

INFLUENCE OF EXPERIMENT DESIGN IN GPA INVESTIGATING WITH RESPECT TO PRNGS

Tomas Brandejsky

University of Pardubice

Department of Software Technologies, Faculty of Electrical Engineering and Informatics
CS Legies Square 565, Pardubice
Czech Republic
Tomas.Brandejsky@upce.cz

Abstract: This paper analyses the influence of experiment parameters onto the reliability of experiments with genetic programming algorithms. The paper is focused on the required number of experiments and especially on the influence of parallel execution which affect not only the order of thread execution but also behaviors of pseudo random number generators, which frequently do not respect recommendation of C++11 standard and are not implemented as thread safe. The observations and the effect of the suggested improvements are demonstrated on results of 720,000 experiments.

Keywords: Genetic Programming Algorithm, Efficiency, Pseudo Random Number Generator, Experiment Repeatability, Number of Experiments, Experiment Results Reliability

1 Introduction

Evolutionary Algorithms (EAs) in this moment represent an extremely powerful tool suitable to many areas of optimization, design, modelling, etc. There exist many kinds of these algorithms and much more specific algorithms and their implementations. During the research of Genetic Programming Algorithms (GPAs) (a rich analysis is summarized by [1]), a kind of EAs suitable especially to constructive and modelling problems, the need to eliminate external influence to results of executed experiments was recognized.

It is a commonly known fact that Evolutionary Algorithms like Genetic Algorithms, Evolutionary Strategies (ESs), Genetic Programming Algorithms (GPAs), etc. are stochastic or influenced by Pseudo-Random Number Generators (PRNGs) or another sources of entropy, respectively. These entropy sources are applied in many phases of GPA such as the generation of the initial population, evolutionary operators (especially mutation and cross-over), etc. On the opposite side, there were published works about evolutionary algorithms working without this entropy source replacing it by deterministic chaos system equations or even by an aperiodic function [2-4].

In the paper [5] the influence of PRNGs onto GPA efficiency was analyzed. This influence of PRNG onto the resulting efficiency of the algorithm was unexpectedly small and unclear. Thus, this paper is focused onto the analysis of next influences affecting the GPA function, which might impact experiment results and decrease their quality.

During studies of EAs behavior (e.g. the influence of PRNGs) it is needed to eliminate other known phenomena which are capable to mask the studied behavior. There exist three basic parameters representing the efficiency of EA – computational time, number of evaluations of the fitness function and number of evolutionary cycles. Computational time gives the worst correlation with EA quality because it also depends on the used supporting algorithms. For example, the computational complexity of some RNGs exceed more than 10 times the computational complexity of the simplest ones like the standard C/C++ rand() function.

Such experiment problems are easily identifiable, and their elimination also is not complicated. While experimenting with parallel versions of EAs, the problem of experiment repeatability was observed. Even the same experiments with the same PRNG and constant initial seed magnitude give different results (i.e. they are not repeatable). This situation occurs not because of the instability of the studied algorithm (or even due to its chaotic properties) but it is caused by external influences, especially by task switching which depends on other processes in the computer, including e.g. network communication, see the following Table 1. There are numbers of needed iteration for achieving the required precision (number of square roots) of Lorenz system symbolic regression problem described in the following text for the same initial PRNG seed magnitude.

Table 1: Result of simple test of symbolic regression (numbers represent needed iteration cycles to achieve required residual error for the same initial state including PRNGs seed magnitude)

x	y	z
7	28	47
8	124	84
45	122	22
10	221	53
6	179	51
26	229	24
4	276	41
3	160	61
5	139	29
30	427	15

The above described observation implies a significant question on how many experiments are needed to eliminate the influence of random perturbations of PRNG generated number series and, thus, to eliminate perturbations of computed experiment series? This question is caused by the non-stationarity of short number series generated by PRNG. Such observations complicate the organization and the evaluation of experiments with EAs focused on the comparison of efficiency of different algorithms, the influence of PRNGs onto the operations of the algorithms, etc.

2 Influences

As it was mentioned above, the main influences complicating the organization, execution and evaluation of the experiments with EAs are

- the influence of other tasks running on the same node of the computing cluster,
- PRNG number series stationarity,
- thread switching (if the used number generator is shared between threads or if threads mutually communicate).

The first point is caused by the fact that other tasks in the operating system influence in the next point described task switching and also onto PRNGs, if they are not implemented as thread safe. The problem is, that each operating system also runs many different “service” tasks like network communication support, cluster operation supports, etc. These tasks cannot be stopped during experiments and they can cause changes of the task run order during experiments. Also, the task scheduler of the used operating system plays its role. Linux systems frequently offer normal, batch, round-robin and FIFO CPU schedulers, or degree of their determinism, respectively.

