytechnic College, Ethiopia. From 2021 to date, she has been working as Electrical and Software Engineer at MTT Consulting Architects and Engineer's Plc. During her work time at MTT consulting architects and engineer's plc she participates in Ethiopia mega projects as Electrical building designer and as well as a software engineer.