SAKARYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
http://www.saujs.sakarya.edu.tr/en/

Title: A Review of Genetic Programming Popular Techniques, Fundamental Aspects, Software Tools and Applications

Authors: Davut ARI, Barış Baykant ALAGÖZ

Recieved: 2020-09-10 16:32:01

Accepted: 2021-02-15 23:00:48

Article Type: Research Article

Volume: 25
Issue: 2
Month: April
Year: 2021
Pages: 397-416

How to cite
Davut ARI, Barış Baykant ALAGÖZ; (2021), A Review of Genetic Programming Popular Techniques, Fundamental Aspects, Software Tools and Applications. Sakarya University Journal of Science, 25(2), 397-416, DOI: https://doi.org/10.16984/saufenbilder.793333
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/60672/793333

Sakarya University Journal of Science 25(2), 397-416, 2021

A Review of Genetic Programming: Popular Techniques, Fundamental Aspects, Software Tools and Applications

Davut ARI*[1], Barış Baykant ALAGÖZ[2]

Abstract

Genetic Programming (GP) is one of the evolutionary computation (EC) methods followed with great interest by many researchers. When GP first appeared, it has become a popular computational intelligence method because of its successful applications and its potentials to find effective solutions for difficult practical problems of many different disciplines. With the use of GP in a wide variety of areas, numerous variants of GP methods have emerged to provide more effective solutions for computation problems of diverse application fields. Therefore, GP has a very rich literature that is progressively growing. Many GP software tools developed along with process of GP algorithms. There is a need for an inclusive survey of GP literature from the beginning to today of GP in order to reveal the role of GP in the computational intelligence field. This survey study aims to provide an overview of the growing GP literature in a systematic way. The researchers, who need to implement GP methods, can gain insight of potentials in GP methods, their essential drawbacks and prevalent superiorities. Accordingly, taxonomy of GP methods is given by a systematic review of popular GP methods. In this manner, GP methods are analyzed according to two main categories, which consider the discrepancies in their program (chromosome) representation styles and their methodologies. Besides, GP applications in diverse problems are summarized. This literature survey is especially useful for new researchers to gain the required broad perspective before implementing a GP method in their problems.

Keywords: Genetic programming, gp types, gp applications, gp software

1. INTRODUCTION

GP is an EC type that allows computers to automatically formulate the solution of problems without making an assumption for the problem solution formulation [1-2]. Since the first appearance of the GP idea, it has been used to solve different analysis and modeling problems in different fields by using genetic coding techniques.

While spreading of the standard GP interdisciplinary domains, the researchers observed that computation capabilities of the standard GP cannot be sufficient for very hard

*Corresponding author: dari@beu.edu.tr
[1] Bitlis Eren University, Faculty of Engineering and Architecture, Computer Engineering Department,13100,Bitlis.
ORCID: https://orcid.org/ 0000-0001-6439-7957
[2] İnönü University, Faculty of Engineering, Computer Engineering Department,44280, Malatya.
E-Mail: baykant.alagoz@inonu.edu.tr.
ORCID: https://orcid.org/ 0000-0001-5238-6433

computational problems in different disciplines [3-5]. This becomes a central motivation for development of more advanced GP algorithms, and enhanced GP variants have been proposed to solve the difficult problems encountered in applications. Researchers mostly concentrated their research effort on improvements of representation formats of GP programs (individuals of the population).

Looking at the literature, it is seen that GP has been widely preferred in the problems that need symbolic regression for data modeling [6]. Accordingly, GP applications come out in many different disciplines such as classification problems[7], production scheduling [8], climate change analysis [9], energy and energy saving [10-12] besides educational technologies [13], urbanization [14], building [15], hydrology [16], medicine[17]. Additionally, GP has widely utilized in many computer sciences problems such as in computer vision[18], image processing [19], signal processing [20], artificial neural network design [21]. Moreover, GP methods were used in the field of evolutionary hardware [22] and circuit design [23]. One can find many application of GP in the field of economy e.g. finance [24] and trading [25] problems.

The use of GP methods in many disciplines and fields, not limited to the above-mentioned fields, shows that it is a popular and versatile calculation intelligence method.

This paper aims to review GP literature to gain a perfective on progress of GP algorithms, foundations of GP structures, popular GP trends and software tools without need of a deep knowledge of GP. For this reason, this study covers the introduction of GP essentials, review of popular methods and practices, and providing recent aspects on the latest developments on GP methodologies. Becoming a popular computing tool led the GP have a rich literature. The current study surveys the GP literature without complicating the topic by giving the deepened technical details of GP algorithms and tools.

Organization of the paper is as follows: Section 2 presents comparative introduction of fundamental GP methods. In Section 3, the taxonomy of GP methods is provided in two major branches that classify GP approaches according to discrepancies in their representation formats and their methodological differences. Section 4 presents an overview on considerable GP applications. Section 5 introduces the software tools that were developed and used by researchers.

2. GENETIC PROGRAMMING AND THE NATURE OF PROBLEMS TO APPLY

GP can be seen as a special EC method in which the individuals, also known as programs or chromosomes, in the population are basically represented in the form of computer programs. In GP, a computer program expresses a population individual that can be a candidate solution to a problem. As shown in Figure 1, in GP, the population of the programs, namely individuals, is generated randomly, as in other evolutionary computation methods [2]. Then, each candidate program (individual) created is tested for compatibility according to its ability to solve the problem. In the next stages, high-fitting computer programs are selected subjected to evolutionary processes such as crossover and mutation, thus enabling them to become more appropriate programs for the solution in the next generation. These evolution processes are repeated until the GP termination criteria are met.

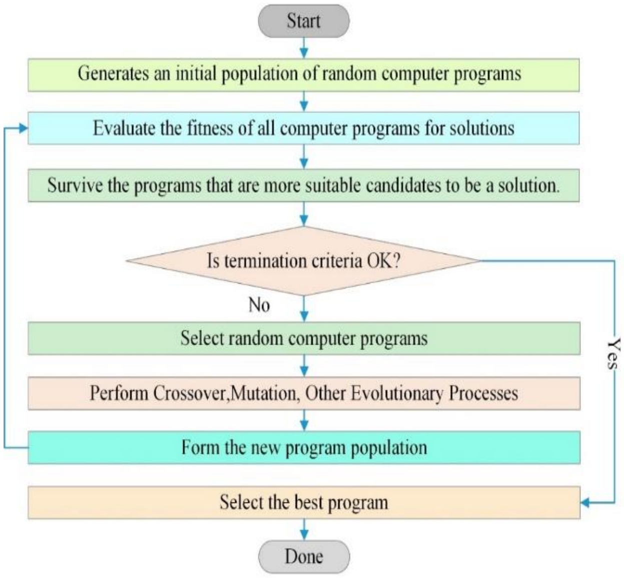

Start
Generates an initial population of random computer programs
Evaluate the fitness of all computer programs for solutions
Survive the programs that are more suitable candidates to be a solution.
Is termination criteria OK?
No
Yes
Select random computer programs
Perform Crossover,Mutation, Other Evolutionary Processes
Form the new program population
Select the best program
Done

Figure 1 The basic algorithm of Genetic Programming

The program, which has the best solution according to the needs of the targeted problem, continues to exist in time. Hence, it is the best fitting solution to the problem, which are able to maintain its existence until the termination criterion of the GP algorithm is met [26]. This property is known as elitism in evolutionary computation. The GP methods use elitist evolution strategies in order to reach better individuals and they commonly transfer individuals with higher fitness values to the next generations. It can contribute to reducing bloat problems of GP methods [27]. The main difference of the standard GP compared to Genetic Algorithm (GA) is that the program representation of GP can be variable-length instead of being the fixed-length representations such as bits, real numbers, and symbols in the GA [8], [26].

The GP method may be appropriate to use, especially if the interrelationships between the relevant variables in the problem are not known or poorly understood, or in case of a doubt that known relationship among the parameters may be wrong [28]. In other words, when the mathematical model is not known or valid, GP can provide acceptable solutions via model exploration and provide its best-fit solution to the problem.

In addition, if there are large amounts of data, which requires in-depth analysis, classification, and clustering on the computer, the GP can yield satisfactory results. The main reason for this advantage comes from the fact that individuals representing candidate solutions in GP are more flexible and adaptable than other metaheuristic methods.

3. POPULAR GENETIC PROGRAMMING APPROACHES

In the historical development of the GP, a tree-based structure was initially used by Koza [1]. In this review study, the most frequently used GP methods in the literature are introduced briefly and the works, devoted for introduction and analysis of the GP methods. These efforts help researchers to have an overview on a wide range of GP studies, and in case of need for deepened knowledge, they can easily reach the related papers that are considered in the scope of this study. For taxonomical organization of the literature knowledge, the hierarchy of GP methods is build in two main branches that are formed regarding the program representation formats and methodological differences.

3.1 Genetic Programming Types According to Program Representation Format

The most commonly used program representation formats of GP in the literature are tree, linear and graphical representations. For instance, one can consider standard GP [1] for tree-based representation, Cartesian Genetic Programming (CGP) [29] for graphical representation, and Linear Genetic Programming (LGP) [30] for linear representation.

3.1.1 Tree-Based Genetic Programming

The tree-based GP is the first and the most widely used representation format [1]. Hence, this type of representation is called the standard GP in the literature. Mainly, programs in software development processes composed of reusable program parts such as sub-functions, functions, and classes in the form of repeated steps. In contrast, in tree-based GP, population individuals are usually expressed by syntax trees rather than lines of code [2], [28].

