A Review of Genetic Programming Application for Data Modeling

J. O. Shonubi [1], D. B. Johnson [2], F. E. Onuodu [3]

[1] *Department of Computer Science, Federal Polytechnic Ekowe, Bayelsa State, Nigeria*

[2] *Department of Computer Science, Ignatius Ajuru University of Education, Rivers State, Nigeria*

[3] *Department of Computer Science, University of Port Harcourt, River State, Nigeria*

***Abstract*--- Genetic programming is a recent field in the family of Evolutionary Computing which is gaining wide recognition both theoretically and practically as it well suitably useable in domains that do not have clear solutions to the problems. This article reviews the use of genetic programming (GP) as an efficient tool to explore data modeling. The researchers implemented data modelling using Eureqa – a genetic programming application.**

Keywords: **artificial intelligence, evolutionary computing, genetic programming.**

I. INTRODUCTION

There are real world problems that are encountered in human endeavour which seems or are too difficult to solve. Some of these examples include travelling salesman or knapsack problem. Some of the problem increases as the problem size increases with no known feasible exact solution methods.

Evolutionary algorithm has become a popular approach to solving these complex problems by exploiting biological evolution following Darwin theory of evolution. Evolutionary Computing (EC) mimics or simulates Darwin's theory of biological evolution, adaptation and natural selection [1]. Evolutionary Computing forms the core standard for all evolutionary algorithm such as genetic algorithm (GA), genetic programming (GP), Evolutionary programming, evolution strategy, differential evolution as shown in fig. 1 below.

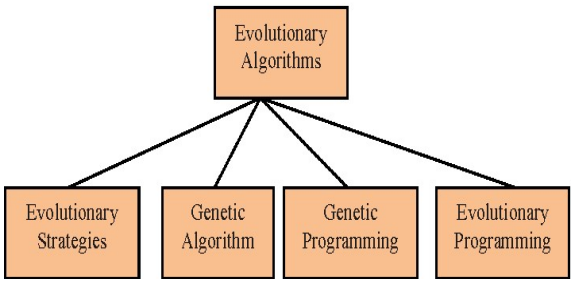

Fig. 1. Classification of Evolutionary Algorithm. Adapted from [2]

Reference [3] observed that "just as the way evolution and natural selection has resulted in the formation of organisms that are competent and best suitable inhabitants to live in any natural environment, the principle has been applied in computational science to evolve solutions to complex engineering problems which are subject to random and chaotic environments similar to the circumstances in which natural evolution has occurred".

Genetic programming is one of the branches of EC or EA which deals with the problem of optimization which is concerned with finding the best solutions to a problem. The candidate solutions which form the population are evaluated one after the other to find the best-fit solutions with a value that shows the quality of the solution by a fitness function. New generations are then formed with the best-fit following the law of natural selection referred to as survival-of-the-fit-test in biology. Genetic programming is similar to genetic algorithm in this aspect that it uses genetic operators such as selection, cross-over and mutation in its algorithms. However, the uniqueness of genetic programming is that it performs these operators over symbolic expression or formulae or programs rather than over numbers which represent the candidate solutions [4].

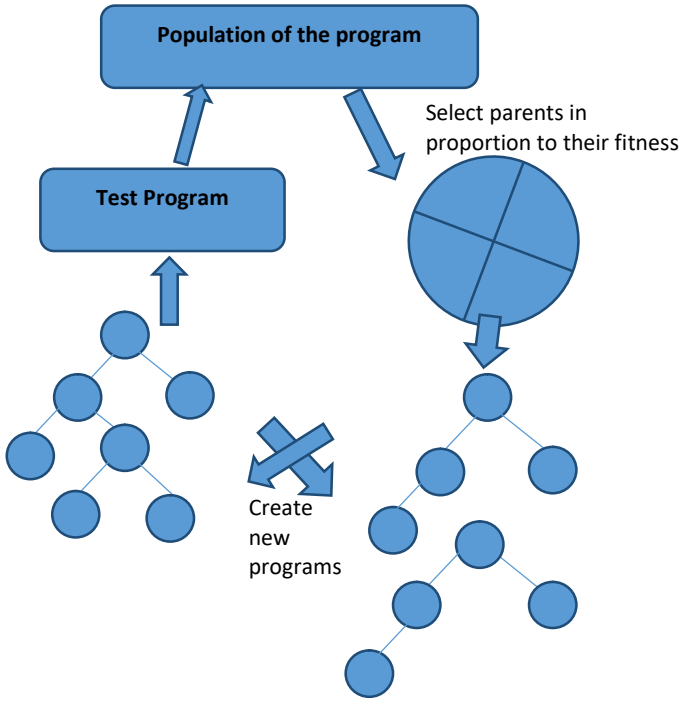

Fig. 2. Genetic Programming Cycle (taken from (Langley 1987)) Adapted from[5]

