



---

# VERSATILE FUNCTION IN GPA

*T. Brandejsky\**

---

**Abstract:** The paper, devoted to continuous versatile function application in the Genetic Programming Algorithm (GPA), begins with a discussion of similarities between GPA with versatile function and neural network. Then, the function set influence on GPA efficiency is discussed. In the next part, there is described a hybrid evolutionary algorithm that combines GPA for structure development and Evolutionary Strategy (ES) for parameters and constant optimization; which is herein much more significant than in the standard GPA. There is also discussed the setting of parameters of this hybrid algorithm and due to a different function set. The original idea of a versatile function, which origins come from the area of fuzzy control systems, is formulated and explained. Four different implementations of this versatile function are discussed. On the base of experiments with the hybrid evolutionary algorithm providing symbolic regression of precomputed Lorenz attractor system data representing its dynamic behaviour; the comparison of three variants of versatile functions was formulated. The paper also presents ways how to set up hybrid evolutionary algorithm parameters like population sizes as well as limits of maximal population numbers for both algorithms: GPA for structural development and nested ES for parameters optimization. The versatile function concept is applicable but it requires the hybrid evolutionary algorithm use as it is explained in the paper.

Key words: *hybrid evolutionary algorithm, genetic programming, versatile function, neural network*

Received: July 13, 2020

DOI: 10.14311/NNW.2020.30.025

Revised and accepted: December 30, 2020

## 1. Introduction

J. Koza formed Genetic Programming as an optimization problem in [7] extending previous research of Michael Lynn Cramer [3]. GPA in this task optimizes structure and parameters of an arising model on the base of information in a training data set or fitness function. The name Genetic Programming is given by the first idea to develop evolutionary computer programs. One of the first applications published by J. Koza 1994 [8] was symbolic regression (discovering of model described by algebraic equation fitting training data). Later, many different application domains were opened such as analogue electric circuit design [9], synthesis of topology for

---

\*Tomas Brandejsky; Faculty of Electrical Engineering and Cybernetics of the UPce, Namesti CS Legii 565, 530 02 Pardubice, Czech Republic, E-mail: [tomas.brandejsky@upce.cz](mailto:tomas.brandejsky@upce.cz)

controller [10], application of GP to the synthesis of complex kinematic mechanisms [11] and many others.

The paper starts with chapter one containing discussion of a similarity between Neural Network (NN) and GPA with versatile function concepts and function set influence on GPA efficiency and landscape space geometry. Then it continues with the discussion of the study motivation in chapter 2. In the third chapter, the versatile function is described. The fourth chapter is devoted to a description of the GPAes algorithm used for tests of the versatile function concept including a discussion of four versions of the Evolutionary Strategy (ES) optimizer. Then the chapter describing experiments (and their results) is present. The 6th chapter deals with result discussion and the 7th one with the conclusion.

### 1.1 Neural network and GPA with versatile function similarities

Naturally, the use of only versatile function in equations developed by the GPA tends to many homogeneous structures than the use of a typical set of many functions containing e.g. “plus”, “minus”, “sine”, “cosine” and many other ones. In the symbolic regression application this structure has the form of tree and it resembles multiple input, single output layered neural network.

This is not the only similarity. Learning of such structure much more concentrates to the fitting of parameters than to the development of the structure. This is inverse situation than e.g. in symbolic regression problem solving by GPA with above mentioned typical function set, as it will be discussed below. Also, the number of versatile functions is smaller than the number of typical GPA building function ones. Typically, the only versatile function is used. This fact tends to increased requirements to parameter optimization which extends beyond capabilities of typical GPA. Thus, the application of hybrid GPA is felicitous. Until now, such concept was not studied by information accessible to the author.

The above-described situation also changes the learning mechanism, where the parameter identification plays a more significant role than in the standard symbolic regression. It also underlines the sense of hybrid evolutionary algorithm application. The effect of a significantly larger number of ES (parameter optimizer) cycles than the GPA ones is demonstrated in the Experiments chapter. In the future, alternative optimizers known in the deep learning area will be tested in the place of ES parameters optimizer. If the GPA function is limited to random generating of the initial population (or even the single individual), the random neural network with neurons described by versatile function is learned by ES algorithm.

### 1.2 Influence of used function set

Many parameters and design choices are influencing resulting GPA behaviours and properties. Requirements on them vary in dependency on the application domain. In the first place, it is possible to mention the influence of terminal and function set. This set must contain all fundamental terminals and functions needed for solving the actual problem. There are two contradictory requirements. Terminal and function sets should be minimalistic. If it is complicated, the probability of

occurrence of the needed function is small and it decreases the efficiency of the GPA. On the opposite side, if some useful function is missing it can be replaced by any complicated equivalent of the Taylor series, or the GPA will end with a large error of fitness function. It is also useful if the function set is free of mutually interchangeable operators like twin  $\sin()$  and  $\cos()$  functions. The presence of both is not needed (one is enough) and decreases the efficiency of GPA which tends to swap them repeatedly without any positive effect on the fitness error.

