

A Comparative Study of Automatic Programming Techniques

Sibel Arslan and Celal Öztürk

Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Turkey

E-mail: sibel.arslan2@icisleri.gov.tr, celal@erciyes.edu.tr

Keywords: automatic programming, genetic programming, artificial bee colony programming, symbolic regression, prediction, feature selection

Received: January 3, 2018

Automatic programming, an evolutionary computing technique, forms the programs automatically and is based on higher level features that can be easily specified than normal programming languages. Genetic Programming (GP) is the first and best-known automatic programming technique that is applied to solve many practical problems. Artificial Bee Colony Programming (ABCP) is one of the latest proposed automatic programming method that combines evolutionary approach with swarm intelligence. GP is an extension version of Genetic Algorithm (GA) and ABCP is based on Artificial Bee Colony (ABC) algorithm. The main differences of these automatic programming techniques and their conventional algorithms (GA and ABC) are modeling solution. In ABC same as GA, the solutions are represented fixed code blocks. In GP and ABCP, the positions of food sources are expressed in tree structure that is composed of different combinations of terminals and functions that are specifically defined as problems. This paper presents a review on GP and ABCP and they are worked in symbolic regression, prediction and feature selection problems which are widely tackled by researchers. The results of the ABCP compared with results of GP show that this algorithm is a powerful optimization technique for structural design.

Povzetek: Predstavljena je primerjalna analiza tehnik avtomatskega programiranja.

1 Introduction

Computer programming is the process of obtaining a program that can be executed machine to use the necessary information to perform a task. Automatic programming is a computer programming technique which automatically generates the program code [1]. It provides practical solutions for many machine learning methods such as Artificial Neural Network (ANN), Decision Trees (DT), Support Vector Machines (SVM), Genetic Programming (GP), Artificial Bee Colony Programming (ABCP). GP, most well-known automatic programming method, was developed by Koza [2]. GP is an extension of Genetic Algorithm (GA) and the basic steps for the GP algorithm are similar to the steps of GA. ABCP is recently proposed automatic programming technique which is based on the Artificial Bee Colony Algorithm (ABC) [3]. The goal of this paper is to evaluate the success of the models obtained from these automatic programming methods in symbolic regression, prediction and feature selection problems and review papers related to the problems.

Symbolic regression is a type of regression problem aimed at finding the most appropriate mathematical model in terms of accuracy and complexity of data. There are works investigating the problem of symbolic regression with automatic programming techniques, mostly with GP [3-10]. ABCP was proposed for the first time as a new method for the symbolic regression problem and compared with GP [3]. Faris proposed solving the symbolic regression problem using GP model was compared to least square estimation, GA and particle

swarm intelligence models based on estimating the parameters of the nonlinear regressive curve of the cutting tool [4]. According to the benchmarks in the paper, GP was found to be superior performance. In [5], two versions of GP (standard GP and multi-population GP) were compared with ANN on pharmaceutical formulation symbolic regression problem. Compared to successful ANN models, GP models provided a significant advantage, parametric equations that can be interpreted and analyzed more easily. Gene Expression Programming (GEP) [6], Immune Programming (IP) [7], Ant Colony Programming (ACP) [8-10] are other automated programming techniques that used in the problem of symbolic regression. GEP, proposed on both GA and GP, is flexible at genetic operations due to its linear code blocks and its parse trees [6]. IP is based on Artificial Immune System (AIS) and is a domain-independent approach [7] where antigens can be represented in the tree structure express solutions similar to GP. ACP [8, 9] and Dynamic Ant Colony Programming (DAP) which dynamic version of ACP [10] are the main ACP examples that are inspired on ant colony algorithm.

Automatic programming techniques have been solved prediction problems where most of applications are based on evolutionary optimization techniques [11-19]. Seidy proposed a new stock market prediction model using the Particle Swarm Optimization with Center of Mass Technique (PSOCOM) which was more successful results than the particle swarm optimization based models according to the prediction accuracy [11]. Manjusha et al. used Naive Bayes and J48 algorithms to diagnose potentially fatal dermatological diseases with similar

symptoms [12]. They developed the interface which the probability of recurrence of each disease was predicted. In [13] Box-Jenkins (BJ) and ANN were used together to model monthly water consumption in Kuwait. The input layer variables in the artificial neural network were obtained with BJ, the average error was considered, and the more successful results were obtained than the traditional methods. Ant Colony Optimization (ACO) technique was used to generate qualified bankruptcy prediction rules [15]. The Association Rule Miner (ARM) technique was used to group rules and eliminate irrelevant rules in the paper. Etemadi et al. compared Multiple Discriminant Analysis (MDA) and the GP in bankruptcy prediction modeling [18]. GP model was found to produce more accurate results than the MDA model, which is produced as an accurate bankruptcy forecasting model considering both the quality of the sample companies and the estimated duration. Searson et al. used multigene GP demonstrating it with an application in which a predictive symbolic Quantitative Structure Activity Relationship (QSAR) model of *T. pyriformis* aqueous toxicity was as successful as the QSAR models on the same data [19].