PRNG generates a vector of pseudo random numbers. These vectors can be often described by a long period function. Thus, on an extremely long vector this function exhibits similar behaviors, like the stationary of the random number series.

In the presented experiments a special kind of GPA, called GPA-ES, is used. This algorithm uses a combination of GPA for structural development and Evolutionary Strategy (ES) algorithm for optimization of each individual parameter. This duality allows to eliminate the influence of the ill-identified parameters. GPA-ES algorithm is described by Algorithm 1.

In the performed experiments with GPA and especially GPA-ES algorithms, the needed number series generated by PRNG, or series of experiments respectively are not small because of the following reasons. Even if the populations are large and, thus, in each cycle of GPA or GPA-ES algorithm a lot of random numbers are required, and the period length of some PRNGs is not big, the needed number of experiments to investigate the state space can be large, because e.g. the number of generated random numbers for GPA might not be a divisor of the length of PRNG cycle and there is also the influence of task switching, which also makes the PRNG period longer. These reasons ask a significant question about how many experiments are needed to achieve reliable results of experiments with GPA algorithms.

The following figure describes the average number of iterations of the Lorenz attractor system symbolic regression for 50 to 1,200 experiments. The modelled system is described in the following text. The results were obtained on a multi-core system with 12 parallel threads without improvements of the experiment organization suggested in the next chapters. The population size was 1,000 individuals.

Algorithm 1: Studied GPA-ES algorithm with parameter preoptimization

```

1: for all individuals do Initialize() end for;
2: for all individuals do Evaluate() → fitness end for;
3: Sort(individuals);
4: if Terminal_condition() then stop end if;
5: for all individuals do
    select Rand() of
        case a do Mutate() → new_individuals;
        case b do Symmetric_crossover() → new_individuals;
        case c do One_point_crossover() → new_individuals;
        case d do Re-gerating() → new_individuals;
    end select;
    end for;
6: for all individuals do
    New ES_algorithm_object
    for all ES_individuals do Initialize() end for;
    evaluate() → ES_fitness
    for all ES_cycles do for all ES_individuals do Evaluate → ES_fitness end for;
        for all ES_individuals do intelligent_crossover() => new_ES_individuals
            Evaluate => new_ES_fitness
        end for;
    end for;
    for all ES_individuals do
        if new_ES_fitness>ES_fitness then
            ES_individual = new_ES_individual;
            fitness = new_fitness;
        end if;
        Sort(ES_individuals, ES_fitness);
    end for;
    end for;
    new_individual=ES_individual[0]; new_fitness = ES_fitness[0];
7: for all individuals do
    if new_fitness<fitness then
        individual = new_individual;
        fitness = new_fitness;
    end if;
    end for;
8: goto 3;

```

Now, looking that there are observable differences between PRNGs on fig. 1, the results are not reliable. After 10,000 experiments we obtain different order of PRNGs and smaller differences between them.

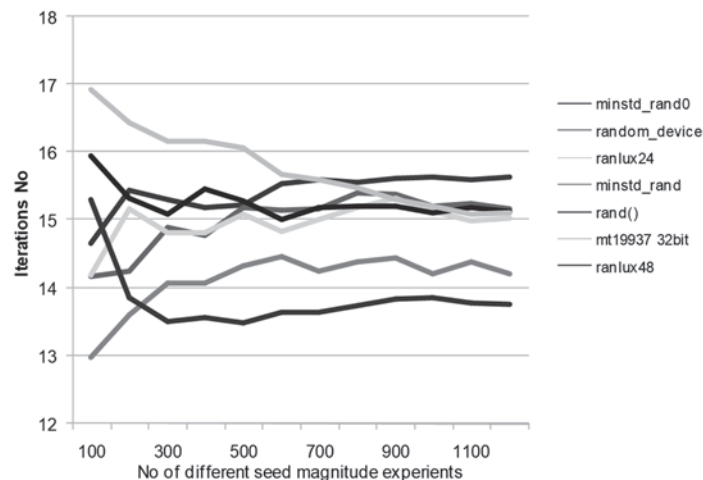


Figure 1: Dependency of number of iterations on number of experiments of Lorenz Y variable symbolic regression

3 Experiment Organization

As in the previous works, Lorenz attractor system equations reconstruction from precomputed data by symbolic regression using GPA-ES algorithm [6] was used. The Lorenz attractor system is described by equations (1) with parameters (2).