## 2. Motivation

GPA works on discontinuous and non-linear space of functions. The use of versatile function allows us to change the space of functions to a continuous space of parameters. Thus, at the beginning of this work was the expectation that the GP task will be easier if the problem domain is continuous.

The idea of a fuzzy gamma operator started at the beginning of this work. It was introduced by Constantin von Altrock [1] to combine properties of fuzzy logic AND operator and OR operator.

Genetic Programming was developed on the base of the Genetic Algorithm with the idea, that programming is an optimization of randomly generated instructions and parameters. State-space of GPA is discrete from the viewpoint of change of operators like addition, subtraction or substitution during mutation and crossover operations. Discrete spaces disqualify some optimization methods and complicate the work of the remaining ones, as was discussed above. A possible way suggested in this work is the use of only one kind of versatile function capable of continuously changing its property depending on the magnitude of its control parameter represented by real numbers. A versatile continuous function for GPA presented in this work is based on the weighted average of particular function results. The number of control parameters of these functions is not limited. In this work, the structure of the solution created from versatile functions is developed by GPA and parameters set up by Evolutionary Strategy within the frame of the hybrid algorithm.

The final aim of this work will be a limitation of evolution on the level structure development and transformation of the whole symbolic regression problem to the continuous optimization of a parameter set. Thus, it is the transformation of the structure development problem onto the problem of estimation of the versatile function parameters. These versatile functions shall have a lot of parameters depending on their dimension – the number of simple binary functions they combine into a final versatile one.

## 3. Versatile function concept

The term versatile function denotes in this paper a binary function combining properties of several functions. Depending on parameter magnitudes, properties of a versatile function are closer to one or another original function as in the case of von Altrock's fuzzy gamma operator. The main differences are in the combination of any number of original functions and in the style of this combination, which is not only proportional like in version 1, but can take the form of linear interpolation in

versions 2, 3 and 4. During the work on the first application of the above-described ideas, there were developed three different versions of a versatile function suitable for the solution of the Lorenz attractor equation symbolic regression problem. As well as for any other symbolic regression problem, there exists an optimal set of functions containing all needed functions and no other. In the case of Lorenz attractor equations, this set contains “+”, “-” and “\*” functions (6). During the work, the following three different designs of a versatile function were formulated.

### 3.1 Version 1

The first version of versatile function was very close to the original von Altrock’s gamma operator, especially in the two-dimensional (two functions) case, see (1).

$$r = \lambda_1 \lambda_2 f_1(x_1, x_2) + (1 - \lambda_1) f_2(x_1, x_2) + (1 - \lambda_2) f_3(x_1, x_2) \quad (1)$$

The Eq. (2) presents a combination of functions “+”, “-” and “\*” requiring two independent parameters  $\lambda_1$  and  $\lambda_2$ .

$$r = \lambda_1 \lambda_2 (x_1 + x_2) + (1 - \lambda_1)(x_1 - x_2) + (1 - \lambda_2) x_1 x_2 \quad (2)$$

Experiments with the first version were not very satisfactory. Especially, the evolutionary process was very slow. The parameter magnitudes were going to proper values for a long time, as will be presented in the experiment section. This situation was caused by the two main problem points of this concept: The first of them is the composite influence of two independent versatile function parameters  $\lambda_1$  and  $\lambda_2$  onto particular functions “+” and “-”. The second one is the impossibility to express relations as  $a(x + y) + b(xy)$  directly, where  $a \neq (1 - b)$ . In this concept, additional functions (multiplication or even versatile function and additional constants) are necessary. It complicates resulting structures and makes evolution slower.

### 3.2 Version 2

The next version of versatile function uses linear interpolation of three binary functions  $f_1(x_1, x_2)$ ,  $f_2(x_1, x_2)$ ,  $f_3(x_1, x_2)$  in the form (3). Thus, it is hard to optimize  $\lambda$  parameters, but it is faster to evaluate than previous version 1.

$$r = \lambda_1 f_1(x_1, x_2) + \lambda_2 f_2(x_1, x_2) + \lambda_3 f_3(x_1, x_2) \quad (3)$$

The version for functions “+”, “-” and “\*” for Lorenz attractor system symbolic regression has the form (4).

$$r = \lambda_1(p_1 + p_2) + \lambda_2(p_1 - p_2) + \lambda_3 p_1 p_2 \quad (4)$$

Also, the related experiment description and comparison is described in the chapter 5.

### 3.3 Version 3

This version is derived from version 2. The interpolation between operators “+” and “−” is replaced by the linear combination of parameters. It can make computation more efficient than in the previous version and cover a wide set of resulting functions.

$$r = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 (p_1 p_2) \quad (5)$$

## 4. Used GPA algorithm

To test this kind of building function for GPA, the problem of symbolic regression was chosen for simple measurement of significant solution quality attributes like computing time, the number of needed evolutionary steps or final fitness error. The real misunderstanding can be caused by two different definitions of the Symbolic Regression (SR) problem.