In the recent researches, the increase in the number of features in the data sets necessitated the use of feature selection methods. These methods are used to eliminate noisy and unnecessary features in collected data so that the data set can be expressed more reliably and at the same time classification achieve high success rates. Various optimization methods have been applied to solve feature selection problems [20-33]. Lujan et al. proposed an automatic programming technique called quadratic programming based on quadratic function optimization for multiclass problems [22]. The work was found more efficient than the Maximal Relevance (MaxRel) and Minimal-Redundancy-Maximum-Relevance (mRMR) on large data sets. In [23], statistical and entropy-based feature ranking methods were compared with different data sets. It was shown that the accuracy of the classifier was influenced by the choice of ranking indices. Brown et al. investigated the apparent statistical assumptions of feature selection criteria based on mutual knowledge [24]. They derived the objective function using the conditional probabilities of the training labels. When the results were evaluated, the Joint Mutual Information (JMI) criterion provided the best balance of accuracy, stability and flexibility criteria for small data sets. In [27], two-stage automatic programming technique called differential development based Named Entity Recognition (NER) was proposed. In the first stage, Conditional Random Field (CRF) and SVM classifiers were used in feature selection problems. In the second stage, classifiers according to F scale score were selected and combined using differential development based classifier collection technique. The technique was more successful than the other traditional methods. Yu et al. showed that the GP can be used as a feature selector and cancer classifier [30]. Selecting the discriminative genes of GP, expressing the relations between the genes as mathematical equations were proof that GP can be used in this field. In addition, training sets and GP classifiers obtained from the validation set in the work were tested GP successfully classified tumor classes

and were more successful than various classification methods. The (k-Nearest Neighbor (k-NN) and GP based decision trees are applied to feature selection and were compared in terms of classification performance in [31]. Arqub et al. proposed an algorithm based on GA for the solutions of nonlinear systems of second-order boundary value problems [32]. The results show that the algorithm is very effective and convenient in linear and nonlinear cases with less computational generation and less time. Continuous Genetic Algorithm (CGA) is introduced solving systems of second-order boundary value problems [33]. The influence of different parameters, including the initialization method, the selection method, the rank-based ratio, the evolution of nodal values, the population size, the crossover and mutation probabilities, the step size, and the maximum nodal residual is studied in the paper. The algorithm had better performance than some modern methods.

GP and ABCP is a successful automatic programming techniques which were based on GA and ABC. In this paper, we have compared GP and ABCP on the main applications of automatic programming which are symbolic regression, prediction and feature selection problems. The organization of the paper is as follows: GP is described in Section 2, ABCP is introduced in Section 3, and Experimental Design is presented Section 4 and Results are discussed in Section 5. The paper is concluded in Section 6 with remarking the future work.

2 Genetic programming

GP, most well-known automatic programming method, expresses solutions as tree structures. The trees are randomly generated according to tree depth which is previously determined. The production of tree nodes is provided by terminals (constants or variables such as x , y , 5) and functions (arithmetic operators such as $+$, $-$, $/$, \max). The representation of the tree is shown in Fig. (1) [34]. The root node connects to the branches each of them consists of more than one component. In all cases the model of solution is found by analyzing the entire tree.

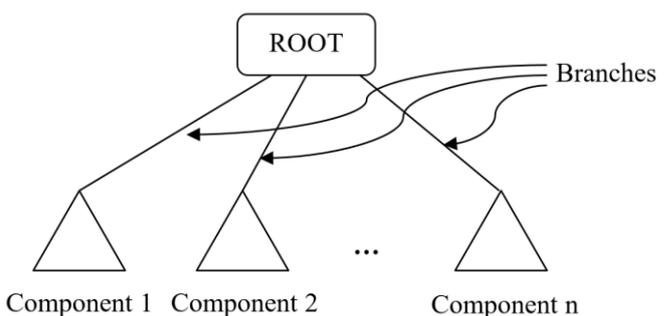


Figure 1: Representation of tree.

A flow chart of GP is given in Fig (2). Initial population is produced and the fitness of the solutions according to the determined fitness function is assessed. The production of the individuals in the initial population is based on full method, grow method, or ramped half-and-half method [35]. In the full method, nodes are selected from the function set until they reach the maximum tree

depth. In the grow method, nodes are randomly selected from a set of all terminals and functions. In the ramped half-and-half method, 50% full and 50% grow method are used to produce trees in various widths and depths [2]. GP aims to increase the number of individuals with high quality survival and decrease the number of low quality individuals. Individuals with high quality are more likely to pass on to the next generation. Almost all selection operators of GA can be used (mostly tournament selection methods) in GP [36]. Individuals the optimization of problems is developed them with exchange operators such as reproduction, crossover and mutation. The crossover operator allows the hybrid of two selected individuals to produce a new individual. The subtrees taken from the two randomly selected crossover points of the parent trees are crossed to obtain new trees. The mutation operator provides unprecedented and unexplored individuals. A randomly generated node or tree is usually exchanged in the mutation process instead of the node selected from the tree. The best individuals of the previous generation are transferred to the current generation with elitism. The stopping criterion is checked that the individuals reach a certain value or the predetermined number of generations and then, the program is terminated when the stopping criterion is satisfied.

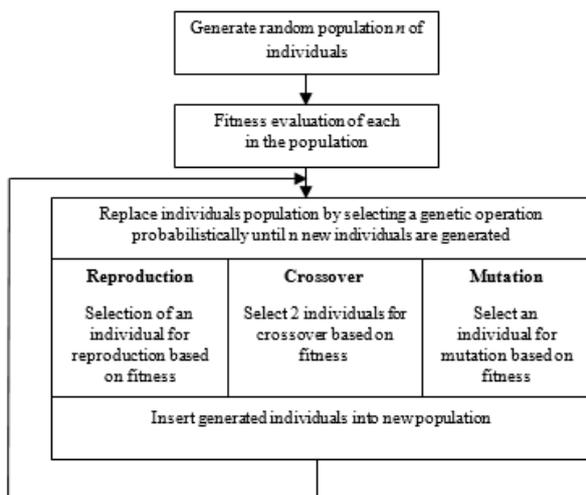


Figure 2: The flow chart of Genetic Programming.