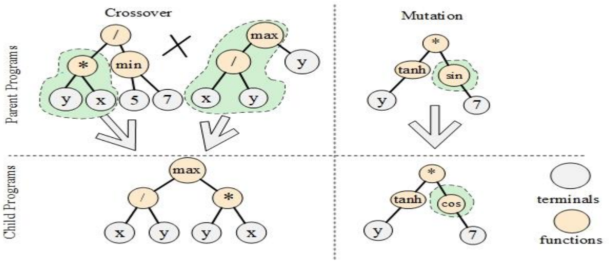


Figure 2 Standard (Tree-based) GP structure and its evolution steps

Figure 2 shows the tree-based GP structure and its evolution steps. For example, the tree representation of the program *x*y/min(5,7)* is processed in the evolutionary operation part of the figure. The variables and constants {*x*, *y*, *5*, *7*} in the program are called terminals in the leaves of the tree, while the arithmetic operations {*, / , min()} are called functions that are represented on internal nodes. In tree-based GP, program representations, namely candidate solutions, are formed by the placement of terminals and functions on nodes of a tree graph.

3.1.2 Cartesian Genetic Programming

Cartesian Genetic Programming(CGP) [31] was first added to the literature by Miller in 1999. It was adopted as an innovative genetic programming type in the early 2000s [29]. As shown in Figure 3, programs in CGP are directly represented by grid graphics. This graphics consist of a two-dimensional grid of computation nodes. The term *"cartesian"* also takes its name from this grid arrangement of entities. Genes that make up the genotype (individual representations) in CGP are integers representing where a node receives its data at inputs, the actions on the data are performed at the node. The output data appears at the output of nodes for processing of the following nodes [32-33].

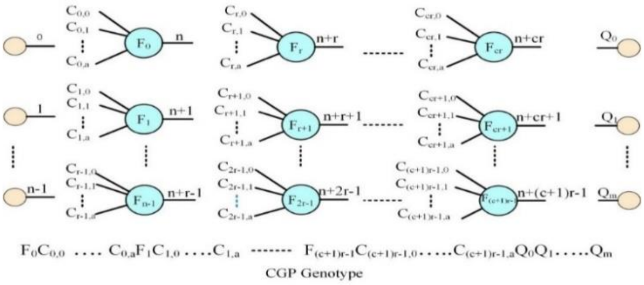


Figure 3 CGP graphic representation form [32]

Unlike the tree representations, CGP can represent individuals with a graphical drawing, the solutions can have multiple outputs, and the nodes can be used repeatedly within the repeating structures. In particular, it is a popular and easily adaptable representation method for the solution of many problems. Also CGP is an appropriate method of GP to represent many computational process such as equations, state machines, neural networks, algorithms, and electronic circuits [33].

As it can be seen in Figure 3, the CGP allows the internal calculations to reuse of data, as the outputs of the nodes in the graph can be reused multiple times. This property has attracted the attention of researchers working in the field of evolutionary computing and genetic programming since the CGP first appeared.

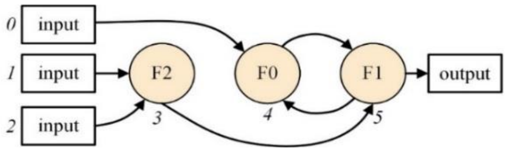
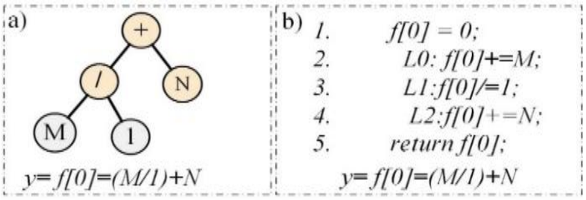


Figure 4 Recurrent CGP[34] chromosome structure

In order to overcome the difficulties encountered in engineering problems, CGP types that have many different features have been proposed in the literature. Figure 4 shows the Recurrent CGP [35], which is a type of CGP that allows the creation of repetitive/circular charts. Since the focus of this review article is not a comprehensive elaboration of all CGP variants, for further details on CGP variants, readers can consider some articles that focus on the CGP methods in literature [32], [36].

3.1.3 Linear Genetic Programming

Linear Genetic programming (LGP) is another GP variant, which differs in the program representation format. The individuals(programs) of LGP are represented by a linear code structure, as shown in Figure 5(b). The main difference between LGP and the tree-based GP is that instead of using a tree graph representation of the programs, it consists of a set of instructions, which are analogous to the machine codes in the register [30], [37].



```
f[0] = 0;
   L0: f[0]+=M;
   L1:f[0]/=1;
    L2:f[0]+=N;
return f[0];
```

Figure 5 a) Standard GP, b) Linear GP [37-38]

In Figure 5, f[0] represents temporary program variables while *L*'s represent command lines. Here, LGP has a structure that allows the variable to take different values in each command line. Each command structure contains an operator, an array of functions, and a return value [39].Since the LGP can be represented very similar to the programming languages, it can be more effective to synthesize computer programs than the standard GP [40-41].

3.1.4 Stack-based Genetic Programming

Stack-based Genetic Programming (SBGP) is another variant of GP, which was proposed by Perkis [42]. The representation format of SBGP consists of programs as lists of nodes of functions or terminals that receive their inputs from a stack and place their outputs on a stack [43]. It is a less preferred representation format compared to other representation formats.

3.2 GP Types According to Their Methodological Differences

Since the standard GP encounters technical complications (e.g. bloating, growing complexity) in solution of difficult problems, in addition to the progresses based on new representation formats in the literature, many different GP variants have also been proposed to improve performance of GPs by performing methodological enhancements on GP methods.

3.2.1 Strongly-typed GP

Strongly-typed Genetic Programming (STGP) [44], is an improved type of the standard GP by using data type restrictions. STPG is similar to standard GP, but the programs have a structure in which each node has different data types.

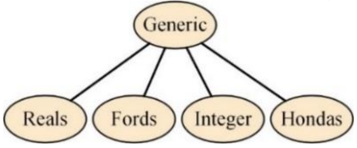


Figure 6 Strongly-typed GP [44]

As shown in Figure 6, the data types of functions and terminals need to be specified in STGP and therefore the population is only constructed with syntactically correct decomposed trees, which significantly reduces the search area[45]. Without data type limitation in the program nodes of the standard GP, the GP may cause processing numerous combinations of trees. In contrast, in STGP, the characteristics of each node are predetermined.

3.2.2 Multi-Gene GP

Multi-Gene Genetic Programming (MGGP) [46] can be accounted as a multi-gene form of the standard GP [1]. The classic symbolic regression problems employ the standard GP to evolve a population of programs that are represented in the form of single tree. Therefore, each of the trees represents a mathematical formulation that is a candidate solution of the regression problem. In contrast, MGGP benefits from a mathematical formulation that is a weighted linear combination of the outputs from a number of GP trees [47]. MGGP was suggested in the study of solving a complicated symbolic regression problem [46]. MGGP can be accounted as a powerful GP variant that effectively accumulates the abilities of standard GP solutions to empower prediction skills of regression model [47]. Therefore, it is very suitable to solve regression and modeling problems.

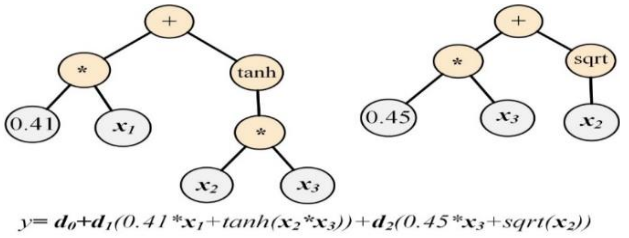

+
*
tanh
0.41
x_1
*
x_2
x_3
+
*
sqrt
0.45
x_3
x_2
$y= d_0+d_1(0.41*x_1+tanh(x_2*x_3))+d_2(0.45*x_3+sqrt(x_2))$

Figure 7 Multi-gene symbolic model [48]

When compared to the standard GP, the depth of the tree representation of genes can remain relatively shallow in MGGP, and it is can contribute to the relief of the bloating problem. The typical multiple gene models are shown in Figure 7. This model results in a mathematical formulation of an output variable based on weighted sum of two tree expressions of three input variables (x_1, x_2, and x_3). Such a formulation of MGGP have been shown to be more efficient than the standard GP in nonlinear problems [53-54].

3.2.3 Multi-objective GP

Standard GP is usually optimized for a single objective, and it has a single fitness function [50], [51]. Standard GP is insufficient in multi-objective optimization tasks. In a multi-objective optimization (MOGP) problem, problem-solution is optimized with respect to the multiple goals or objective functions [2].

A disadvantage of using a single objective in the optimization process of the GP is that the evolutional solution models can become extremely complex. Therefore, the two main objectives of MOGP are devised for minimizing complexity while maximizing fitness value [51].

3.2.4 Gene Expression Programming

In Gene Expression Programming (GEP), the genome or chromosome consists of one or more expression trees (ET) of linear, symbolic, fixed lengths consisting of one or more genes [52]. GEP is one of the most frequently used GP types in many different fields. GEP genes consist of a *head* and *tail*. Despite the fact that the head contains symbols representing functions (elements from the *F* function set) and terminals (elements from the *T* terminal set), the tail contains only the terminals. If we express, for a problem, *t* is the length of the tail, *h* is the length of the head, *n* is the number of arguments, the relationship between these variables is expressed by the following equation(1).

$$t = h\,(n\text{-}1)\ +1 \quad [52](1)$$

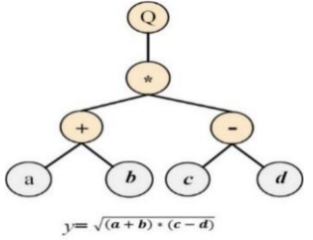


Figure 8 A sample GEP tree [52]

The program representation shown in Figure 8 is actually the phenotype of GEP individuals, the genotype can easily produced from the phenotype as follows:

01234567
Q*+-abcd

GEP selects population individuals by using one or more genetic operators according to their fitness, and it provides a variety of genetic variations [52]. In GEP, the individuals consist of strings of fixed length structure which are expressed as nonlinear entities of different sizes and shapes such as expression trees [52]. GEP allows solution of high complexity problems on personal computers [52] because GEP has rather simple genetic operations that minimize the need for powerful hardware for evolutionary computing [53][54].