While some authors understand SR as the way to find any description of training data set, other ones add requirements on precision of the model or its comparability to a solution produced by humans. Such approaches [6] are more computationally expensive but opens new application domains. To solve this Highly Accurate SR (HASR) problems, it is sometimes efficient to use hybrid evolutionary algorithms described in the next paragraph. In this work, the second approach to SR (and concluding HASR) is applied.

### 4.1 Hybrid evolutionary algorithms in symbolic regression

Koza in his work [8] identified a weak point of the Genetic Programming application when applied to symbolic regression in the identification of parameter (constant) magnitudes. Till now, there were presented many modifications of GPA extending its abilities by linear or non-linear optimization techniques such as [12] or [4]. Also, the work [13] concludes with the recommendation to limit linear optimization to few steps for its computing capacity requirements. These optimization techniques can be replaced by a genetic or evolutionary algorithm with similar results – the consumption of computing resources of a hybrid algorithm with optimization of solution parameters is significant. The work [5] brings comparison of different constant optimization methods influencing hybrid evolutionary algorithm efficiency.

### 4.2 Used hybrid GP Algorithm with Evolutionary Strategy for parameters optimization

A used GPAes algorithm [2] consists of a standard GPA without automatically defined functions, discussed in [8]. In each evolutionary cycle, the Evolutionary Strategy is applied to optimize the parameters of each individual in the population. The structure of the algorithm is outlined in Fig. 1.

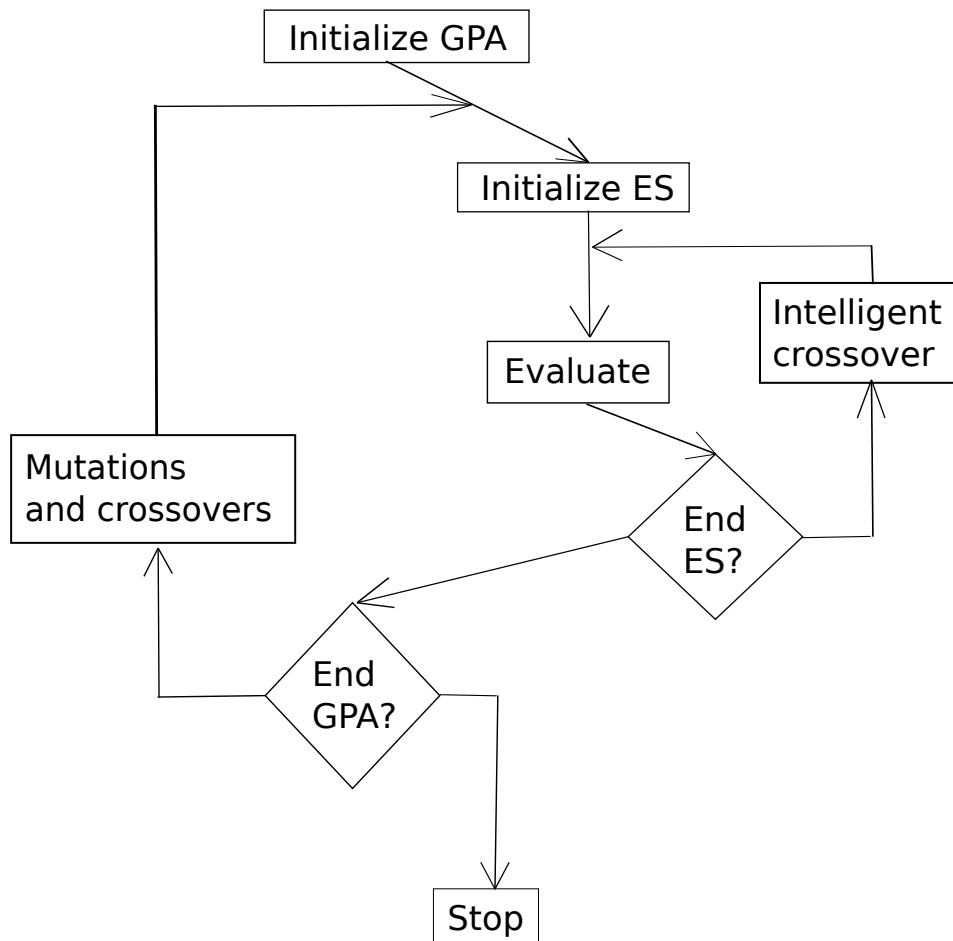


Fig. 1 Structure of used hybrid GP algorithm.