3 Artificial bee colony programming

ABC is a swarm intelligence optimization algorithm that simulates the behavior of honeybees and provides a solution for multi variable problems [37]. ABCP, based on ABC algorithm, was introduced first time as a new method for symbolic regression [3]. In ABC, the positions of the food sources are represented with fixed size arrays. In the ABCP method, the positions of food sources are expressed in tree structure that is composed of different combinations of terminals and functions that are specifically defined for problems [41]. The mathematical relationship of the solution model in ABCP can be represented the individuals in Fig. (3) is described Eq. (1).

In these notations, x is used to represent the independent, and $f(x)$ is dependent.

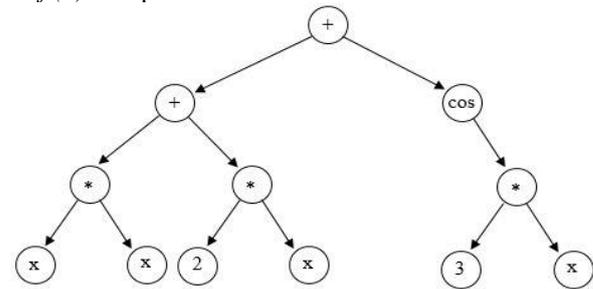


Figure 3: Representation of an example solution in ABCP with tree structure.

$$f(x) = [(x^2 + 2x)] + [\cos 3x] \tag{1}$$

There are three different types of bees in ABCP, each of which is responsible for a food source, employed bee, onlooker bee and scout bee. The position of a food source express a solution (a single parse tree). The number of employed bees is equal to the number of onlooker bees. Quality of the food source in terms of nectar is expressed through the fitness function of the solution. The employed bees search new food sources and shares information about food sources with the onlooker bees. They tend to be more inclined toward quality food sources in line with the information they receive from their employed bees. If a source is abandoned, the employed bee becomes a scout bee that starts to look for a new source randomly. The exhausted of food resources controls a parameter called "limit". For each source, the number of improving trials is kept, and in each cycle, it is checked to see whether the number of trials exceeds the "limit" parameter. If the food source is exhausted, the source is abandoned. Employed bees of an abandoned source turn into a scout bee and look randomly for a new source. Algorithmic steps for the ABCP are given in detail in Fig. (4).

ABCP starts with the production of food source in the initial phase. Similar to producing GP's solutions, food sources by full method, grow method or ramped half and half method are produced [2]. The main difference between ABCP and ABC is the neighborhood mechanism in generating candidate solutions [3]. When a candidate solution (v_i) is generated based on the node x_i which represents the current solution in the tree and a neighbor node solution x_k which is randomly taken from the tree considering predetermined probability p_{ip} . The node selected from the neighbor solution x_k determines what information will be shared with the current solution and how much it will be shared. This sharing mechanism is shown in Fig. (5). Figure 5a and 5b are: node x_i representing the current solution and neighbor node x_k , respectively; neighboring information and the generated candidate solution are given in Figure 5c and Figure 5d, respectively. If the quality of the candidate solution v_i is better than of the current solution x_i , v_i is selected on the other case x_i is going on.

Employed bees share the information they have gained after completing the research process in the sources they are related to with onlooker bees. They select source according to the probability values calculated by Eq. (2)

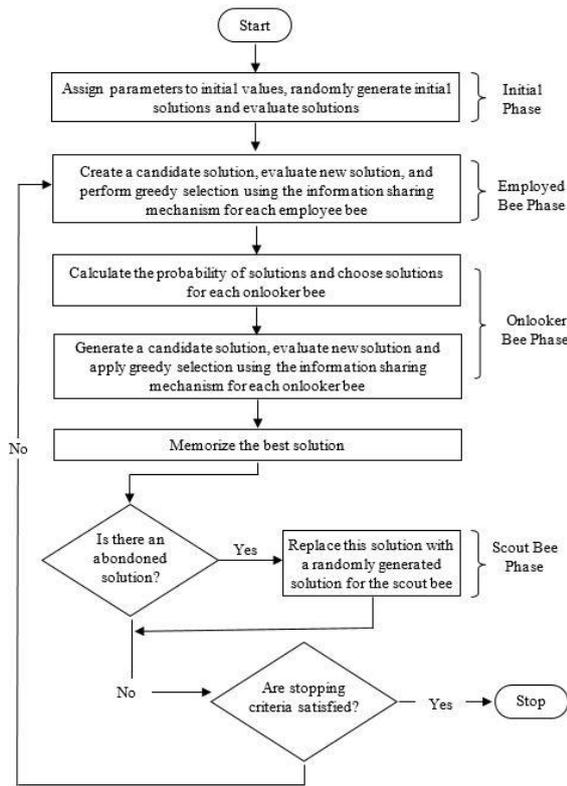


Figure 4: The flow chart of Artificial Bee Colony Programming.