3.2.5 Grammar Guided GP

Grammar Guided Genetic Programming (GGGP or G3P) [55-56] is a variant of the original GP. Grammar-guided GP is also known as grammar-based GP in the literature. Grammar provides many benefits to GP. Undoubtedly the most important benefit is that it can be used as a restriction tool on a flexible search space [57].

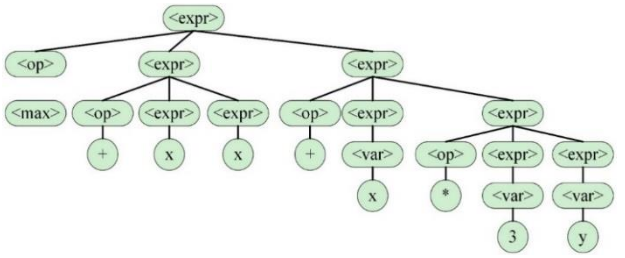


Figure 9 Grammar-based tree[58]

Grammar-based formulations are the basic representation structures of computer science[57]. They are widely used to express constraints in general areas by limiting the expressions that can be used. The individuals of GGGP can use both linear and tree representation methods. In Figure 9, an example of grammar-based tree representation is illustrated.

3.2.6 Grammar Evolution (GE)

As a sub-type of GGGP, the GE [59] is a GP method that performs evolutionary processes in variable-length binary strings instead of real programs. GE perform a mapping technic for create programs in any language using binary strings to creating rules in a Backus-Naur Format (BNF) grammar. This approach is to obtain a syntactically suitable expression from a binary string, which can then be evaluated with a fitness model [59].

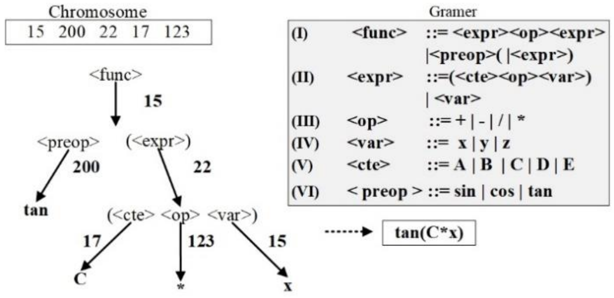

Chromosome
15 200 22 17 123
<func>
15
<preop>
(<expr>)
200
22
tan
(<cte> <op> <var>)
17
123
15
C
*
x
Gramer
(I) <func> ::= <expr><op><expr> |<preop>(|<expr>)
(II) <expr> ::=(<cte><op><var>) | <var>
(III) <op> ::= + | - | / | *
(IV) <var> ::= x | y | z
(V) <cte> ::= A | B | C | D | E
(VI) < preop > ::= sin | cos | tan
tan(C*x)

Figure 10 Performing a sample decoding in GE [60]

The purpose and main cycle are basically the same as standard GP, but they differ in the way that solutions are created and updated [61]. Figure 10 shows an example of decoding with GE. If you want to solve a problem with GE, the first thing to do is to define an appropriate grammar, which is usually done using the Backus-Naur form (BNF). Thus, it is an important step as it defines the search space for the solution of the problem and calculation expressions (individuals) will be place in this search space [62].

3.2.7 Geometric Semantic Genetic Programming

Geometric Semantic Genetic Programming (GSGP) [63] is one of the recently proposed GP types. GSGP establishes a semantic field that uses semantics of possible solutions and benefits from different distance metrics in order to measure the suitability of GP individuals for the solution. There are several methods for defining the semantics of the population program. Depending on the properties of the distance metrics, the semantic field can have different conical forms. The GSGP searches for the solution in the semantic space of the programs. In this way, it facilitates searching in the solution space [64].

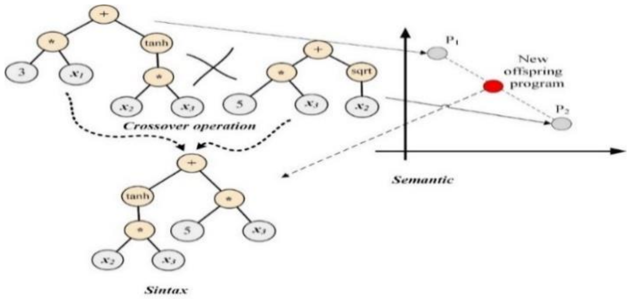


Figure 11 Geometric semantic crossover[65]

Geometric semantic operators change the programs to generate new programs in a semantic space [66]. Figure 11 shows an example of a geometric semantic crossover operation in the semantic space. Before the crossover operation, each individual has its own position in the semantic space. After the crossover operation, a semantic point (value) corresponding to the newly formed offspring program is assigned. Similarly, in the mutation process, a relevant semantic operation of the program individuals is performed, and the geometric semantic values are assigned according to the suitability of new programs [65].

3.3 Some Popular GP Types

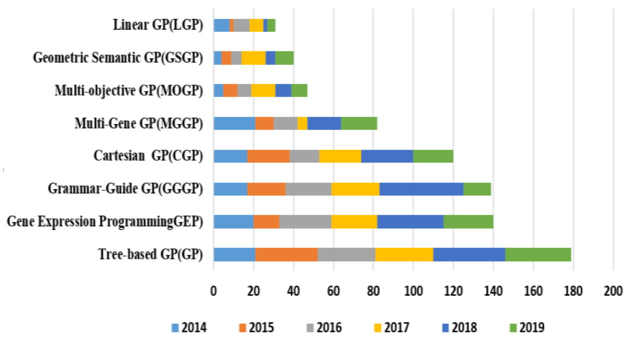


Figure 12 GP variants between 2014 and 2019

It can be useful to illustrate yearly publication distributions of the popular GP types to reveal emerging trends among GP types compares yearly publication rates in the years between 2014 and 2019. Although the tree-based GP is trendy even though it is shortcomings mentioned in Section 2, one can observe that the GEP, GGGP, and Cartesian GP variants also have a frequent use in the publications. The main reason for the widespread use of these GP types is their easy applicability and delivery of many software tools and documents related to these GP types.

4. APPLICATION FIELDS OF GENETIC PROGRAMMING

This section surveys the recent application fields of the GP methods and considerable application articles in these fields. The analysis of application trends is especially useful for new researchers who need to explore application domains of GP algorithms. In order to keep the reviewed articles at a reasonable extend, instead of explaining all GP applications, we summarized the study collections that were intensified into the specific application fields.

Symbolic regression: Symbolic regression is a topic at the intersection of applied mathematics and computer science, which investigate approaches to produce the best symbolic mathematical expression that describes the model of the existing relationships between a well-known set of independent variables and the associated values of dependent variables [67]. Mechanisms of GP methods well suit for characteristics of symbolic regression problems, and it has been used intensely in this area [64], [68]–[70].

Artificial Neural Network (ANN) design: A corporation of Artificial Neural Network (ANN) and Evolutionary Algorithm (EA) is a branch of machine learning, which was named as NeuroEvolution (NE) [71], [72]. There are many studies [21], [72]–[74] that use CGP in the optimization of ANN's topology and networks. In this fashion, hybrid approaches [75] based on collaboration of GP and ANNs have been proposed. Meanwhile, there is a CGP type, namely Recurrent CGP (RCGP) [34], particularly developed for ANNs.

Computer vision: GP algorithms have been utilized in computer vision applications e.g. recognition of human motion [18], improving the performance of the histogram of oriented gradients (HOG) algorithm for human detection problems [76], in robotics[77], human movement modeling [78], improving edge detection in images [79] and in the pattern recognition problems [80] etc.

Circuit Design and Evolvable hardware: Evolutionary Hardware (EHW) is a design approach that uses a reconfigurable hardware structure to develop a circuit that performs a specific function. Hardware can be designed automatically by using GP algorithms without the need for a circuit designer [22]. Due to convenience of program representations to express hardware, the CGP method is employed extensively in circuit design works. As a consequence, the GP has been widely utilized in EHW studies [22], [81], [82] Besides, the GP is frequently used in digital circuit design tasks [23], [83]–[85].

Scheduling: The Scheduling is a process that deals with the allocation of limited resources by serving for the given times. It has been utilized in many production and service industries [86]. The GP has been frequently used in many timing problems such as dynamic job shop scheduling(JSS) [8], [87] production scheduling [88], action scheduling [89] scheduling in heterogeneous network [90], [91]

Environmental, natural disasters and agriculture: GP methods have used especially for data modeling and forecasting in many areas such as carbon emission [92], monitoring of volcanoes [93], earthquake prediction [94], atmosphere studies [95], airflow measurement [96], modeling rainwater quality [97], analysis of agricultural yield response [98], reservoir operations and irrigation [9].

Classification: The relevance of the selected features is one of the important factors that can affect the classification performance. The appropriate feature selection increases distinguishability between classes. However, in some real-world classification applications, there may not be enough information about the available features [99]. GP has been used to explore effective features in classification problems [7], [94], [99] and associative classification [100].