#### 4.3 The different setting of hybrid genetic programming algorithm for versatile function

It is legitimate to expect that a versatile function will ask different behaviour of a hybrid algorithm because it has a lot of parameters to be optimized in non-linear continuous space. On the opposite side, there is smaller pressure to the structure development due to the presence of only one versatile function and two terminals. Thus, it is possible to expect, that it will be useful to run the parameter optimization part of the hybrid algorithm with more iteration cycles than for standard GP function set. Thus, the following experiments will include more different parameter optimization ES cycle limit magnitudes than in the case of the standard function set.

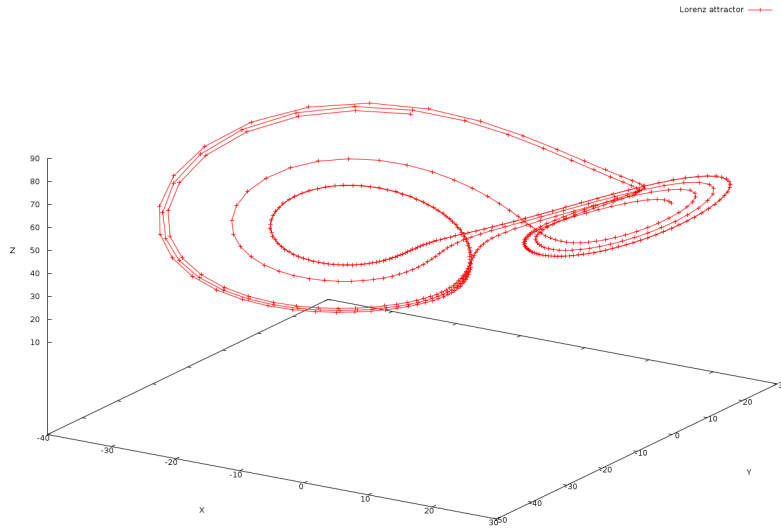


Fig. 2 Lorenz attractor used as a test case.

## 5. Experiments

As it was mentioned above, the Lorenz attractor system served for the first testing of a versatile function concept. This system produces chaotic behaviours for some parameters and it means it is sensitive to errors in the estimation of the model. The equations describing this model are as follows:

$$\begin{aligned}
 x'(t) &= \sigma(y(t) - x(t)) \\
 y'(t) &= x(t)(r - z(t)) - y(t) \\
 z'(t) &= x(t)y(t) - bz(t)
 \end{aligned}
 \tag{6}$$

Used parameters had magnitudes  $\sigma = 16.0$ ;  $r = 45.92$ ;  $b = 4.0$  and initial point has position  $\{x, y, z\} = \{19, 20, 50\}$ .

As was described above in Chapter 3, the testing consists of four different experiments with four minimalistic function sets. They consist of one version of a versatile function and also two terminals representing a constant or variable. Constants and variables were not added into the versatile function, because they are unary, not binary – but such construction is also possible in the future applications of the versatile function concept. Three series of experiments for different function sets were executed. Each set contained constant and variable terminals and a binary versatile function of proper version 1 to 3. A variable training data vector contains 500 samples. The magnitude of constants is a result of optimization by a nested evolutionary strategy algorithm. There were three different tests with the parameters represented by the Tab. I.

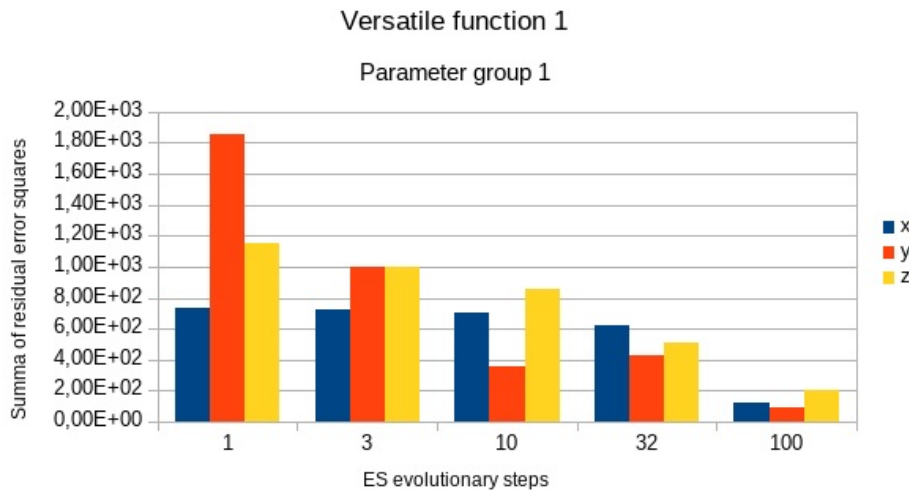
Parameter	Parameter group 1	Parameter group 2	Parameter group 3
GPA population size	100	100	100
GPA populations limit	10	20	20
ES population size	40	40	40

**Tab. I** Versatile function test parameters.

The ES cycle limit was tested for numbers of [1, 4, 10, 40, 100] cycles. Each experiment was repeated 10 times for different PRNG seed magnitudes from the range (1,10) to eliminate random influences. Results are summarised for each versatile function version in the following proper chapters.