$$\begin{aligned} x'(t) &= \sigma(y(t) - x(t)), \\ y'(t) &= x(t)(\rho - z(t)) - y(t), \\ z'(t) &= x(t)y(t) - \beta z(t), \end{aligned} \tag{1}$$

$$\sigma = 16, \quad \beta = 4, \quad \rho = 45.91. \tag{2}$$

In comparison with the previous experiments [5], where sizes of both GPA and ES populations were equal to 1,000 individuals, in these herein presented experiments the size of populations was decreased to 100 individuals to increase number of iterations and thus to increase the resolution of obtained results. The length of the experiment series was increased to 10,000 experiments. The limit of the sum of remaining error squares across the whole 500 data points remains 10^{-7} .

4 Examined Generators

The examined generators included rand() function defined in <cstdlib> of GNU C++ version 4.8.1 implementing the new version 11 of the C++ language standard, a true random number generator that produces non-deterministic random numbers on the base of stochastic processes [7], 32 bit Mersenne Twister 19937 generator, 48 bit and 24bit subtract-with-carry pseudo-random generator with accelerated advancement and multiplicative congruential pseudo-random number generators – ranlux (a type of linear congruential engine) with 2 different parameter groups of the novel <random> library implemented in the new version 11 of the C++ language. This random library is fully object oriented, contains a large number of random number generators and distribution modifiers and it is thread safe.

The increase of the experiment sensitivity can also help us to decide if hierarchical algorithms like GPA-ES are more sensitive to the type of PRNG controlling GPA developing structures or to the type of PRNG controlling ES optimizing parameters of individuals.

The following experiment results present the summary of experiments with symbolic regression of dataset describing movement of the Lorenz attractor system (1) and (2) using GPA-ES algorithm with GPA and ES population sizes of 100 individuals. The first three graphs, Figure 2 to 4, demonstrate the dependencies between PRNGs controlling the GPA part of the algorithm (denoted by upper case letters) and the ES part of the algorithm (lower case letters) and the average number of iteration cycles for variable X, Y and Z.

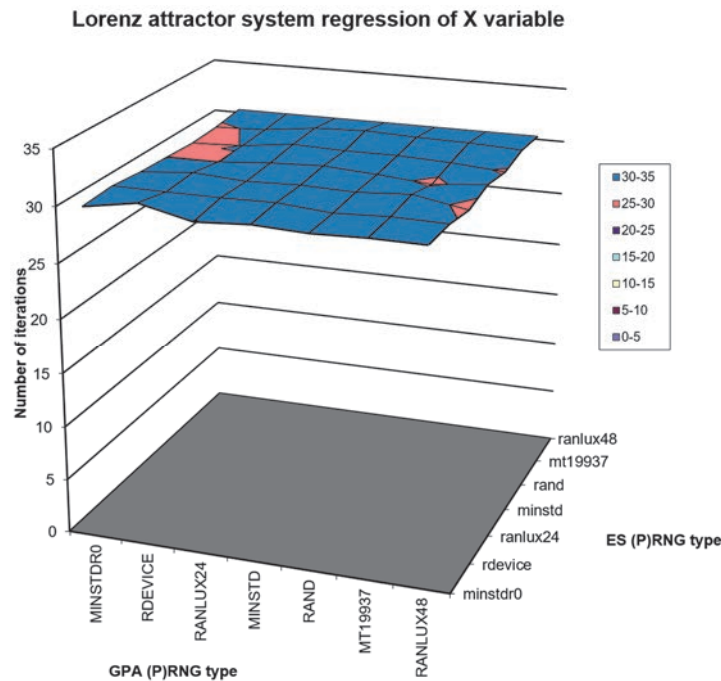


Figure 2: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor X variable symbolic regression using multi-thread version of the algorithm with minimal differences between different PRNGs