Urbanization and Building: The energy management and infrastructure planning problems can require high complexity computational models. To produce feasible solutions for efficiency and sustainability concerns, effective computational intelligence methods are needed to overcome such model complexity. Therefore, GP algorithms have found applications these problems such as energy efficiency in buildings [11], [101], the ground and soil analyzes [15], [102], urban transportation and infrastructure planning [14], [103]

Financing, trade, and economy: The financial market introduces very complex, nonstationary and chaotic data models. To overcome this challenge, GP methods have been implemented and successful results have been reported. One can see GP applications in finance, commerce, and economy problems, for examples supply selection [104], investment management [105]–[107], market analysis [108], financial security [24], stock analysis and management [109-110] etc.

Image processing: Digital imaging technologies and image processing are used in various fields, e.g. medical applications, meteorology, geology, and biology etc. These images may contain noise and requires a preprocessing task [111]. Researchers from image processing field have used the GP methods in image processing studies such as in noise suppression [111-112], image reconstruction [113], feature extraction[114], image classification [115] etc.

Signal processing: GP algorithms has been utilized in the classification of EEG signals [116], which is a very important task in the diagnosis of several diseases and disorders such as epileptic seizures [117], sleep disorders [141]. One may also see GP employments in processing of other medical signals such as classification of electrocardiography (ECG) signals [20], [118], which are the medical signals that are used to diagnose heart problems. The another signal processing application of GP is related to audio signal processing for instance audio signal reconstruction application [119].

Education: With beginning of Industry 4.0, artificial intelligence techniques will be used more frequently in order to reduce the workload of teachers and to provide student-friendly solutions in education systems. Nowadays, online intelligent learning systems are capable of automated assessment of learning activities, and therefore computational intelligence begins to play an important role in education [120]. In this fashion, GP methods have been performed in student performance prediction [13], [121-122].

Hydrology: Hydrology is a branch of water science that widely needs predictions models. GP was widely used in hydrology applications such as precipitation prediction and measurement [123], Rainfall-Runoff modeling [124-125] groundwater quality prediction [126], evapotranspiration estimation [127] etc. A comprehensive review study focuses on GP applications in the field of hydrology [128].

Medicine, Biology and Bioinformatics: GP methods have found a wide application in the medical, biology and bioinformatics fields, particularly for diagnosis, classification, prediction and modeling purposes. For examples, analysis and modeling of blood chemicals [17], [129] such as glucose-dynamics models that are vital for diabetes diseases, data mining in medicine[37] for analyses in asthma and allergy epidemiology [58], predictions of pharmacokinetic parameters [130], and diabetes mellitus [131], and automatic diagnosis of Parkinson disease [132] are some of the applications.

Time Series Prediction: Time series used in many fields such as statistics and econometrics. GP methods are used in time series prediction studies [133-137]

Energy: Nowadays, due to a growing demand for energy all over the globe, it is great importance to improve energy efficiency and reliability by providing more efficient use of energy resources, optimal energy generation and demand balance, consumption prediction, and reduction of energy lost etc. GP methods has been used in energy field such as energy consumption prediction analysis [138-139], energy demand estimation [60], biomass energy analysis [140], risk analysis in nuclear energy systems [141], flexibility analysis in waste-to-energy systems [142] etc.

5. SOFTWARE TOOLS FOR GENETIC PROGRAMMING

This section introduces available software tools that were developed for GP applications.

DEAP: DEAP [143] is an advanced Python-based, open-source evolutionary computing software tool, where target problems can be quickly adapted and tested.

GPLEARN: Gplearn [144] is an open-source Python-based GP library. It uses a tree-based representation format. The Gplearn tool is quite suitable for symbolic regression problems.

GELAB: GELAB is an open-source Matlab[145] library [146] The application of the GGGP types to problems can be conducted by using GELAB in the Matlab environment.

CGP4Matlab: CGP4Matlab [147] is another open-source software tool that enables developing CGP applications in Matlab environment. CGP4Matlab was developed especially for signal processing and image processing problems [148].

GPTIPS 2: GPTIPS [149] is also an open-source Matlab GP software development tool. Many articles [150-152] used this software tool, and this indicates the interest of researchers to implement this tool in their works. Also, the GPTIPS tool was used for symbolic regression problems [153].

PonyGE2: PonyGE2 [154-155] is a open-source software tool that offers a development environment in Python language. This tool includes grammar-based types of GP.

GeneXproTools: GeneXproTools [156] is a desktop commercial software that enables analysis with various data model. This professional software was developed for specialists with knowledge of statistics, mathematics, machine learning or programming. It includes GEP type and it is used in emotion classification study [157] and in soil temperature analysis [158].

GEP4J: GEP4J [159] is an open source Java library. It allows developing GEP applications in Java environment.

PyGEP: PyGEP [160] is an open-source Python library. It allows GEP applications in Python.

JGEP: JGEP [161] is another open-source Java library where GEP applications can be developed.

EpochX: EpochX [162] is an open-source GP software development package. It has a Java framework that is specially developed to analyze evolutionary automated programming. EpochX is licensed under the GNU LGPL version 3 license.

Karoo GP: Karoo GP [163] is a GP open-source software package written in Python. With this tool, both symbolic regression and classification studies can be carried out easily. Karoo GP is a scalable platform with multicore and the TensorFlow support [164], which may facilitate working with real-world data. Karo GP is licensed under the MIT license.

GISMO: GISMO [165] is a software package for open source, multipurpose GP studies with parallel computing capability [166-167].

KNIME: KNIME [168-169] is a cross-platform for data analysis, reporting and integration platforms. In addition, data processing and visualization operations in the form of new modules or nodes can be performed by KNIME. KNIME is available in commercial and free versions. Researchers can access free versions and contribute to improvements [170].

GSGP-C++:GSGP-C++ 2.0 [171] is an open source C ++ GP development library. It supports GSGP [172-173].

GPLAB:GPLAB [174] a software package for developing GP applications in Matlab environment. It has been claimed by the study that the GP bloating problem can be controlled in GPLAB [175].

6. SOME RECENT REVIEW WORKS ON GENETIC PROGRAMING

Several review type works related with applications of the genetic programming have been presented. A framework of product scheduling and the related works have been explained in [8]. The role of genetic programming in the empirical modeling has been surveyed[176]. An interpretation of the term of ”emergence” according to genetic programming was discussed and related works was mentioned in [177]. A review of the semantic methods in the genetic programming has been presented in [178]. Survey works for use of GP in the specific fields were also provided; for instance, in the water resource engineering [128], the energy efficiency [179], and civil engineering [180]. Some survey works addressed specific GP approaches such as grammar-based Genetic programming [26], gene expression programming [181]. However, a general review of GP domain, which presents an outlook for the growing GP literature in a systematic way, is limited [28]. This review study aims to provide a general view for the GP field from beginning to the recent developments for new researchers who decide to implement GP methods in their applications.

7. CONCLUSIONS

This review article reveals the fact that GP topic has turned into a fast growing interdisciplinary field, where the powerful and flexible computation potential of GP algorithms has been utilized for the solution of wide-range of real world problems in many disciplines. Therefore GP literature has been widely expanded and still expanding with new types of GPs so that the standard GP can not sufficiently respond such diversity in problem types and their difficulty levels. As problems are getting harder and the need for progressing GP algorithms keeps on.

The current paper also highlights yearly publication trends of GP variants from 2014 and 2019 and the application fields that GP utilization has been intensified. These efforts give a clue for recent trends in GP algorithms and their spreading fields of application. In addition, a detailed list of GP software development tools has been mentioned for researchers who step into GP world. Many developers of the GP software tool aim to provide user-friendly tools that can be used not only by those who are experts in software but also those who do not have programming knowledge. In addition, thanks to the increasing computation power in the today’s computers and high-performance processors, GP algorithms can run faster than before. This progress allows implementation of GP algorithm in more complicated and complex real world problems as an effective computational intelligence tools.

This literature survey also aims to be a good starting point for new researchers who want to start working on GP algorithms. Thus, the paper provides an overview to the field of GP and its development stages.

One can conclude that, according to the increase in the number of recent works of GP taken into consideration, GP will continue its progress as a strong branch of computational intelligence by increasing its popularity in the future applications. It is very expectable that by adding additions to the basic features of GP, emergence of fresh GP types with superior computing abilities, which can give better results in much more complex and complicated problems, will be a continuing trend.

Funding

The authors received no financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

Authors' Contribution

All authors have contributed in experimental study and writing of the manuscript equally.

The Declaration of Ethics Committee Approval

The authors declare that this document does not require an ethics committee approval or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the article and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] J. R. Koza, ‘Genetic programming: on the programming of computers by means of natural selection’, MIT press.,1992.

[2] R. Poli, W. B. Langdon, and N. F. McPhee, ‘A Field Guide to Genetic Programing’, no. March,Lulu Enterprises, UK Ltd, 2008.

[3] M. Amir Haeri, M. M. Ebadzadeh, and G. Folino, ‘Statistical genetic programming for symbolic regression’, Appl. Soft Comput., vol. 60, ,pp. 447–469, 2017.

[4] M. O’Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf, ‘Open issues in Genetic Programming’, Genet. Program. Evolvable Mach., vol. 11, no. 3–4, ,pp. 339–363, 2010.

[5] A. Cano and S. Ventura, ‘GPU-parallel subtree interpreter for genetic programming’, Proceedings of the 2014 conference on Genetic and evolutionary computation - GECCO ’14, no. July, New York, New York, USA,ACM Press,pp. 887–894, 2014.

[6] L. F. Dal Piccol Sotto and V. V. De Melo, ‘Investigation of linear genetic programming techniques for symbolic regression’, Proc. - 2014 Brazilian Conf. Intell. Syst. BRACIS 2014, ,pp. 146–151, 2014.