### 5.1 Experiments with versatile function version1

The first group of experiments was computed for function set 1. The sums of residual error squares for all three variables x, y, and z of Lorenz attractor obtained during symbolic regression for different hybrid evolutionary algorithm settings are displayed in Fig. 3 to 5.



**Fig. 3** Lorenz attractor regression sum of residual error squares for versatile function version 1 and parameter group 1.

### 5.2 Experiments with versatile function version 2

The second set of experiments tested versatile function implemented as linear combination of addition, subtraction and multiplication. It eliminates some weak points of the previous one.



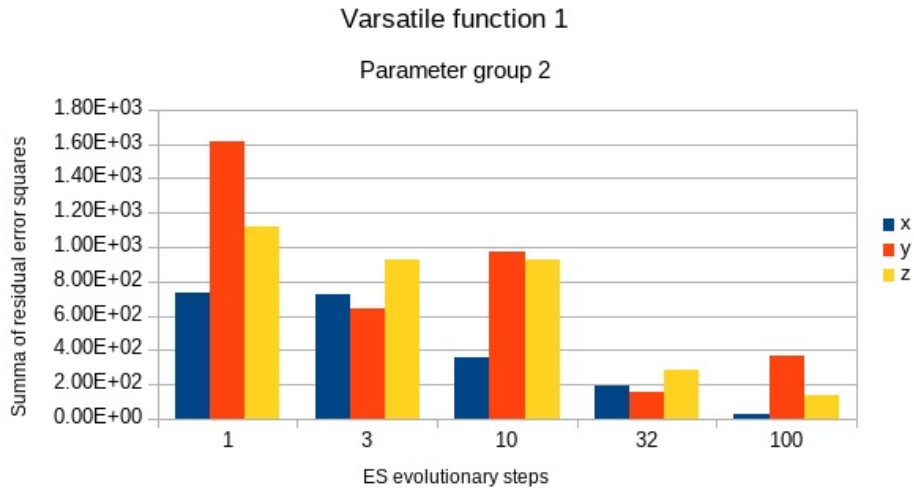


Fig. 4 Lorenz attractor regression sum of residual error squares for versatile function version 1 and parameter group 2.

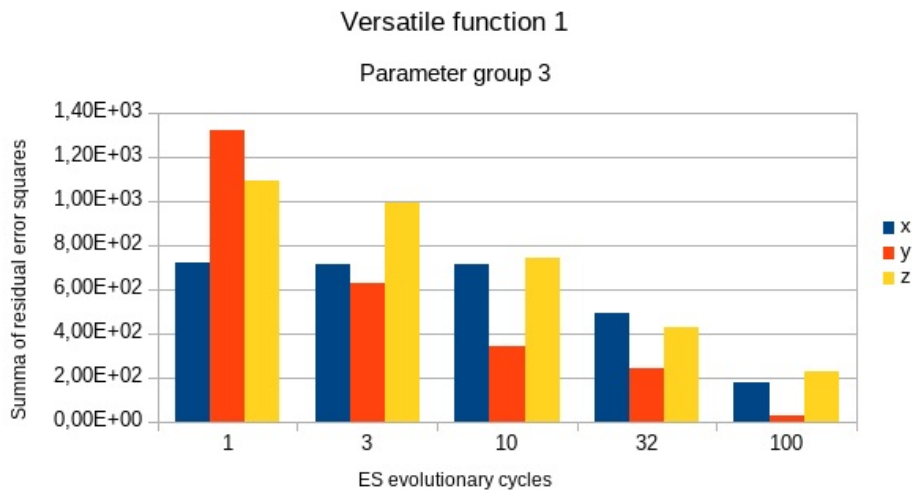
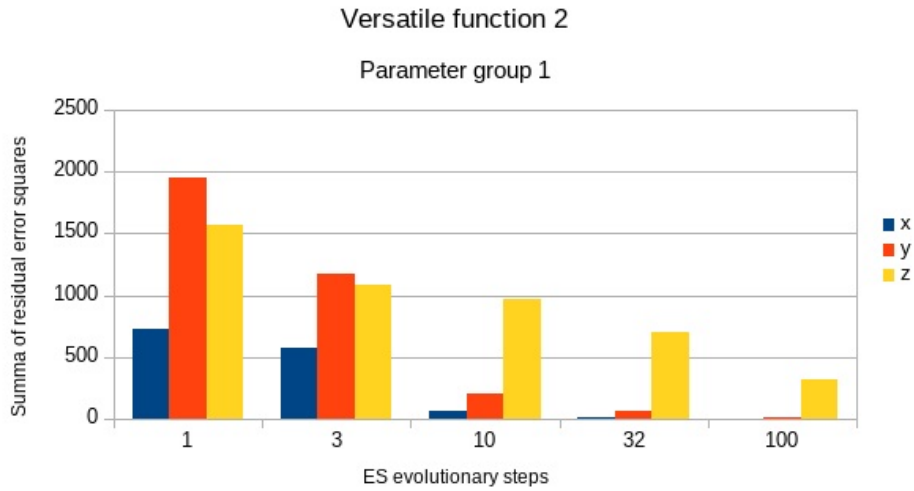