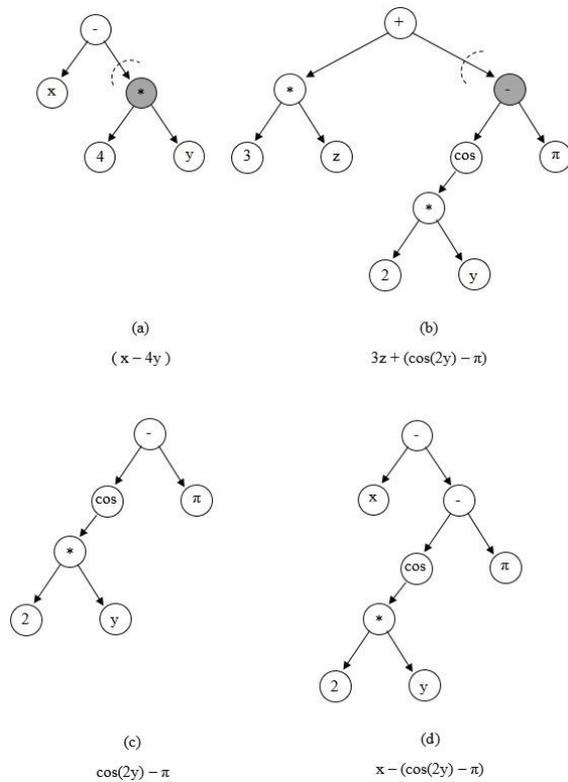


Figure 5: Example of information sharing mechanism.

depending on the nectar quantities of sources within the information they receive from employed bees' information.

$$p_i = \frac{0.9 * fit_i}{fit_{best}} + 0.1 \quad (2)$$

Where fit_i quality of the solution i , fit_{best} quality of the best solution current solutions. If a source is more qualified, the probability of selecting the source increases. After selecting the sources to be searched, the onlooker bees find new sources like employed bees. The amount of nectar of the new found source is checked. If the new source has a higher amount of nectar, the new source is taken to memory and the old source is deleted from memory. Therefore, the onlooker bees show a greedy selection like the employed bees.

In ABCP, the penalty point of the relevant sources is increased by one when no better sources can be found for each employed bee and onlooker bee. When a better source of any source is discovered, that source's penalty point is reset. Once all the employed bees and onlooker bees have completed the search operations in each cycle, the penalty points of sources are checked [42].

4 Experimental design

This section, three different experiments with benchmark data sets have been studied and performance values of GP and ABCP have been compared using similar parameter values and the results have been discussed.

4.1 Experiments

In the first experiment, the performance of models evolved by GP and ABCP are evaluated in the symbolic regression. Training data set for the 4-input non-linear Cherkassy function expressed in Eq. (3) [38]. The objective of the GP and ABCP are to evolve a symbolic function of x_1 , x_2 , x_3 , and x_4 that closely approximates y .

$$y = \exp(2x_1 \sin(\pi x_4)) + \sin(x_2 x_3) \quad (3)$$

In the second experiment, the output values in the pH data set [39] are predicted. The data in this experiment is taken a simulation of a pH neutralization process with one output (pH), which is a non-linear function of the four inputs.

Concrete pressure compressive strength data [40] is taken to study feature selection performance of GP and ABCP in the last dataset. Concrete pressure compressive strength data is highly nonlinear of input values. The outputs being modelled are produced by automatic programming methods of the concrete data and the independent variables are: cement (x_1), blast furnace slag (x_2), fly ash (x_3), water (x_4), superplasticizer (x_5), Coarse aggregate (x_6), fine aggregate (x_7), age (x_8). The noise, which was added to concrete dataset, consist of 50 input variables ($x_9, x_{10}, \dots, x_{58}$) with random values in range [-500,500]. The number of inputs and instances of the problems are shown in Table 1.

4.2 Fitness function - parameters

The performance of models are evaluated by the Root Mean Square Error (RMSE) on both the training set and the test set. The fitness function is shown Eq. (4).

| Name | #Inputs | #Total Instances | #Training Instances | #Test Instances | Noise |
|-----------|---------|------------------|---------------------|-----------------|-------------------------------|
| Cherkassy | 4 | 500 | 400 | 100 | - |
| pH | 4 | 990 | 700 | 299 | - |
| Concrete | 8 | 1030 | 773 | 257 | 50 input variables [-500,500] |

Table 1: Benchmark Problems.

$$fitness = \sqrt{\frac{\sum_{t=1}^n (y_{pred} - y_{actual})^2}{n}} \quad (4)$$

Where n define the data size, y_{actual} is y values from data set, y_{pred} is the predicted y value by obtained solution.

The complexity of the solution is calculated as in Eq. (5) in proportion to the depth of the tree and the number of nodes.

$$C = \sum_{k=1}^d n * k \quad (5)$$

Where C is tree complexity, d is the depth of the solution, and n is the number of nodes at depth. The parameters for GP and ABCP are summarized in Table 2. The *add3* function is the sum of three variables ($x_1 + x_2 + x_3$) and the *mult3* function is the multiplication of three variables ($x_1 * x_2 * x_3$). If the divisor value is equal to zero, the result is 1, otherwise the normal division is performed in the *rdivide* function. The *ifbte* and the *iflte* indicates the condition of the nodes. Eq. (6) and Eq. (7) describe how the functions operate condition expressions.

$$X = ifbte(A, B, C, D) \\ if(A \geq B) then X = C else X = D \quad (6)$$

$$X = iflte(A, B, C, D) \\ if(A < B) then X = C else X = D \quad (7)$$

The population size was taken high value according to the curse of the data sets in the literature. When experiments are evaluated, it is observed that the optimum results were obtained where population/ colony size was set to 100 for Cherkassy, 200 for pH and 300 for Concrete. Therefore, other parameters like generation number and maximum tree depth values was set to make fair test with same parameters like in the literature. In this work, the stop criterion is decided by using the maximum generation number for both GP and ABCP.

5 Results & discussions

This section demonstrate symbolic regression, prediction and feature selection abilities of ABCP and GP, set of experiments conducted.