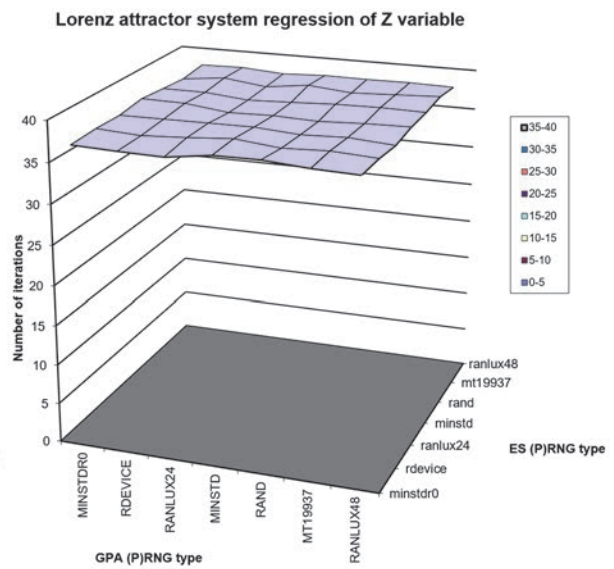
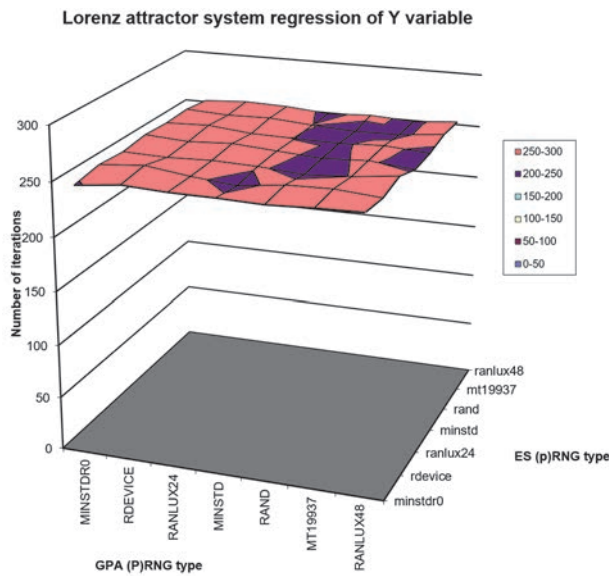


Figure 3: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor Y variable symbolic regression using multi-thread version of the algorithm with minimal differences between different PRNGs

Figure 4: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor Z variable symbolic regression using multi-thread version of the algorithm with minimal differences between different PRNGs

In the case on single-task execution eliminating task switching we obtain different graphs Fig. 5 to 7.

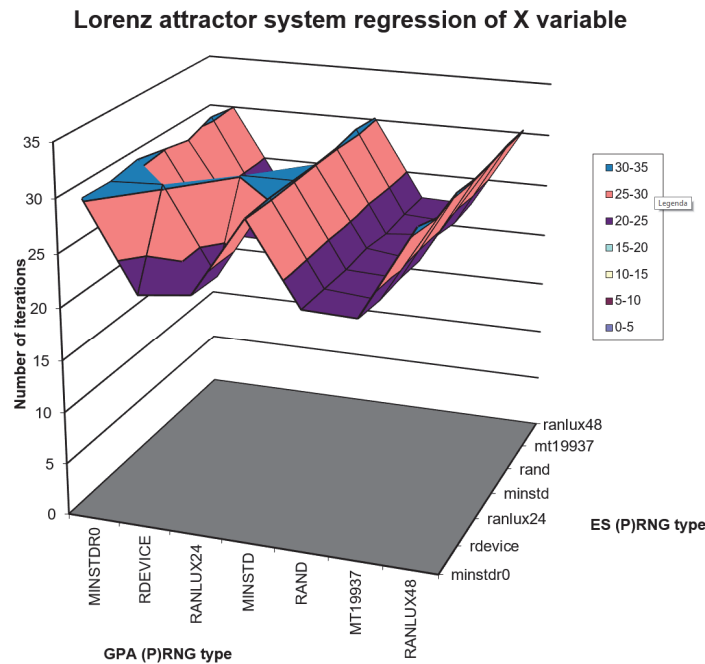


Figure 5: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor X variable symbolic regression using single-thread version of the algorithm with significant differences between studied combinations of PRNGs.