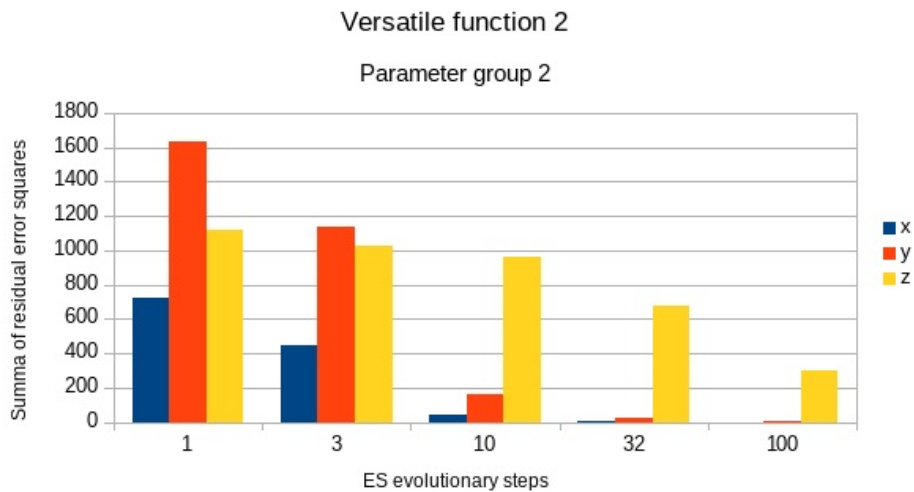
Fig. 5 Lorenz attractor regression sum of residual error squares for versatile function version 1 and parameter group 3.

### 5.3 Experiments with versatile operator version 3

The third set of experiments replaces “+” and “-” functions by multiplication of particular arguments by a related constant. It gives bigger expressivity and allows the description of a wide set of expressions, but it also needs three parameters.

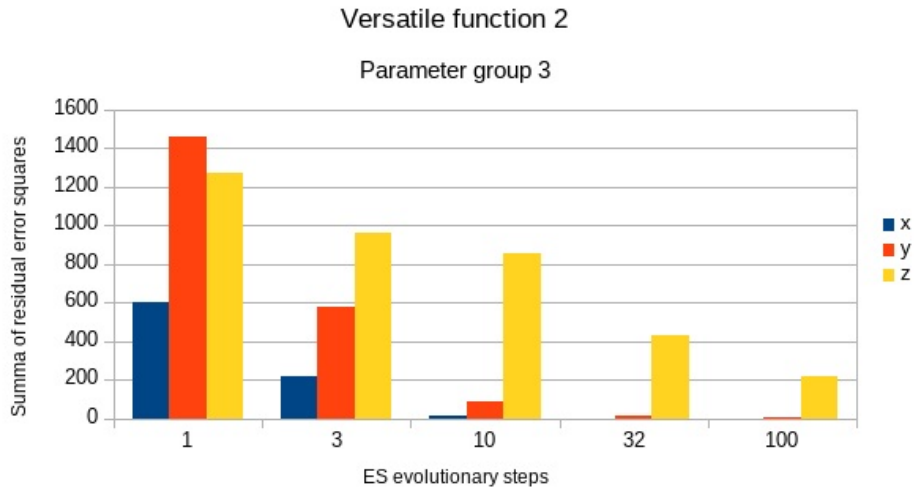


**Fig. 6** Lorenz attractor regression sum of residual error squares for versatile function version 2 and parameter group 1.

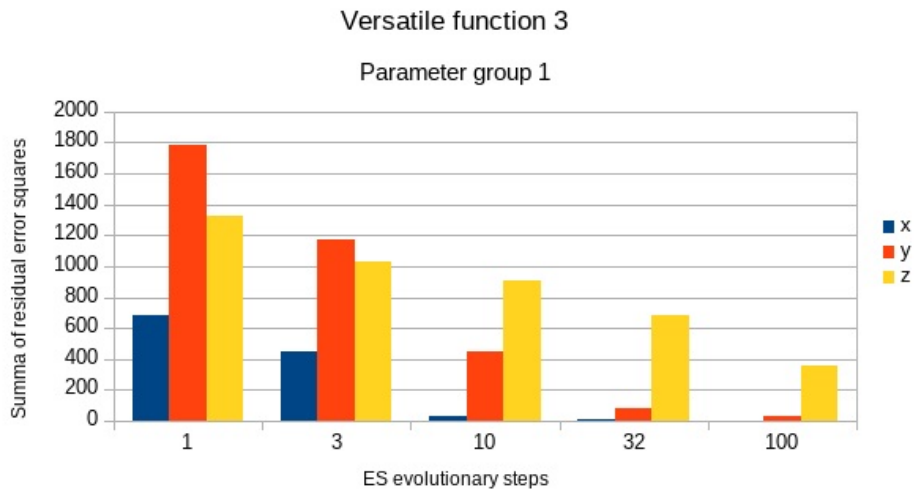


**Fig. 7** Lorenz attractor regression sum of residual error squares for versatile function version 2 and parameter group 2.