5.1 Simulation Results

The experiments are run 30 times independently for ABCP and GP and the obtained results are demonstrated in Table 3 for the problems. The R^2 values of the best cases of GP and ABCP are also presented in the table for training and test sets.

| Parameters | Problems | | | | | |
|---------------------------|--|------|---------------------------|------|---|------|
| | Cherkassy | | pH | | Concrete | |
| | GP | ABCP | GP | ABCP | GP | ABCP |
| Population / Colony size | 100 | 100 | 200 | 200 | 300 | 300 |
| Iteration size | 100 | 100 | 200 | 200 | 500 | 500 |
| Maximum tree depth | 12 | 12 | 12 | 12 | 12 | 12 |
| Tournament size | 6 | - | 25 | - | 15 | - |
| Mutation ratio | 0.14 | - | 0.14 | - | 0.14 | - |
| Crossover ratio | 0.84 | - | 0.84 | - | 0.84 | - |
| Direct reproduction ratio | 0.02 | - | 0.02 | - | 0.02 | - |
| Functions | +,-, *, tan, sin, cos, square, max, min, exp, ifbte, iflte | | +,-, *, tanh, add3, mult3 | | +,-, *, rdivide, sin, cos, exp, rlog, add3, mult3 | |
| Constans | [-10,10] | | [-10,10] | | [-10,10] | |

Table 2: Parameters for GP and ABCP.

| Metric | Criteria | Problems | | | | | |
|--------|---------------|-----------|------|-------|------|----------|-------|
| | | Cherkassy | | pH | | Concrete | |
| | | GP | ABCP | GP | ABCP | GP | ABCP |
| Mean | RMSE | 0.07 | 0.03 | 0.92 | 0.77 | 11.64 | 10.50 |
| Std | RMSE | 0.03 | 0.02 | 0.14 | 0.07 | 2.38 | 1.51 |
| Best | RMSE | 0.02 | 0.01 | 0.60 | 0.59 | 8.83 | 8.26 |
| Worst | RMSE | 0.11 | 0.09 | 12.60 | 0.88 | 16.77 | 14.57 |
| Best | R^2_{train} | 0.97 | 1.00 | 0.90 | 0.96 | 0.72 | 0.76 |
| Best | R^2_{test} | 0.98 | 1.00 | 0.91 | 0.96 | 0.70 | 0.73 |

Table 3. Obtained results of GP and ABCP.

While obtaining the results of GP on symbolic regression problem, the GPTIPS (an open source symbolic regression solution toolbox) [39] is modified and used in this work. It indicates ABCP has much better training

performance than GP on these datasets. The best mean fitness value of all runs was found in the 0.0323 Cherkassy symbolic regression problem in ABCP. The standard deviation of ABCP is in 10% range where it is in 20% range at GP for pH dataset. However all criteria are worse than the other datasets due to the noise in the concrete dataset, ABCP has comparable performance than GP.

The curve fitting of the y_{actual} and the y_{pred} values for the training and test data set at this best fitness value is expressed in Figure 6 for Cherkassy, Figure 7 for pH, Figure 8 for Concrete dataset in ABCP.

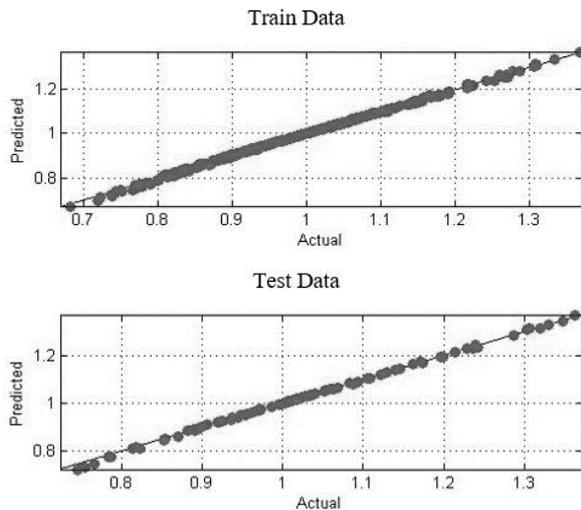


Figure 6: Predicted and actual data points on Cherkassy in best ABCP.

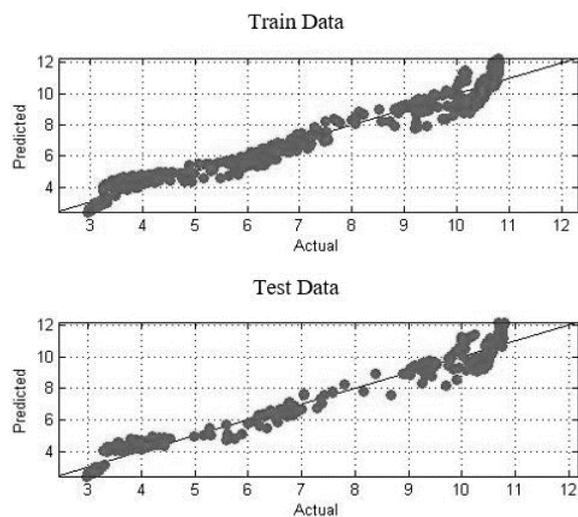


Figure 7: Predicted and actual data points on pH in best ABCP.

Figures 6-7 and 8 show the evolution plots for all datasets. The y_{actual} and the y_{pred} values are close to each other for training and test data as seen from the curves.

5.2 Analysis of evolved models

The evolved models of best solutions in ABCP of all runs are shown in Table 4. Mathematical models have been obtained using all inputs in the Cherkassy and pH datasets. The evolved model for Concrete has *Blast*

Furnace Slag, Age, Cement, Plastic input parameters of 8 input parameters are selected. It should be noticed that only the x_{17} input is taken from the 50 added noise parameters. The presence of only one of the noise parameters in the equation is an indication that the ABCP is successful in feature selection.