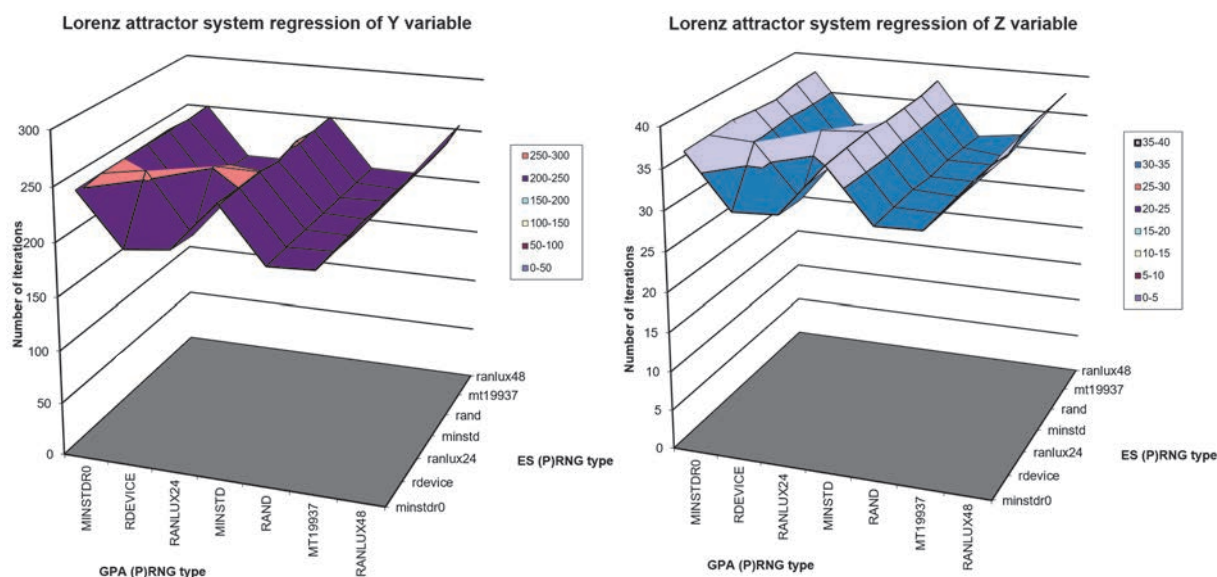


Figure 6: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor Y variable symbolic regression using single-thread version of the algorithm with significant differences between studied combinations of PRNGs.

Figure 7: Dependency of iterations number on number used PRNGs in experiments with Lorenz attractor Z variable symbolic regression using single-thread version of the algorithm with significant differences between studied combinations of PRNGs.

5 Conclusions

In this work, the disturbing effects that depreciate the quality of experiments were identified and their elimination, that was described in the paper, increased the quality of the results. It allowed us to study the influences of GPA and ES part onto composed GPA-ES algorithm, but in the future, it will allow us to produce trustworthy comparisons of different evolutionary algorithms.

As an unscheduled side effect, the longtime non-stationarities of standard PRNGs were observed. It means that comparisons of generators must be based on at least 10^6 observations of presented system, or more versatile it is possible to ask for the use of more than 10^9 random numbers in each experiment (this number depends on numbers of experiments, individuals, and evolutionary cycles).

Presented experiments also points out that the GPA part of the GPA-ES algorithm is more sensitive to PRNG type than the ES one.

Acknowledgement: Computational resources were supplied by the Ministry of Education, Youth and Sports of the Czech Republic under the Projects CESNET (Project No. LM2015042) and CERIT-Scientific Cloud (Project No. LM2015085) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

References

- [1] Poli, R., Langdon, W.B., McPhee N.F.: *A field guide to genetic programming*, Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk> (2008). (With contributions by J. R. Koza).
- [2] Brandejsky, T.: Limited randomness Evolutionary Strategy Algorithm. In: R. Matousek (ed.) *Proceedings of 21st International Conference on Soft Computing – MENDEL 2015*, no. 21 in MENDEL, pp. 53–62. Brno University of Technology, VUT Press, Brno (2015). ISSN 2194-5357. ISBN 978-3-319-19823-1.
- [3] Skanderova, L., Zelinka, I., Šaloun, P.: Chaos Powered Selected Evolutionary Algorithms. *Advances in Intelligent Systems and Computing* 210, 111–124 (2013)
- [4] Cantú-Paz, E.: On Random Numbers and the Performance of Genetic Algorithms. *Proceeding GECCO '02 Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, p. 311–318 (2002)
- [5] Brandejský, T.: Problems of Analyse of PRNGs influence onto the GPA – ES algorithm behaviours. In: *Mendel 2016 22nd International Conference on Soft Computing*. Brno University of Technology, Faculty of Mechanical Engineering, pp. 57–60 Brno (2016). ISSN 1803-3814. ISBN 978-80-214-5365-4.
- [6] Brandejský, T.: Evolutionary Systems in Complex Signal Analysis. In: *ISCS 2014: Interdisciplinary Symposium on Complex Systems*. Dordrecht: Springer, p. 101–108 (2015). ISSN 2194-7287. ISBN 978-3-319-10758-5.
- [7] Matousek, R., Popela, P., Kudela, J.: Heuristic Approach to Stochastic Quadratic Problem: VAR and CVAR Cases. *MENDEL journal series*, Vol. 23 (2017), No.1, pp. 73–78. Brno (2017), ISSN: 1803-3814.