[7] B. Tran, B. Xue, and M. Zhang, ‘Genetic programming for feature construction and selection in classification on high-dimensional data’, Memetic Comput., vol. 8, no. 1, ,pp. 3–15, 2016.

[8] S. Nguyen, Y. Mei, and M. Zhang, ‘Genetic programming for production scheduling: a survey with a unified framework’, Complex Intell. Syst., vol. 3, no. 1, ,pp. 41–66, 2017.

[9] P.-S. Ashofteh, O. B. Haddad, and H. A. Loáiciga, ‘Evaluation of Climatic-Change Impacts on Multiobjective Reservoir Operation with Multiobjective Genetic Programming’, J. Water Resour. Plan. Manag., vol. 141, no. 11, ,p. 04015030, 2015.

[10] M. Castelli, L. Trujillo, L. Vanneschi, and A. Popovič, ‘Prediction of energy performance of residential buildings: A genetic programming approach’, Energy Build., vol. 102, ,pp. 67–74, 2015.

[11] A. Tahmassebi and A. H. Gandomi, ‘Building energy consumption forecast using multi-objective genetic programming’, Measurement, vol. 118, no. January, ,pp. 164–171, 2018.

[12] S. H. A. Kaboli, A. Fallahpour, J. Selvaraj, and N. A. Rahim, ‘Long-term electrical energy consumption formulating and forecasting via optimized gene expression programming’, Energy, vol. 126, ,pp. 144–164, 2017.

[13] A. Zafra and S. Ventura, ‘Multi-instance genetic programming for predicting student performance in web based educational environments’, Appl. Soft Comput., vol. 12, no. 8, ,pp. 2693–2706, 2012.

[14] A. K. Patnaik and P. K. Bhuyan, ‘Application of genetic programming clustering in defining LOS criteria of urban street in Indian context’, Travel Behav. Soc., vol. 3, ,pp. 38–50, 2016.

[15] A. Keshavarz and M. Mehramiri, ‘New Gene Expression Programming models for normalized shear modulus and damping ratio of sands’, Eng. Appl. Artif. Intell., vol. 45, ,pp. 464–472, 2015.

[16] M. Shoaib, A. Y. Shamseldin, B. W. Melville, and M. M. Khan, ‘Runoff forecasting using hybrid Wavelet Gene Expression Programming (WGEP) approach’, J. Hydrol., vol. 527, ,pp. 326–344, 2015.

[17] I. De Falco, A. Della Cioppa, T. Koutny, M. Krcma, U. Scafuri, and E. Tarantino, ‘Genetic Programming-based induction of a glucose-dynamics model for telemedicine’, J. Netw. Comput. Appl., vol. 119, no. December 2017, ,pp. 1–13, 2018.

[18] L. Liu, L. Shao, X. Li, and K. Lu, ‘Learning Spatio-Temporal Representations for Action Recognition: A Genetic Programming Approach’, IEEE Trans. Cybern., vol. 46, no. 1, ,pp. 158–170, 2016.

[19] L. Shao, L. Liu, and X. Li, ‘Feature learning for image classification via multiobjective genetic programming’, IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 7, ,pp. 1359–1371, 2014.

[20] M. Feli and F. Abdali-Mohammadi, ‘A novel recursive backtracking genetic programming-based algorithm for 12-lead ECG compression’, Signal, Image Video Process., vol. 13, no. 5, ,pp. 1029–1036, 2019.

[21] M. Suganuma, S. Shirakawa, and T. Nagao, ‘A genetic programming approach to designing convolutional neural network architectures’,

Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO ’17, New York, New York, USA,ACM Press,pp. 497–504, 2017.

[22] J. Mora, R. Salvador, and E. de la Torre, ‘On the scalability of evolvable hardware architectures: comparison of systolic array and Cartesian genetic programming’, vol. 20, no. 2,Springer US, 2019.

[23] M. Sikulova, G. Komjathy, and L. Sekanina, ‘Towards compositional coevolution in evolutionary circuit design’, IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - IEEE ICES 2014 IEEE Int. Conf. Evolvable Syst. Proc., ,pp. 157–164, 2014.

[24] H. Li and M. L. Wong, ‘Financial Fraud Detection by using Grammar-based Multi-objective Genetic Programming with ensemble learning’, 2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc., ,pp. 1113–1120, 2015.

[25] A. Fallahpour, E. U. Olugu, S. N. Musa, D. Khezrimotlagh, and K. Y. Wong, ‘An integrated model for green supplier selection under fuzzy environment: application of data envelopment analysis and genetic programming approach’, Neural Comput. Appl., vol. 27, no. 3, ,pp. 707–725, 2016.

[26] M. W. Khan and M. Alam, ‘A survey of application: Genomics and genetic programming, a new frontier’, Genomics, vol. 100, no. 2, ,pp. 65–71, 2012.

[27] R. Poli, N. F. McPhee, and L. Vanneschi, ‘Elitism reduces bloat in genetic programming’, GECCO’08 Proc. 10th Annu. Conf. Genet. Evol. Comput. 2008, ,pp. 1343–1344, 2008.

[28] R. Poli, W. B. Langdon, N. F. McPhee, and J. R. Koza, ‘Genetic programming an introductory tutorial and a survey of techniques and applications’, Tech Rep CES475, vol. 18, no. October 2007, ,pp. 1–112, 2007.

[29] P. T. J.F. Miller, ‘Cartesian genetic programming’, Proceedings of the European Conference on Genetic Programming, ,pp. 121–132, 2000.

[30] M. Brameier and W. Banzhaf, ‘Evolving Teams of Predictors with Linear Genetic Programming’, Genet. Program. Evolvable Mach., vol. 2, no. 4, ,pp. 381–407, 2001.

[31] J. F. Miller, ‘An empirical study of the efficiency of learning boolean functions using a Cartesian Genetic Programming approach’, Proc. Genet. Evol. Comput. Conf., vol. 2, ,pp. 1135--1142, 1999.

[32] J. F. Miller, ‘Cartesian Genetic Programming’, Berlin, Heidelberg,Springer Berlin Heidelberg,pp. 17–34, 2011.

[33] J. F. Miller, ‘Cartesian genetic programming: its status and future’, no. April,Springer US, 2019.

[34] A. J. Turner and J. F. Miller, ‘Recurrent Cartesian Genetic Programming of Artificial Neural Networks’, Genet. Program. Evolvable Mach., vol. 18, no. 2, ,pp. 185–212, 2017.

[35] A. J. Turner and J. F. Miller, ‘Recurrent Cartesian Genetic Programming’, ,pp. 476–486, 2014.

[36] A. Manazir and K. Raza, ‘Recent developments in cartesian genetic programming and its variants’, ACM Comput. Surv., vol. 51, no. 6, 2019.

[37] M. Brameier and W. Banzhaf, ‘A comparison of linear genetic programming and neural networks in medical data mining’, IEEE Trans. Evol. Comput., vol. 5, no. 1, ,pp. 17–26, 2001.

[38] H. Shavandi and S. Saeedi Ramyani, ‘A linear genetic programming approach for the prediction of solar global radiation’, Neural Comput. Appl., vol. 23, no. 3–4, ,pp. 1197–1204, 2013.

[39] T. Hu, J. L. Payne, W. Banzhaf, and J. H. Moore, Genetic Programming and Evolvable Machines, vol. 13, no. 3. ,pp. 305–337, ,pp. 305–337, 2012.

[40] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin., ‘Genetic Programming: An Introduction.’,San Francisco, CA,Morgan Kaufmann Publishers, 1998.

[41] A. Guven and O. Kisi, ‘Monthly pan evaporation modeling using linear genetic programming’, J. Hydrol., vol. 503, ,pp. 178–185, 2013.

[42] T. Perkis, ‘Stack-based genetic programming’, Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, IEEE,pp. 148–153, 1994.

[43] D. S. P. Elko B. Tchernev, ‘Control structures in linear and stack-based Genetic

Programming', Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference, Seattle, Washington, USA., 2004.

[44] D. J. Montana, 'Strongly Typed Genetic Programming', Evol. Comput., vol. 3, no. 2, ,pp. 199–230, 1995.

[45] T. D. Haynes, D. A. Schoenefeld, and R. L. Wainwright, 'Type Inheritance in Strongly Typed Genetic Programming', 1996.

[46] G. W. Hinchliffe, M. P., Willis, M. J., Hiden, H., Tham, M. T., McKay, B., & Barton, 'Modelling chemical process systems using a multi-gene genetic programming algorithm', In Genetic Programming: Proceedings of the First Annual Conference, ,pp. 56–65, 1996.

[47] A. H. Gandomi and A. H. Alavi, 'A new multi-gene genetic programming approach to nonlinear system modeling. Part I: Materials and structural engineering problems', Neural Comput. Appl., vol. 21, no. 1, ,pp. 171–187, 2012.

[48] W. M. Searson DP, Leahy DE, 'GPTIPS: An Open Source Genetic Programming Toolbox For Multigene Symbolic Regression', International MultiConference of Engineers and Computer Scientists 2010 (IMECS), ,pp. 77–80, 2010.

[49] D. Searson, M. Willis, and G. Montague, 'Co-evolution of non-linear PLS model components', J. Chemom., vol. 21, no. 12, ,pp. 592–603, 2007.

[50] R. Dai, Y. Gao, S. Huang, F. Ning, and Z. Feng, 'Multi-objective Genetic Programming based Automatic Modulation Classification', 2019 IEEE Wireless Communications and Networking Conference (WCNC), vol. 2019-April, IEEE,pp. 1–6, 2019.