A. *Evolution of Candidates*

Candidates are evolved using evolutionary operators such as;

a. Crossover – creating new generations of solutions by combining the solutions of two parent-candidates. The new generation is the child of the previous generation. changing the sub-tree at the mark will produce new children.
b. Mutation – creating of new generation from a parent generation by changing the value of either the nodes (operators) or the leaves (terminals).

B. *Genetic Programming for data modelling*

Data modelling can be said to be the act of searching through a space for mathematical equations that will be used to model data to predict or provide solutions to problems that have no known way solving them. They provide a good fit for numerical data. [6] reveals that genetic programming is widely accepted since it can be successfully used to find a mathematical data model because it can find the shape of the equation as well as the composition of primitive functions, input variables and values of coefficients at the same time, without any additional information except the performance of the analyzed expression on the data. But the key issue to make GP algorithm work is a proper representation of a problem.

II. METHODOLOGY

The researchers made used of Eureqa genetic programming application to model data obtained during a simple pendulum experiment using time (t) and angle (Θ) as inputs.

i. *Terminals* – the terminal inputs is the time in seconds which range from (0.02 to 0.4s). 50 random numbers were generated for the time (t) using the excel function RANDBETWEEN.

ii. *Operators* – the researchers tired various types of operators in this research work which include (+, - , *, / , sin, cos).

A. *Fitness function*

Fitness function is used to measure the performance of candidate solutions individually and how well they fit into the problem

We used the fitness function $x = f(t)$ where

$x = angle\ of\ swing\ (measured\ in\ rad)$
$t = time\ duration\ (measured\ in\ \text{sec})$

The researchers exhaustively repeated this experiment while changing the operators/parameters and selected the following equation models from a list of various models due to the low complexity and the fitness of these models.

$x = \cos(t\text{^}2)/(t\text{^}2 * \cos(\cos(\cos(t\text{^}2)/t)))$ – (1)

$x = t\text{^}2/sin(t - sin(t))$ - (2)

These models become the parent-solutions (first generation). The parse tree for the mathematical models (equations) are drawn in fig. 3 and fig. 4 below;

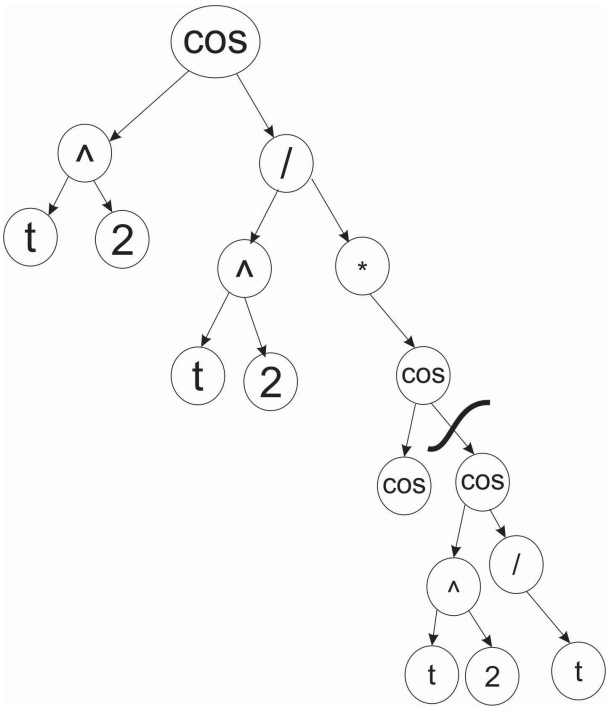

Fig. 3.Parse tree showing the first model.

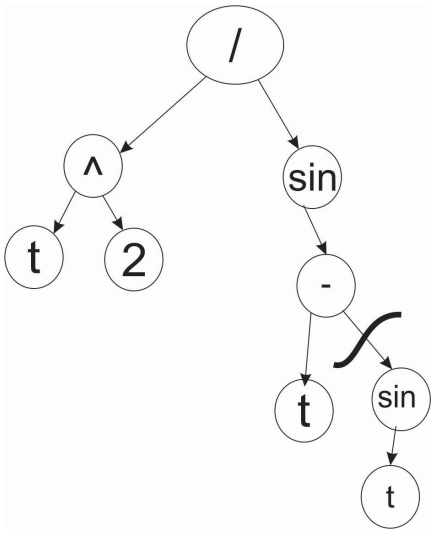

Fig.4.Parse tree showing the second model.