Thus, it is hard to optimize them. E.g., the best solution for  $x$  variable gives solution  $x'(t) = -16.2983 * x(t - 1) + 16.386 * y(t - 1) + 0.00526773 * x(t - 1) * y(t - 1)$  with sum of square errors for 500 samples equal to 0.631018 on the place of precise solution  $x'(t) = 16 * (y(t - 1) - x(t - 1))$ .



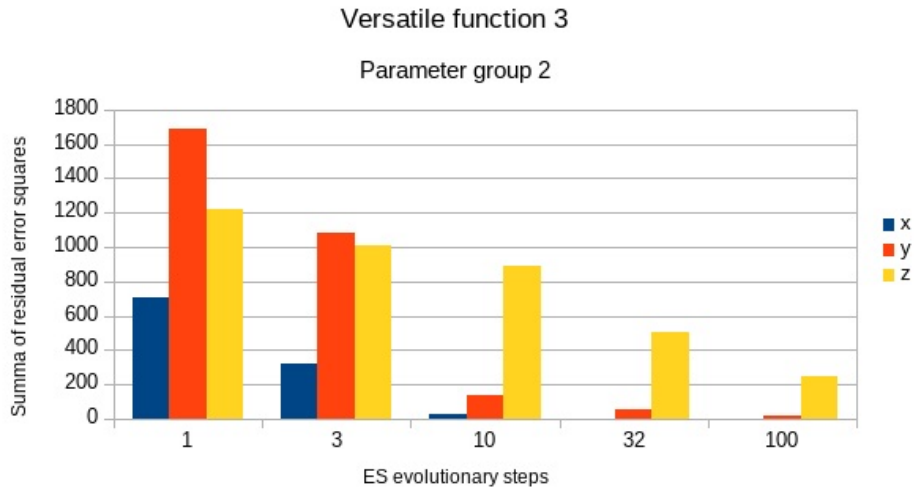
**Fig. 8** Lorenz attractor regression sum of residual error squares for versatile function version 2 and parameter group 3.



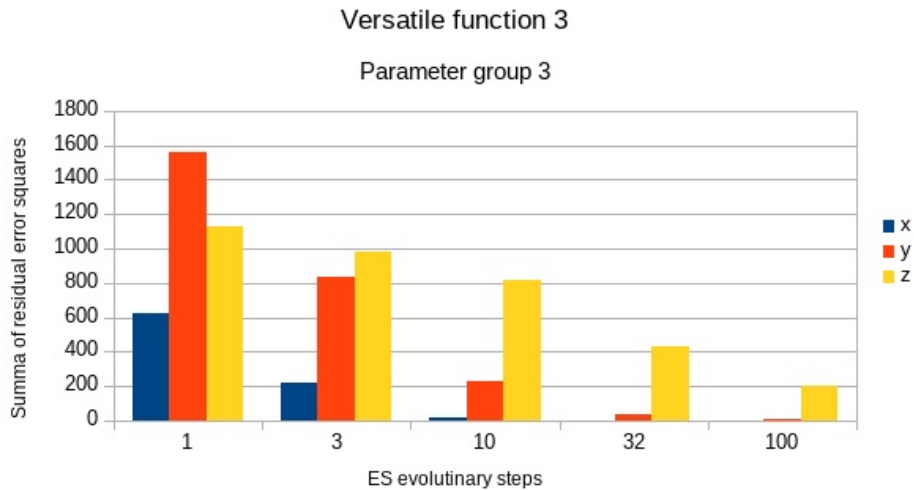
**Fig. 9** Lorenz attractor regression sum of residual error squares for versatile function version 3 and parameter group 1.

## 6. Discussion

The number of fitness function evaluations in the above-presented graphs was the function of the number of evolutionary cycles and individuals in the population. The above-presented results of experiments point out that the number of fitness



**Fig. 10** *Lorenz attractor regression sum of residual error squares for versatile function version 3 and parameter group 2.*



**Fig. 11** *Lorenz attractor regression sum of residual error squares for versatile function version 3 and parameter group 3.*

function evaluations influences the resulting remaining error much more for ES than for GPA components of the hybrid evolutionary algorithm. Such observation agrees with above-presented expectation, that the versatile function will require a different proportion of GPA and ES cycles than for a typical set of functions used in the standard GPA or HGPA. While the best efficiency for a typical function set

occurs for 2 or 3 constant optimization cycles, a versatile function requires tens or even hundreds of evolutionary cycles for its complicated structure. Similarly, it is sufficient to use tens of GPA cycles, and a bigger number does not bring improvements justifying the increase of computing resources use.