[51] A. H. Gandomi, S. Sajedi, B. Kiani, and Q. Huang, 'Genetic programming for experimental big data mining: A case study on concrete creep formulation', Autom. Constr., vol. 70, ,pp. 89–97, 2016.

[52] C. Ferreira, 'Gene Expression Programming: a New Adaptive Algorithm for Solving Problems', ,pp. 1–22, 2001.

[53] A. Ş. Şahin, E. Dikmen, and S. Şentürk, 'A gene expression programming approach for thermodynamic properties of working fluids used on Organic Rankine Cycle', Neural Comput. Appl., vol. 31, no. 8, ,pp. 3947–3955, 2019.

[54] C. Ferreira, 'Gene Expression Programming', vol. 21,Springer Berlin Heidelberg, 2006.

[55] Man Leung Wong and Kwong Sak Leung, 'Inducing logic programs with genetic algorithms: the Genetic Logic Programming System', IEEE Expert, vol. 10, no. 5, ,pp. 68–76, 1995.

[56] M. L. Wong and K. S. Leung, 'Evolutionary Program Induction Directed by Logic Grammars', Evol. Comput., vol. 5, no. 2, ,pp. 143–180, 1997.

[57] R. I. McKay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O'neill, 'Grammar-based Genetic programming: A survey', Genet. Program. Evolvable Mach., vol. 11, no. 3–4, ,pp. 365–396, 2010.

[58] R. V. Veiga, H. J. C. Barbosa, H. S. Bernardino, J. M. Freitas, C. A. Feitosa, S. M. A. Matos, N. M. Alcântara-Neves, and M. L. Barreto, 'Multiobjective grammar-based genetic programming applied to the study of asthma and allergy epidemiology', BMC Bioinformatics, vol. 19, no. 1, ,pp. 1–16, 2018.

[59] M. O'Neill and C. Ryan, 'Grammatical evolution', IEEE Trans. Evol. Comput., vol. 5, no. 4, ,pp. 349–358, 2001.

[60] J. M. Colmenar, J. I. Hidalgo, and S. Salcedo-Sanz, 'Automatic generation of models for energy demand estimation using Grammatical Evolution', Energy, vol. 164, ,pp. 183–193, 2018.

[61] C. Martín, D. Quintana, and P. Isasi, 'Evolution of trading strategies with flexible structures: A configuration comparison', Neurocomputing, vol. 331, ,pp. 242–262, 2019.

[62] I. Fajfar, Á. Bűrmen, and J. Puhan, 'Grammatical evolution as a hyper-heuristic to evolve deterministic real-valued optimization algorithms', vol. 19, no. 4,Springer US, 2018.

[63] A. Moraglio, K. Krawiec, and C. G. Johnson, 'Geometric Semantic Genetic Programming', Berlin,Springer, Berlin, Heidelberg,pp. 21–31, 2012.

[64] Q. Chen, B. Xue, and M. Zhang, 'Improving Generalization of Genetic Programming for Symbolic Regression with Angle-Driven Geometric Semantic Operators', IEEE Trans. Evol. Comput., vol. 23, no. 3, ,pp. 488–502,

2019.

[65] L. Vanneschi, M. Castelli, I. Goncalves, L. Manzoni, and S. Silva, 'Geometric semantic genetic programming for biomedical applications: A state of the art upgrade', 2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc., ,pp. 177–184, 2017.

[66] J. P. Papa, G. H. Rosa, and L. P. Papa, 'A binary-constrained Geometric Semantic Genetic Programming for feature selection purposes', Pattern Recognit. Lett., vol. 100, ,pp. 59–66, 2017.

[67] F. Casadei, J. F. B. S. Martins, and G. L. Pappa, 'A multi-objective approach for symbolic regression with semantic genetic programming', Proc. - 2019 Brazilian Conf. Intell. Syst. BRACIS 2019, ,pp. 66–71, 2019.

[68] Z. Qu, Y. X. Chen, L. Liu, Y. Xie, and Q. Zhou, 'The Algorithm of Concrete Surface Crack Detection Based on the Genetic Programming and Percolation Model', IEEE Access, vol. 7, ,pp. 57592–57603, 2019.

[69] Y. Wang, N. Wagner, and J. M. Rondinelli, 'Symbolic regression in materials science', MRS Commun., vol. 9, no. 3, ,pp. 793–805, 2019.

[70] P. DIng, Q. Qian, H. Wang, and J. Yao, 'A Symbolic Regression Based Residual Useful Life Model for Slewing Bearings', IEEE Access, vol. 7, ,pp. 72076–72089, 2019.

[71] G. M. Khan, R. Arshad, and N. M. Khan, 'Efficient Prediction of Dynamic Tariff in Smart Grid Using CGP Evolved Artificial Neural Networks', 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), vol. 2017-Decem, IEEE,pp. 493–498, 2017.

[72] J. M. Melo Neto, H. S. Bernardino, and H. J. C. Barbosa, 'Hybridization of Cartesian Genetic Programming and Differential Evolution for Generating Classifiers Based on Neural Networks', 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE,pp. 1–8, 2018.

[73] R. Arshad, G. M. Khan, and S. A. Mahmud, 'Smart bandwidth management using a recurrent Neuro-Evolutionary technique', Proc. Int. Jt. Conf. Neural Networks, ,pp. 2240–2247, 2014.

[74] M. M. Khan, G. M. Khan, and J. F. Miller, 'Evolution of neural networks using Cartesian Genetic Programming', IEEE Congress on Evolutionary Computation, IEEE,pp. 1–8, 2010.

[75] M. Mazari and D. D. Rodriguez, 'Prediction of pavement roughness using a hybrid gene expression programming-neural network technique', J. Traffic Transp. Eng. (English Ed., vol. 3, no. 5, ,pp. 448–455, 2016.

[76] M. Wiglasz and L. Sekanina, 'Evolutionary approximation of gradient orientation module in HOG-based human detection system', 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), vol. 2018-Janua, IEEE,pp. 1300–1304, 2017.

[77] J. Leitner, A. Forster, and J. Schmidhuber, 'Improving robot vision models for object detection through interaction', 2014 International Joint Conference on Neural Networks (IJCNN), IEEE,pp. 3355–3362, 2014.

[78] W. He, S. Hu, S. Li, J. Jin, and K. Li, 'Human Motion Model Construction Based on Gene Expression Programming', Springer Singapore,pp. 473–485, 2018.

[79] W. Fu, M. Johnston, and M. Zhang, 'Low-level feature extraction for edge detection using genetic programming', IEEE Trans. Cybern., vol. 44, no. 8, ,pp. 1459–1472, 2014.

[80] H. Al-Sahaf, M. Zhang, A. Al-Sahaf, and M. Johnston, 'Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving rotation-invariant texture image descriptors', IEEE Trans. Evol. Comput., vol. 21, no. 6, ,pp. 825–844, 2017.

[81] J. Mora, A. Otero, E. De La Torre, and T. Riesgo, 'Fast and compact evolvable systolic arrays on dynamically reconfigurable FPGAs', 10th Int. Symp. Reconfigurable Commun. Syst. ReCoSoC 2015, ,pp. 1–7, 2015.

[82] Z. Vasicek, 'Bridging the Gap Between Evolvable Hardware and Industry Using Cartesian Genetic Programming', ,pp. 39–55, 2018.

[83] Z. Vasicek and L. Sekanina, 'Evolutionary design of approximate multipliers under different error metrics', 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, IEEE,pp. 135–140, 2014.

[84] V. Mrazek and Z. Vasicek, 'Parallel optimization of transistor level circuits using cartesian genetic programming', Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17, New York, New York, USA,ACM Press,pp. 1849–1856, 2017.

[85] F. Castejón and E. J. Carmona, 'Automatic design of analog electronic circuits using grammatical evolution', Appl. Soft Comput., vol. 62, ,pp. 1003–1018, 2018.

[86] M. Đumić, D. Šišejković, R. Čorić, and D. Jakobović, 'Evolving priority rules for resource constrained project scheduling problem with genetic programming', Futur. Gener. Comput. Syst., vol. 86, ,pp. 211–221, 2018.

[87] J. Park, Y. Mei, S. Nguyen, G. Chen, and M. Zhang, 'An investigation of ensemble combination schemes for genetic programming based hyper-heuristic approaches to dynamic job shop scheduling', Appl. Soft Comput., vol. 63, ,pp. 72–86, 2018.

[88] M. Đurasević, D. Jakobović, and K. Knežević, 'Adaptive scheduling on unrelated machines with genetic programming', Appl. Soft Comput., vol. 48, ,pp. 419–430, 2016.

[89] M. A. Abud Kappel, 'Action scheduling optimization using cartesian genetic programming', Proc. - 2019 Brazilian Conf. Intell. Syst. BRACIS 2019, ,pp. 293–298, 2019.

[90] T. Dou, Y. Kaszubowski Lopes, P. Rockett, E. A. Hathway, and E. Saber, 'GPML: an XML-based standard for the interchange of genetic programming trees', Genet. Program. Evolvable Mach., no. 0123456789, 2019.

[91] T. Saber, D. Fagan, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, 'A Hierarchical Approach to Grammar-Guided Genetic Programming: The Case of Scheduling in Heterogeneous Networks', ,pp. 225–237, 2018.

[92] A. Garg, L. Gao, W. Li, S. Singh, X. Peng, X. Cui, Z. Fan, H. Singh, and C. M. M. Chin, 'Evolutionary framework design in formulation of decision support models for production emissions and net profit of firm: Implications on environmental concerns of supply chains', J. Clean. Prod., vol. 231, ,pp. 1136–1148, 2019.