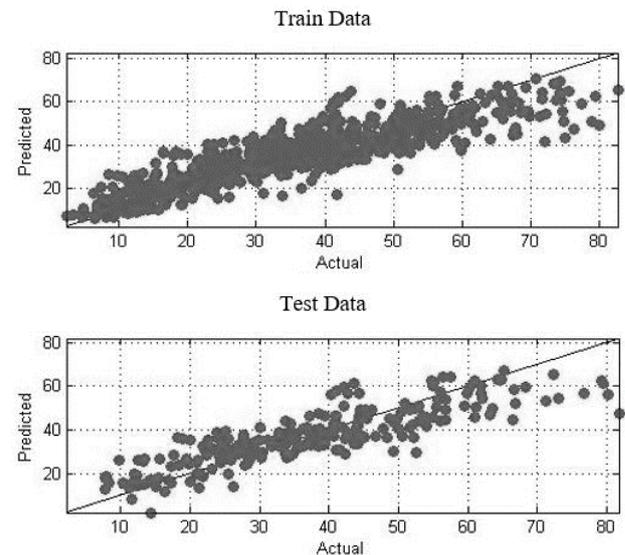


Figure 8: Predicted and actual data points on Concrete in best ABCP.

The total number of nodes tree has, tree depth, tree complexity of the best solution is given Table 5 for each dataset. As seen in Table 5, the noise parameters added to the inputs in the problems increase the difficulty of the problems. Increasing the difficulty enlarges the solution trees and increases their complexity. Since the Cherkassy function is easier than other problems, the complexity of the tree is the least problem.

| Problem | ABCP | | |
|-----------|-----------------------|---------------------------------|-------------------------------|
| | Total number of nodes | Depth of the best solution tree | Best solution tree complexity |
| Cherkassy | 25 | 7 | 118 |
| pH | 72 | 11 | 467 |
| Concrete | 80 | 12 | 657 |

Table 5: Best solution tree information for each data set.

6 Conclusion

In this paper, the automatic programming methods have been examined on symbolic regression, prediction and feature selection. The results of the symbolic regression on Cherkassy function, prediction on pH and feature selection concrete datasets are used to compare GP and ABCP methods. In all three experiments, ABCP demonstrate higher performance than GP in terms of finding more accurate mathematical modeling in symbolic

| Dataset | Model of Best of Run Individual | Number of Inputs |
|-----------|---|------------------|
| Cherkassy | $y = ((\exp(3x_4x_1) * (\exp(x_2x_3) + x_4x_1)) + x_4x_1) * \exp(x_4x_1))$ | 4 |
| pH | $y = A + B + C$ $A = (\tanh(\tanh(x_4 + (x_3 * \tanh(x_2)))) + 2x_3) * \tanh(\tanh(x_2)) - x_2$ $B = x_2^2 * (\tanh((x_4 - (x_1 - 8.935 + (x_1 + x_3 + x_4) * \tanh(\tanh(x_2)))) * \tanh(x_2))) + x_2$ $C = (x_4 + (x_2 + \tanh(\tanh(\tanh(x_2)))) + \tanh(\tanh(2x_2 + x_4))) + ((x_2^2 * x_3) + x_2) + \tanh(x_2) + \tanh(\tanh(x_2) - (x_1 - x_3)) - x_4$ | 4 |
| Concrete | $y = \left \log\left(\frac{A}{B}\right) + (C) \right $ $A = \left \tanh\left(\tanh\left(\sqrt{\sqrt{Age}}\right)\right) + \left(\frac{\sqrt{\log(Cement)}}{-2.997} + Slag\right) * Age \right.$ $\left. * \frac{\tanh\left(\tanh\left(\log\left(\frac{Plastic}{-3.877}\right)\right)\right)}{\frac{\tanh(\tanh(x_{17}))}{(-3.877)}} \right $ $B = (\sqrt{Cement})^2 + (Cement * (-3.877))^2$ $C = (-3.877) + \left \left(\sqrt{\log(Age)^2} * \sqrt{Cement} - \left(\frac{\tanh(\tanh(Plastic))}{\tanh\left(\frac{\tanh(Cement)}{-3.877}\right)} \right) \right) \right $ $- \tanh\left(\tanh\left(\log\left(\frac{Plastic}{-3.877}\right)\right)\right) / \tanh\left(\tanh\left(\frac{\tanh(Cement)}{-3.877}\right)\right)$ | 5 |

Table 4: Models of Best Run ABCP and GP.

regression, better fitting in prediction ability and effective in finding important features along with the presence of redundant features.

In the future, we are intending to investigate several interesting researches. Simulation works will be done to model the fundamental classification problems (cancer, diabetes, heart, gene diseases etc.) with GP and ABCP to get performance results. In addition, we are planning to work the Multi-Gen Genetic Programming and Multi-Hive ABCP and compare the results with standard GP and ABCP to enhance the symbolic regression, prediction and future selection abilities of solutions.

Acknowledgement

This project was supported by Scientific Research Project Foundation of Erciyes University (Project ID: FBA-12-4029).

References

[1] Alan.W. Biermann. *Automatic Programming: A Tutorial on Formal Methodologies*, J. Symbolic Computation, pp. 119-142, 1985. [https://doi.org/10.1016/s0747-7171\(85\)80010-9](https://doi.org/10.1016/s0747-7171(85)80010-9)

[2] John. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992. <https://doi.org/10.1007/BF00175355>.