The researchers further applied the crossover operator to get newer generations of models by swapping the nodes at the cut shown on the first and second parent-solutions in fig 3 and fig. 4 above. This operation will breed new programs or models which are referred to as the children-solutions. The parse trees for the children generations shown in fig 5 and fig. 6 below.

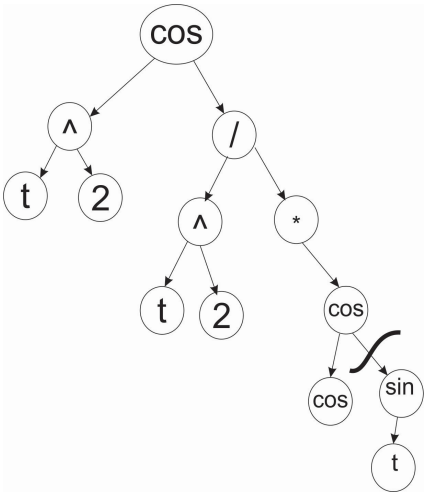

Fig. 5. Parse tree for child 1 after cross over

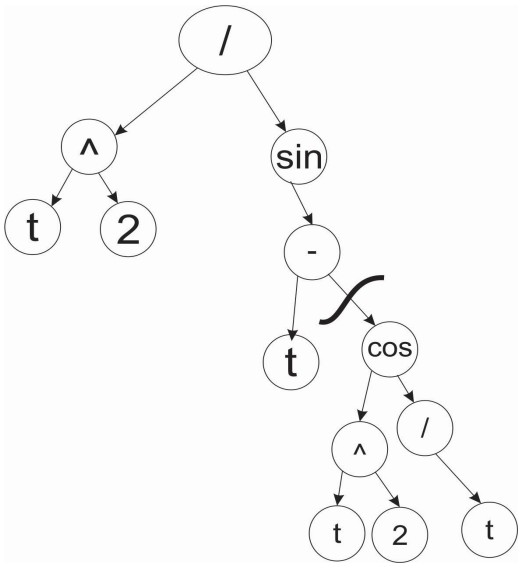

Fig. 6.Parse tree for child 2 after crossover

The equation derived for child 2 is given below;

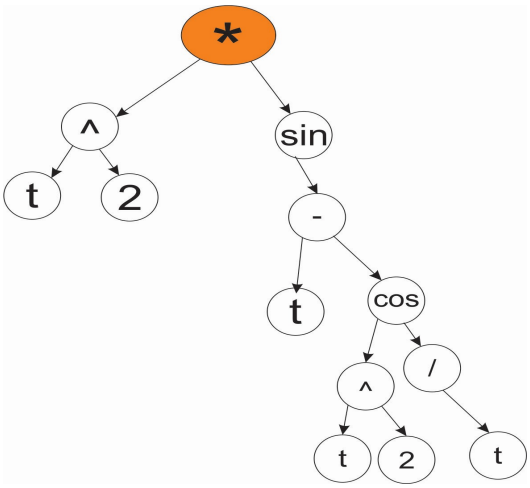

Fig. 7. Parse tree showing mutation on terminal

$$x = cos(t\text{^}2)/(t\text{^}2 * cos(cos(\sin(t)))) - (3)$$

$$x = t\text{^}2/sin(t - cos(t^2/t)) \quad - (4)$$

The equation 3 above is clearly a different solution from equation 1 because of the replacement of the branch node at the cut signified in fig. 3. Similarly, the equation 4 above is also different from equation 2 because of the new node introduced at the cut earlier specified in fig. 4.

In genetic programming, the nodes of parse trees represent the operators while he leaves of the tree represent the terminals. Therefore, the researchers further performed mutation operation to achieve newer generations by replacing a node(i.e operator/function) with another function and a leave (terminal) with another terminal. This operation was applied to the two children-solutions in fig. 5 and fig 6 to get newer children generations displayed below in fig. 7 and fig 8.