[93] G. Khattak, M. S. Khan, G. M. Khan, F. Huenupan, and M. Curilem, 'Automatic Classification of seismic signals of the Chilean Llaima Volcano using Cartesian Genetic Programming based Artificial Neural Network', ,pp. 36 (6 .)-36 (6 .), 2017.

[94] K. M. Asim, A. Idris, T. Iqbal, and F. Martínez-Álvarez, 'Seismic indicators based earthquake predictor system using Genetic Programming and AdaBoost classification', Soil Dyn. Earthq. Eng., vol. 111, no. February, ,pp. 1–7, 2018.

[95] T. Zerenner, V. Venema, P. Friederichs, and C. Simmer, 'Downscaling near-surface atmospheric fields with multi-objective Genetic Programming', Environ. Model. Softw., vol. 84, ,pp. 85–98, 2016.

[96] M. Khandelwal, D. J. Armaghani, R. S. Faradonbeh, P. G. Ranjith, and S. Ghoraba, 'A new model based on gene expression programming to estimate air flow in a single rock joint', Environ. Earth Sci., vol. 75, no. 9, ,pp. 1–13, 2016.

[97] K. Zhang, A. Deletic, P. M. Bach, B. Shi, J. M. Hathaway, and D. T. McCarthy, 'Testing of new stormwater pollution build-up algorithms informed by a genetic programming approach', J. Environ. Manage., vol. 241, no. January, ,pp. 12–21, 2019.

[98] A. S. Kima, S. Traore, Y.-M. Wang, and W.-G. Chung, 'Multi-genes programing and local scale regression for analyzing rice yield response to climate factors using observed and downscaled data in Sahel', Agric. Water Manag., vol. 146, ,pp. 149–162, 2014.

[99] J. Ma and G. Teng, 'A hybrid multiple feature construction approach for classification using Genetic Programming', Appl. Soft Comput. J., vol. 80, ,pp. 687–699, 2019.

[100] F. Padillo, J. M. Luna, and S. Ventura, 'A Grammar-Guided Genetic Programing Algorithm for Associative Classification in Big Data', Cognit. Comput., vol. 11, no. 3, ,pp. 331–346, 2019.

[101] O. May Tzuc, I. Hernández-Pérez, E. V. Macias-Melo, A. Bassam, J. Xamán, and B. Cruz, 'Multi-gene genetic programming for predicting the heat gain of flat naturally ventilated roof using data from outdoor environmental monitoring', Measurement, vol. 138, ,pp. 106–117, 2019.

[102] A. Tahmasebi Poor, A. Barari, M. Behnia, and T. Najafi, 'Determination of the ultimate limit states of shallow foundations using gene expression programming (GEP) approach', Soils Found., vol. 55, no. 3, ,pp. 650–659, 2015.

[103] S. K. Beura and P. K. Bhuyan, 'Operational Analysis of Signalized Street Segments Using Multi-gene Genetic Programming and Functional Network Techniques', Arab. J. Sci. Eng., vol. 43, no. 10, ,pp. 5365–5386, 2018.

[104] A. Fallahpour, E. U. Olugu, S. N. Musa, D. Khezrimotlagh, and K. Y. Wong, 'An integrated model for green supplier selection under fuzzy environment: application of data envelopment analysis and genetic programming approach', Neural Comput. Appl., vol. 27, no. 3, ,pp. 707–725, 2016.

[105] H. H. Chen, C. B. Yang, and Y. H. Peng, 'The trading on the mutual funds by gene expression programming with Sortino ratio', Appl. Soft Comput. J., vol. 15, ,pp. 219–230, 2014.

[106] C. H. Lee, C. B. Yang, and H. H. Chen, 'Taiwan stock investment with gene expression programming', Procedia Comput. Sci., vol. 35, no. C, ,pp. 137–146, 2014.

[107] I. Contreras, J. I. Hidalgo, L. Nuñez-Letamendía, and J. M. Velasco, 'A meta-grammatical evolutionary process for portfolio selection and trading', Genet. Program. Evolvable Mach., vol. 18, no. 4, ,pp. 411–431, 2017.

[108] F. Hauser, J. Huber, and B. Kaempff, 'Costly Information in Markets with Heterogeneous Agents: A Model with Genetic Programming', Comput. Econ., vol. 46, no. 2, ,pp. 205–229, 2015.

[109] H. Qu and X. Li, 'Building Technical Trading System with Genetic Programming: A New Method to Test the Efficiency of Chinese Stock Markets', Comput. Econ., vol. 43, no. 3, ,pp. 301–311, 2014.

[110] B. Ghaddar, N. Sakr, and Y. Asiedu, 'Spare parts stocking analysis using genetic programming', Eur. J. Oper. Res., vol. 252, no. 1, ,pp. 136–144, 2016.

[111] A. Khmag, A. R. Ramli, S. A. R. Al-haddad, S. Yusoff, and N. H. Kamarudin, 'Denoising of natural images through robust wavelet thresholding and genetic programming', Vis. Comput., vol. 33, no. 9, ,pp. 1141–1154, 2017.

[112] S. G. Javed, A. Majid, and Y. S. Lee, 'Developing a bio-inspired multi-gene genetic programming based intelligent estimator to reduce speckle noise from ultrasound images', Multimed. Tools Appl., vol. 77, no. 12, ,pp. 15657–15675, 2018.

[113] M. Yamagiwa, E. Kikuchi, M. Uehara, M. Murakami, and M. Yoneyama, 'Reconstruction for Artificial Degraded Image Using Constructive Solid Geometry and Strongly Typed Genetic Programming', 2009 International Conference on Complex, Intelligent and Software Intensive Systems, IEEE,pp. 162–168, 2009.

[114] P. Thong-On and U. Watchareeruetai, 'Detection of fibrosis in liver biopsy images using multi-objective genetic programming', 2017 9th Int. Conf. Inf. Technol. Electr. Eng. ICITEE 2017, vol. 2018-Janua, ,pp. 1–6, 2017.

[115] M. Iqbal, B. Xue, H. Al-Sahaf, and M. Zhang, 'Cross-Domain Reuse of Extracted Knowledge in Genetic Programming for Image Classification', IEEE Trans. Evol. Comput., vol. 21, no. 4, ,pp. 569–587, 2017.

[116] Í. M. Miranda, C. Aranha, and M. Ladeira, 'Classification of EEG signals using genetic programming for feature construction', Proceedings of the Genetic and Evolutionary Computation Conference, New York, NY, USA,ACM,pp. 1275–1283, 2019.

[117] A. Bhardwaj, A. Tiwari, R. Krishna, and V. Varma, 'A novel genetic programming approach for epileptic seizure detection', Comput. Methods Programs Biomed., vol. 124, ,pp. 2–18, 2016.

[118] M. feli and F. Abdali-Mohammadi, '12 Lead Electrocardiography Signals Compression By a New Genetic Programming Based Mathematical Modeling Algorithm', Biomed. Signal Process. Control, vol. 54, ,p. 101596, 2019.

[119] N. M. Khan and G. M. Khan, 'Audio Signal Reconstruction Using Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN)', 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), vol. 2018-Janua, IEEE,pp. 568–573, 2017.

[120] N. Pillay, 'The impact of genetic programming in education', Genet. Program. Evolvable Mach., no. October 2018, 2019.

[121] W. Xing, R. Guo, E. Petakovic, and S. Goggins, ‘Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory’, Comput. Human Behav., vol. 47, ,pp. 168–181, 2015.

[122] Y. Li, C. Yuan, C. Zhang, S. Li, K. Sun, and X. Wang, ‘A Novel Approximation Algorithm Based on Genetic Programming in Digital Learning Environment’, 2015 International Conference of Educational Innovation through Technology (EITT), IEEE,pp. 33–36, 2015.

[123] S. Liu and H. Shi, ‘A Recursive Approach to Long-Term Prediction of Monthly Precipitation Using Genetic Programming’, Water Resour. Manag., vol. 33, no. 3, ,pp. 1103–1121, 2019.

[124] J. Chadalawada, V. Havlicek, and V. Babovic, ‘A Genetic Programming Approach to System Identification of Rainfall-Runoff Models’, Water Resour. Manag., vol. 31, no. 12, ,pp. 3975–3992, 2017.

[125] A. Danandeh Mehr and V. Nourani, ‘Season Algorithm-Multigene Genetic Programming: A New Approach for Rainfall-Runoff Modelling’, Water Resour. Manag., vol. 32, no. 8, ,pp. 2665–2679, 2018.

[126] A. Aryafar, V. Khosravi, H. Zarepourfard, and R. Rooki, ‘Evolving genetic programming and other AI-based models for estimating groundwater quality parameters of the Khezri plain, Eastern Iran’, Environ. Earth Sci., vol. 78, no. 3, ,pp. 1–13, 2019.

[127] J. Shiri, A. A. Sadraddini, A. H. Nazemi, O. Kisi, G. Landeras, A. Fakheri Fard, and P. Marti, ‘Generalizability of Gene Expression Programming-based approaches for estimating daily reference evapotranspiration in coastal stations of Iran’, J. Hydrol., vol. 508, ,pp. 1–11, 2014.

[128] A. Danandeh Mehr, V. Nourani, E. Kahya, B. Hrnjica, A. M. A. Sattar, and Z. M. Yaseen, ‘Genetic programming in water resources engineering: A state-of-the-art review’, J. Hydrol., vol. 566, no. September, ,pp. 643–667, 2018.

[129] J. I. Hidalgo, J. M. Colmenar, J. L. Risco-Martin, A. Cuesta-Infante, E. Maqueda, M. Botella, and J. A. Rubio, ‘Modeling glycemia in humans by means of Grammatical Evolution’, Appl. Soft Comput. J., vol. 20, ,pp. 40–53, 2014.