[3] Dervis Karaboga, Celal Ozturk, Nurhan Karaboga and Beyza Gorkemli. *Artificial bee colony programming for symbolic regression*, Information Sciences, 209, pp. 1–15, 2012. <https://doi.org/10.1016/j.ins.2012.05.002>

[4] Hossam Faris. *A Symbolic Regression Approach for Modeling the Temperature of Metal Cutting Tool*, International Journal of Control and Automation. 6(4), 2013.

[5] Panagiotis Barmpalexis, Kyriakos Kachrimanis, Athanasios Tsakonas, E. Georgarakis. *Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation*, Chemometrics and Intelligent Laboratory Systems, 107, pp. 75–82, 2011. <https://doi.org/10.1016/j.chemolab.2011.01.012>

[6] Xin Li. *Self-Emergence of Structures in Gene Expression Programming*, Ph.D. Thesis, University of Illinois at Chicago, 2006.

[7] Petr Musilek, Adriel Lau, Marek Reformat and Loren Wyard-Scott, *Immune programming*, Information Sciences, 176, pp. 972–1002, 2006.

- <https://doi.org/10.1016/j.ins.2005.03.009>.
- [8] Mariusz Boryczka, *Ant colony programming: application of ant colony system to function approximation*, Intelligent Systems for Automated Learning and Adaptation: Emerging Trends and Applications, pp. 248–272, 2010.
<https://doi.org/10.4018/978-1-60566-798-0.ch011>.
- [9] Olivier Roux, Cyril Fonlupt. *Ant programming or, how to use ants for automatic programming*, Proceedings of ANTS'2000, pp. 121–129, 2000.
- [10] Shinichi Shirakawa, Shintaro Ogino, Tomoharu Nagao. T. Nagao, *Dynamic ant programming for automatic construction of programs*, IEEE Transactions on Electrical and Electronic Engineering, pp. 540–548, 2008.
<https://doi.org/10.1002/tee.20311>.
- [11] Essam.El. Seidy. *A New Particle Swarm Optimization Based Stock Market Prediction Technique*, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, No. 4, 2016.
<https://doi.org/10.14569/ijacsa.2016.070442>.
- [12] K.K. Manjusha, K. Sankaranayanan, P. Seenaa. *Data Mining in Dermatological Diagnosis: A Method for Severity Prediction*, International Journal of Computer Applications, (0975 – 8887) Vol. 117 – No.11, 2015.
- [13] Sana BuHamra, Nejib Smaoui, Mahmoud Gabr. *The Box–Jenkins analysis and neural networks: prediction and time series modelling*, Applied Mathematical Modelling, Vol: 27, pp. 805–815, 2003.
[https://doi.org/10.1016/s0307-904x\(03\)00079-9](https://doi.org/10.1016/s0307-904x(03)00079-9).
- [14] Gianluca Bontempi. *Machine Learning Strategies for Time Series Prediction, Machine Learning Group*, Computer Science Department Boulevard de Triomphe - CP 212, Hammamet, 2013.
Retrieved from: <http://www.ulb.ac.be/di>.
- [15] A. Martin, V. Aswathy, V.Prasanna Venkatesan. *Framing Qualitative Bankruptcy Prediction Rules Using Ant Colony Algorithm*, International Journal of Computer Applications, 0975 – 8887 ,Volume 41– No.21, 2012.
<https://doi.org/10.5120/5827-8143>.
- [16] Shuzhan Wan, Shengwu. Xiong and Yi Liu, *Prediction based multi-strategy differential evolution algorithm for dynamic environments*, Evolutionary Computation (CEC), 2012 IEEE Congress, pp. 10-15, 2012.
<https://doi.org/10.1109/cec.2012.6256628>.
- [17] Dandan .Li, Wanxin. Xue, Yilei Pei . *A high-precision prediction model using Ant Colony Algorithm and neural network*, International Conference on Logistics, Informatics and Service Sciences (LISS), 2015.
<https://doi.org/10.1109/liss.2015.7369696>.
- [18] Hossein Etemadi, Ali Asghar Anvary Rostamy, Hossain Farajzadeh Dehkordi . *A genetic programming model for bankruptcy prediction: Empirical evidence from Iran*, Expert Systems with Applications, Vol: 36, pp. 3199–3207, 2009.
<https://doi.org/10.1016/j.eswa.2008.01.012>.
- [19] Pamela Dominic, David Edward Leahy, Mark J. Willis. *Predicting the toxicity of chemical compounds using GPTIPS: a free open source genetic programming toolbox for MATLAB*, Intelligent Control and Computer Engineering, Lecture Notes in Electrical Engineering, Vol. 70, Springer, pp. 83-93, 2011.
https://doi.org/10.1007/978-94-007-0286-8_8.
- [20] Yudong Zhang, Shuihua Wanga, Preetha Phillips, Genlin Ji. *Binary PSO with mutation operator for feature selection using decision tree applied to spam detection*, Knowledge-Based Systems, Vol: 64, pp. 22–31,2014.
<https://doi.org/10.1016/j.knosys.2014.03.015>.
- [21] Mark A. Hall , *Correlation-based Feature Selection for Machine Learning*, PhD Thesis, The University of Waikato, 1999.
- [22] Irene Rodriguez Lujan, Ramon Huerta, Charles Elkan, Carlos Santa Cruz. *Quadratic Programming Feature Selection*, Journal of Machine Learning Research, 11, pp. 1491-1516, 2010.
- [23] Jasmina Novaković, Perica Strbac, Dusan Bulatović. *Toward Optimal Feature Selection Using Ranking Methods And Classification Algorithms*, Yugoslav Journal of Operations Research, Vol: 21, Number 1, pp. 119-135, 2011.
<https://doi.org/10.2298/yjor1101119n>.
- [24] Gavin Brown, Adam Pocock, Ming-Jie Zhao, Mikel Luján. *Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection*, Journal of Machine Learning Research, Vol: 13, pp. 27-66, 2012.
- [25] Riyaz Sikora, Selwyn Piramuthu . *Framework For Efficient Feature Selection In Genetic Algorithm Based Data Mining*, European Journal of Operational Research, Vol:180, pp. 723–737, 2007.
<https://doi.org/10.1016/j.ejor.2006.02.040>.
- [26] Shital C. Shah, Andrew Kusiak. *Data mining and genetic algorithm based gene/SNP selection*, Artificial Intelligence in Medicine, Vol: 31, pp. 183–196, 2004.
<https://doi.org/10.1016/j.artmed.2004.04.002>.
- [27] Utpal Kumar Sikdar, Asif Ekbal, Sriparna Saha. *Differential Evolution based Feature Selection and Classifier Ensemble for Named Entity Recognition*, Proceedings of COLING 2012: Technical Papers. COLING 2012, Mumbai, December, pp. 2475–2490, 2012.
<https://doi.org/10.1007/s10032-011-0155-7>.
- [28] Yuanning Liu, Gang Wang, Huiling Chen, Hao Dong, Xiaodong Zhu, Sujing Wang . *An Improved Particle Swarm Optimization for Feature Selection*, Journal of Bionic Engineering, Vol: 8, 2011.
[https://doi.org/10.1016/s1672-6529\(11\)60020-6](https://doi.org/10.1016/s1672-6529(11)60020-6).
- [29] Bing Xue, Mengjia Zhang, Will N. Browne. *Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach*, IEEE Transactions on Cybernetics, Vol. 43, No. 6, 2013.
<https://doi.org/10.1109/tsmcb.2012.2227469>.