The mutation was done on the first-child changing the value of the terminal with the yellow colour from 2 to 3 as seen in the parse tree in fig. 7 above. The new equation for the model is given below;

$$x = cos(t\text{^}3)/(t\text{^}2 * cos(cos(\sin(t)))) - (5)$$

In the same vein, mutation was done on the second-child by changing the function of the node colored orange from division (/) to multiplication (*). The equation for the model is stated below;

$$x = t\text{^}2 * sin(t - cos(t^2/t)) \quad - (6)$$

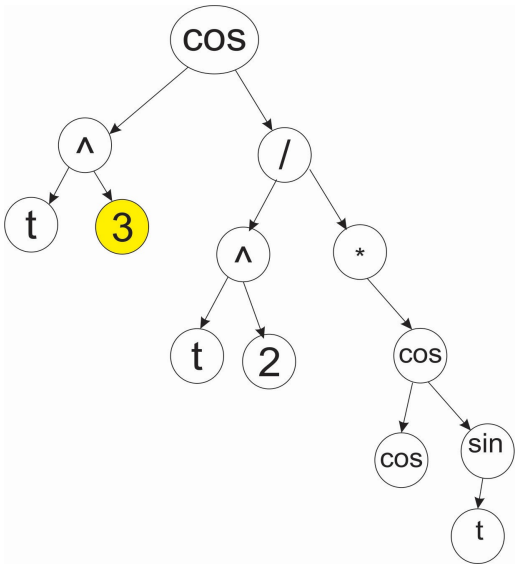

Fig. 8.Parse tree showing mutation on node

B. Findings

It was discovered during the experiment that the number of generation size increases with the fitness of the model or solution on the function (i.e. the fitter the solution, the higher the generations of the solution) as shown in the table below.

Table 1 – Maximum Error of the Chosen Models

Complexity	Fit	# Generations
31.443256	0.183	56
26.672194	0.276	30

The researchers plotted a graph of accuracy (error) against the complexity of the models referred to as pareto chart. Pareto chart shows the ratio of accuracy and complexity. From the observation made by [7] that "complex but accurate solutions will lie at the lower right, while simple but inaccurate solutions will lie at the top left. The most useful solutions are usually somewhere in the middle, striking the right balance between complexity and accuracy", the researchers were able to justify that the chosen models are within the range of good and accurate complexity as shown in the chart below.

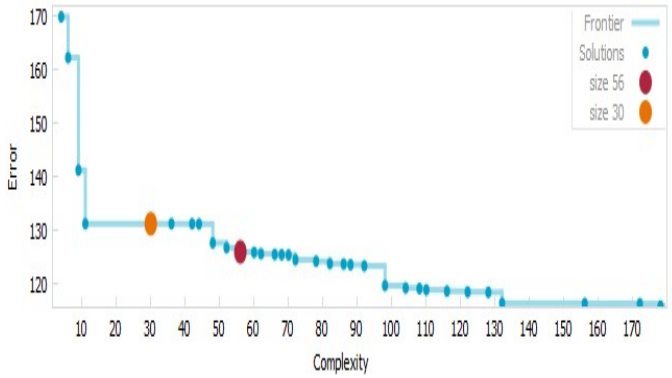

Fig 9. Error /Complexity Pareto for both models

Following [7]'s observation, it can be safely deduced that the two models chosen by the researchers are well suitable to model the data efficiently because both models lie in the middle of the chart.

III. CONCLUSIONS

In the research work, the researchers showed how genetic programming can be used to model real life data given dependent and independent variables. In the real sense, this may be difficult to achieve without implementing genetic programming.

REFERENCES

[1]. A. M. Alzohairy, "Darwin ' s Theory Of Evolution," no. October, 2014.

[2]. P. A. Vikhar, "Classification of Evolutionary Algorithms," *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Evolutionary-algorithms%3A-A-critical-review-and-its-Vikhar/9c4b8f3b53067a34cc13e8b5b3d54071327a1388.

[3]. J. Sreekanth and B. Datt, "Genetic Programming: Efficient Modeling Tool in Hydrology and Groundwater Management," in *Genetic Programming - New Approaches and Successful Applications*, InTech, 2012.

[4]. [4] M. Drahansky *et al.*, "We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %," *Intech*, vol. i, no. tourism, p. 13, 2016.

[5]. S. Winkler, M. Affenzeller, and S. Wagner, "Genetic Programming Based Model Structure Identification Using on-Line System Data," no. June 2014, 1995.

[6]. H. Kwasnicka and E. Szpunar-Huk, "Genetic programming in data modelling," *Stud. Comput. Intell.*, vol. 13, no. May 2006, pp. 105–130, 2006.

[7]. nutonian, "Viewing Model Results in Eureqa - Official User Guide | Nutonian." [Online]. Available: http://formulize.nutonian.com/documentation/eureqa/user-guide/view-results/. [Accessed: 10-Oct-2019].