[130] L. Vanneschi, ‘Improving genetic programming for the prediction of pharmacokinetic parameters’, Memetic Comput., vol. 6, no. 4, ,pp. 255–262, 2014.

[131] M. B. Erdem, Z. Erdem, and S. Rahnamayan, ‘Diabetes mellitus prediction using multi-objective genetic programming and majority voting’, 14th Int. Conf. Comput. Sci. Educ. ICCSE 2019, no. Iccse, ,pp. 953–958, 2019.

[132] R. Senatore, A. Della Cioppa, and A. Marcelli, ‘Automatic diagnosis of Parkinson disease through handwriting analysis: A cartesian genetic programming approach’, Proc. - IEEE Symp. Comput. Med. Syst., vol. 2019-June, ,pp. 312–317, 2019.

[133] D. Moskowitz, ‘Implementing the template method pattern in genetic programming for improved time series prediction’, Genet. Program. Evolvable Mach., vol. 19, no. 1–2, ,pp. 271–299, 2018.

[134] A. Kattan, S. Fatima, and M. Arif, ‘Time-series event-based prediction: An unsupervised learning framework based on genetic programming’, Inf. Sci. (Ny)., vol. 301, ,pp. 99–123, 2015.

[135] M. Graff, H. J. Escalante, F. Ornelas-Tellez, and E. S. Tellez, ‘Time series forecasting with genetic programming’, Nat. Comput., vol. 16, no. 1, ,pp. 165–174, 2017.

[136] L. Yang, J. Zhang, X. Wu, Y. Zhang, and J. Li, ‘A chaotic time series prediction model for speech signal encoding based on genetic programming’, Appl. Soft Comput. J., vol. 38, ,pp. 754–761, 2016.

[137] F. Yang, M. Li, A. Huang, and J. Li, ‘Forecasting time series with genetic programming based on least square method’, J. Syst. Sci. Complex., vol. 27, no. 1, ,pp. 117–129, 2014.

[138] B. Barán, J. Paciello, V. Cañete, and N. Hernández, ‘Estimating electrical energy consumption using Linear Genetic Programming’, Proceedings of the The International Conference on Engineering & MIS 2015 - ICEMIS ’15, New York, New York, USA,ACM Press,pp. 1–7, 2015.

[139] A. Garg, J. S. L. Lam, and L. Gao, ‘Energy conservation in manufacturing operations:

Modelling the milling process by a new complexity-based evolutionary approach', J. Clean. Prod., vol. 108, ,pp. 34–45, 2015.

[140] I. Boumanchar, K. Charafeddine, Y. Chhiti, F. E. M'hamdi Alaoui, A. Sahibed-dine, F. Bentiss, C. Jama, and M. Bensitel, 'Biomass higher heating value prediction from ultimate analysis using multiple regression and genetic programming', Biomass Convers. Biorefinery, vol. 9, no. 3, ,pp. 499–509, 2019.

[141] V. H. C. Pinheiro and R. Schirru, 'Genetic programming applied to the identification of accidents of a PWR nuclear power plant', Ann. Nucl. Energy, vol. 124, ,pp. 335–341, 2019.

[142] J. Hu and W. Guo, 'Flexibility analysis in waste-to-energy systems based on decision rules and gene expression programming', Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., vol. 2019-Octob, ,pp. 988–993, 2019.

[143]'DEAP',https://deap.readthedocs.io/en/master/,accessed 6 February 2020.

[144]'GPLEARN',https://gplearn.readthedocs.io/en/stable/,accessed 6 February 2020.

[145]'Matlab',https://www.mathworks.com/,accessed 7 February 2020.

[146] 'GELAB: A Matlab ToolBox',https://github.com/adilraja/GELAB, accessed 7 February 2020.

[147] 'CGP4Matlab',https://github.com/tiagoinacio/CGP4Matlab,accessed 7 February 2020.

[148] R. Miragaia, G. Reis, F. Fernandéz, T. Inácio, and C. Grilo, 'CGP4Matlab - A Cartesian Genetic Programming MATLAB Toolbox for Audio and Image Processing', Applications of Evolutionary Computation April 4-6, 2018, Parma, Italy,Springer, Cham,pp. 455–471, 2018.

[149] 'GPTIPS 2',https://sites.google.com/site/gptips4matlab/home,accessed 7 February 2020.

[150] I. Pan and S. Das, 'When Darwin meets Lorenz: Evolving new chaotic attractors through genetic programming', Chaos, Solitons & Fractals, vol. 76, ,pp. 141–155, 2015.

[151] D. P. Searson, 'GPTIPS 2: An Open-Source Software Platform for Symbolic Data Mining', Cham,Springer International Publishing,pp. 551–573, 2015.

[152] P. K. Muduli, M. R. Das, S. K. Das, and S. Senapati, 'Lateral Load Capacity of Piles in Clay Using Genetic Programming and Multivariate Adaptive Regression Spline', Indian Geotech. J., vol. 45, no. 3, ,pp. 349–359, 2015.

[153] D. P. Searson, D. E. Leahy, and M. J. Willis, 'Predicting the Toxicity of Chemical Compounds Using GPTIPS: A Free Genetic Programming Toolbox for MATLAB', ,pp. 83–93, 2011.

[154]'PonyGE2',https://github.com/PonyGE/PonyGE2/,accessed 7 February 2020.

[155] M. Fenton, J. McDermott, D. Fagan, S. Forstenlechner, E. Hemberg, and M. O'Neill, 'PonyGE2: grammatical evolution in Python', Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17, New York, New York, USA,ACM Press,pp. 1194–1201, 2017.

[156]'GeneXproTools',https://www.gepsoft.com/,accessed 7 February 2020.

[157] Y. Sun, Z. Li, C. Tang, W. Zhou, and R. Jiang, 'An Evolving Neural Network for Authentic Emotion Classification', 2009 Fifth International Conference on Natural Computation, vol. 2, IEEE,pp. 109–113, 2009.

[158] S. Mehdizadeh, F. Fathian, M. J. S. Safari, and A. Khosravi, 'Developing novel hybrid models for estimation of daily soil temperature at various depths', Soil Tillage Res., vol. 197, ,p. 104513, 2020.

[159]'GEP4J',https://code.google.com/p/gep4j/,accessed 7 February 2020.

[160]'PyGEP',https://code.google.com/p/pygep/.,accessed 7 February 2020.

[161]'JGEP',https://sourceforge.net/projects/jgep/.,accessed 7 February 2020.

[162]'Epochx',https://www.epochx.org/,accessed 7 February 2020.

[163] 'Karoo GP',http://kstaats.github.io/karoo_gp/,accessed 7 February 2020.

[164]'Tensorflow',https://www.tensorflow.org/,accessed 7 February 2020.

[165]'GISMO',http://www0.cs.ucl.ac.uk/staff/W.Langdon/gismo/,accessed 8 February 2020.

[166] W. B. Langdon, ‘Genetic Improvement of Software for Multiple Objectives’, ,pp. 12–28, 2015.

[167] W. B. Langdon and M. Harman, ‘Optimizing existing software with genetic programming’, IEEE Trans. Evol. Comput., vol. 19, no. 1, ,pp. 118–135, 2015.

[168]‘KNIME’,https://www.knime.com/,accessed 8 February 2020.

[169] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, ‘KNIME - the Konstanz information miner’, ACM SIGKDD Explor. Newsl., vol. 11, no. 1, ,pp. 26–31, 2009.

[170] S. O’Hagan and D. B. Kell, ‘Software review: the KNIME workflow environment and its applications in genetic programming and machine learning’, Genet. Program. Evolvable Mach., vol. 16, no. 3, ,pp. 387–391, 2015.

[171] ‘GSGP-C++ 2.0’,http://gsgp.sourceforge.net/,accessed 8 February 2020.

[172] M. Castelli, S. Silva, and L. Vanneschi, ‘A C++ framework for geometric semantic genetic programming’, Genet. Program. Evolvable Mach., vol. 16, no. 1, ,pp. 73–81, 2015.

[173] M. Castelli and L. Manzoni, ‘GSGP-C++ 2.0: A geometric semantic genetic programming framework’, SoftwareX, vol. 10, ,p. 100313, 2019.

[174] ‘GPLAB’,http://gplab.sourceforge.net/,accessed 8 February 2020.

[175] S. Silva and J. Almeida, ‘GPLAB-a genetic programming toolbox for MATLAB’, Proc. Nord. MATLAB Conf., ,pp. 273–278, 2003.

[176] V. K. Dabhi and S. Chaudhary, ‘Empirical modeling using genetic programming: a survey of issues and approaches’, Nat. Comput., vol. 14, no. 2, ,pp. 303–330, 2015.

[177] W. Banzhaf, ‘Genetic programming and emergence’, Genet. Program. Evolvable Mach., vol. 15, no. 1, ,pp. 63–73, 2014.

[178] L. Vanneschi, M. Castelli, and S. Silva, ‘A survey of semantic methods in genetic programming’, Genet. Program. Evolvable Mach., vol. 15, no. 2, ,pp. 195–214, 2014.

[179] I. Costa-Carrapiço, R. Raslan, and J. N. González, ‘A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency’, Energy Build., vol. 210, ,p. 109690, 2020.

[180] Q. Zhang, K. Barri, P. Jiao, H. Salehi, and A. H. Alavi, ‘Genetic programming in civil engineering: advent, applications and future trends’, Artif. Intell. Rev., 2020.

[181] J. Zhong, L. Feng, and Y.-S. Ong, ‘Gene Expression Programming: A Survey’, IEEE Comput. Intell. Mag., vol. 12, no. 3, ,pp. 54–72, 2017.