- [30] Jianjun Yu, Jindan Yu, Arpit A. Almal, Saravana M. Dhanasekaran, Debashis Ghosh, William P. Worzel, Arul M. Chinnaiyan. *Feature Selection and Molecular Classification of Cancer Using Genetic Programming*, Neoplasia, Vol. 9, No:4, pp. 292 – 303, 2007.
<https://doi.org/10.1593/neo.07121>.
- [31] Jacques-Andre Landry, Luis Da Costa and Thomas Bernier. *Discriminant Feature Selection by Genetic Programming: Towards a domain independent multi-class object detection system*, Systemics Cybernetics and Informatics, Vol: 3(1), pp. 76-81, 2006.
- [32] Omer Abu-Arquab, Zaer Abo-Hammour, Shaher Mohammad Momani. *Application of Continuous Genetic Algorithm for Nonlinear System of Second-Order Boundary Value Problems*, Applied Mathematics & Information Sciences, 8, No.1, pp. 235-248, 2014.
<https://doi.org/10.12785/amis/080129>.
- [33] Omer Abu-Arquab, Zaer Abo-Hammour. *Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm*, Information Sciences, Vol: 279, pp. 396-415, 2014.
<https://doi.org/10.1016/j.ins.2014.03.128>.
- [34] Riccardo Poli, William B. Langdon, Nicholas F. McPhee, John R. Koza, *A Field Guide to Genetic Programming*, 2016.
<http://cswww.essex.ac.uk/staff/rpoli/gp-field-guide/>.
- [35] Zhaohui Gan, Tommy W. Chow, W.N. Chau. *Clone selection programming and its application to symbolic regression*, Expert Systems with Applications, Vol: 36, 2009, pp. 3996–4005, 2009.
<https://doi.org/10.1016/j.eswa.2008.02.030>.
- [36] Hajira Jabeen, Abdul Rauf Baig. *Review of Classification Using Genetic Programming*, International Journal of Engineering Science and Technology, Vol: 2, pp. 94-103, 2010.
- [37] Beyza Gorkemli. *Study of Artificial Bee Colony Programming (ABCP) to Symbolic Regression Problems*, PhD Thesis, Erciyes University, Engineering Faculty, Computer Engineering Department, 2015.
- [38] Vladimir Cherkassky, Don Gehring, Filip Mulier. *Comparison of adaptive methods for function estimation from samples*, IEEE Transactions on Neural Networks, Vol: 7 (4), 1996, pp. 969- 984, 1996.
<https://doi.org/10.1109/72.508939>
- [39] Dominic P. Searson, *GPTIPS 2: an open-source software platform for symbolic data mining*. Chapter 22 in Handbook of Genetic Programming Applications, A.H. Gandomi et al., (Eds.), Springer, New York, NY, 2015., <https://sites.google.com/site/gptips4matlab/file-cabinet>.
https://doi.org/10.1007/978-3-319-20883-1_22
- [40] UCI, *Machine Learning Repository*, Concrete Compressive Strength Data Set. <https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>, Access Date: 15.10.2016.
- [41] Sibel Arslan, Celal Ozturk, *Multi Hive Artificial Bee Colony Programming for high dimensional symbolic regression with feature selection*, Applied Soft Computing Journal 78, 515–527, 2019.
<https://doi.org/10.1016/j.asoc.2019.03.014>.
- [42] Sibel Arslan, Celal Ozturk, *Artificial Bee Colony Programming Descriptor for Multi-Class Texture Classification*, Applied Sciences, 9(9), 2019.
<https://doi.org/10.3390/app9091930>.