In the presented 3 groups of experiments with different versatile function implementations, there is no significant difference between them. Only the versatile function version 1 was slightly worse in the presented experiments, but on the base of only one test case – symbolic regression of the Lorenz attractor from the pre-computed data set. On the opposite side, versatile function version 1 is the best for small numbers of iteration cycles, but for higher precision of solutions, it loses its efficiency.

The application of a versatile function requires a different proportion between GPA and ES cycle limits. While for a typical function set containing many different functions, the number of ES cycles limit between 2 and 50 is enough (and the GPA cycle limit is between hundreds and many thousands). In the case of versatile function application, the situation is the opposite.

## 7. Conclusion

The presented paper discusses the idea of a versatile function application in genetic programming. It identifies a requirement for use of a hybrid genetic evolutionary algorithm combining a genetic programming algorithm for structural development and an evolutionary strategy algorithm for tuning of constants. These requirements are observed because some of the versatile function parameter changes are equivalent to structure modification in standard genetic programming. The evolutionary strategy used for tuning of solution equation parameters can be replaced by optimization techniques known from Artificial Neural Networks learning due to the only versatile function presence. This idea will be tested in future works. The application of a versatile function is demonstrated on the Lorenz attractor system symbolic regression.

## Acknowledgement

The work was supported from ERDF/ESF “Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)” (No. CZ.02.1.01/0.0/0.0/17\_049/0008394).

## References

- [1] ALTROCK C. v. Uber den Daumen gepeilt. *Computer & Technik*, 3, pp. 78–92, 1991.
- [2] BRANDEJSKY T. Nonlinear system identification by GPA-ES. In: *Proceedings of the 2012 13th International Carpathian Control Conference, ICCO 2012*, pp. 58–62, 2012, doi: [10.1109/CarpathianCC.2012.6228616](https://doi.org/10.1109/CarpathianCC.2012.6228616).
- [3] CRAMER N.L. A representation for the Adaptive Generation of Simple Sequential Programs. In: Grefenstette, J. J. (ed) *Proceedings of an International Conference on Genetic Algorithms and the Applications*, Carnegie-Mellon University, Pittsburgh, PA, USA, 1985, pp. 183–187.

- [4] FROHLICH J., HAFNER C. Extended and Generalized Genetic Programming for function Analysis. submitted to the Journal of Evolutionary Computation, 1996.
- [5] HLAVAC, V. Genetic programming with either stochastic or deterministic constant evaluation. Neural Network World. 2018, 2, pp. 119–131. ISSN 1210-0552. doi: [10.14311/nnw.2018.28.006](https://doi.org/10.14311/nnw.2018.28.006).
- [6] KORNS M. Extreme Accuracy in Symbolic Regression. In: Riolo R., Moore, J. H., Kotanchek M. (eds) Genetic Programming Theory and Practice XI (Genetic and Evolutionary Computation). pp. 1–30. Springer, New York, 2014, ISBN-13: 978-1493903740, doi: [0.1007/978-1-4939-0375-7-1](https://doi.org/10.1007/978-1-4939-0375-7-1).
- [7] KOZA J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, Massachusetts, 1992, ISBN-13: 978-0262111706.
- [8] KOZA J.R. Genetic Programming II: Automatic Discovery of Reusable Programs. The MIT Press, Cambridge, MA, 1994.
- [9] KOZA J.R., ANDRE D., BENNET III F.H., KEANE M. Genetic Programming 3: Darwinian Invention and Problem Solving. Morgan Kaufman, 1999, ISBN 1-55860-543-6, doi: [10.1109/TEVC.1999.788530](https://doi.org/10.1109/TEVC.1999.788530).
- [10] KOZA J.R., KEANE M.A., STREETER M.J., MYDLOWEC W., YU J., LANZA G. Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers, 2003, ISBN 1-4020-7446-8.
- [11] LIPSON H. How to draw a straight line using a GP: Benchmarking evolutionary design against 19th century kinematic synthesis. In: Keijzer, M. (ed) Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference, International Society for Genetic and Evolutionary Computation, CDROM, 2004.
- [12] MCKAY B., WILLIS M.J., BARTON G.W. Using a Tree Structured Genetic Algorithm to Perform Symbolic Regression. In: Proc. of the 1st Int. Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications. UK, 1995, pp. 487–492, doi: [10.1049/cp:19951096](https://doi.org/10.1049/cp:19951096).
- [13] RAIDL G.R.: A Hybrid GP Approach for Numerically Robust Symbolic Regression, 1998.