Fitness Value Curves Prediction in the Evolutionary Process of Genetic Algorithms Applied to Benchmark Function

Renuá M. Almeida[1], **Rodrigo M. Rodrigues**[1],
Denys M. F. Ribeiro[1], **Otávio N. Teixeira**[1]

[1]Faculty of Computer Engeneering – Federal University of Pará (UFPA)
Tucuruí – PA – Brazil

{renua.meireles,rodrigo.rodrigues,denys.ribeiro}@tucurui.ufpa.br

Abstract. *This work intends to adopt fitness curves prediction from Genetic Algorithms (GAs) proposed in [Almeida et al. 2021], in the context of a more complex function, which is the Schwefel benchmark function. The prediction is performed with the knowledge only of the GA initialization parameters, using the Random Forest model. This approach addresses the main gap in the original work achieving good results, which makes this approach more promising.*

1. Introduction

The Genetic Algorithms (GAs), presented in [Holland et al. 1992], are stochastic search and optimization methods based on biological evolution and inspired by Darwin's theory of "survival of the fittest" [Goldberg 1989], in which there is a set of individuals/solutions that go through a series of operations. In GAs, the individual's fitness indicates how good a solution for a given problem is.

For the execution of a GA it is necessary to define some basic parameters, more specifically: population size, crossover probability, mutation probability, elitism, and stopping criteria, which is commonly defined by a fixed number of generations. And, the performance of the algorithm has strong links to the definition of these elements [Mitchell 1998].

A simplified way to inspect the evolutionary process of a GA is to record the maximum, average, and minimum fitness values of individuals in each generation. With that, it is possible to plot a fitness by generation chart [Almeida et al. 2021].

Based on this, this work aims to reuse the experiences obtained in GA executions and to train a machine learning model to predict the maximum, average, and minimum fitness curves that emerge during the evolutionary process of a GA, and this, only with the knowledge of its initialization parameters.

There is a great number of works that implement some kind of fitness prediction, but they are usually focused in the context of parameter tuning or approximation and meta-models as alternatives for the original fitness functions calculations [Bhattacharya 2013].

This paper is an extended version of a previous work [Almeida et al. 2021], that was presented as a poster with some promising initial experiments and results but initially applied in a one-dimensional function. Now, we propose to expand the results evaluating a harder and more realistic problem, which is the Schwefel function, a N-dimensional common benchmark function.

2. Methodology

2.1. Definition of Features and Targets

The features follows the same of the original work, which are the current generation ($CurrentGen$), population size ($PopSize$), stop generation ($StopGen$), crossover probability ($CxPb$), mutation probability ($MxPb$), elitism ($Elitism$), lower ($LowerBound$) and upper bound of the adopted problem ($UpperBound$). The targets were not changed as well, which are the maximum ($MaxFitness$), average ($AvgFitness$), and minimum ($MinFitness$) fitness values of a certain generation.

The defined problem is the minimization of the Schwefel Function, a common benchmark problem. This function is one of those hard multimodal function that are difficult to solve, as it contains several local minima locations, however global minima 0 is located at $f(x^*) = [1, 1, ..., 1]$ in the domain of $[-500, 500]$ [Hussain et al. 2017]. Mathematically, this function is expressed as $f(x) = \sum_{i=1}^{D} -x_i \sin\sqrt{|x_i|}$, where $D = Dimension$. A common value adopted for dimension is 30 [Hussain et al. 2017].

2.2. Dataset

The details about the algorithm implemented to generate the dataset are given as follows, floating-point was used for chromosome representation, tournament method for selection under the configurations of three participants and one round, two-point method was used for crossover operation, swap method was chosen for mutation, and the replacement of new individuals in the population is random.

Altogether, 1.000 executions were defined, and in each, the initialization parameters were uniformly randomized under predetermined range restrictions, these definitions are given by $Parameters\_Range = \{PopSize : [50, 200], StopGen : [50, 500], CxPb : [0.7, 1.0], MxPb : [0.0, 0.05], Elitism : [0, 10], LowerBound : [-500, -1.0], UpperBound : [1.0, 500]\}$. In each run, G records are added to the data set, where $G = StopGen$. After all executions, approximately 270.000 records were obtained.

The range restrictions were defined based on the parameters commonly adopted by researchers in experiments and tuning processes, where mutation probabilities are low and crossover probabilities are high [Eiben and Smit 2012]. The values for population size and stop generation were defined based on the behavior of the problem, in which the algorithm converged quickly without the need for a high variety of the initial population and a large number of generations. Finally, values for upper and lower domains of the problem were defined according to it's common values [Hussain et al. 2017]. A comparison between the difference between the values adopted and the values adopted in [Almeida et al. 2021] can be seen in figure 1.

	PopSize	StopGen	CxPb	MxPb	Elitism	Executions	Function Bounds	Function Dimension
Almeida et al. 2021	[40, 70]	[40, 70]	[0.5, 0.9]	[0.0, 0.4]	[0, 20]	3000	[-50.0, 50.0]	1
Current work	[50, 200]	[50, 500]	[0.7, 1.0]	[0.0, 0.05]	[0, 10]	1000	[-500.0, 500.0]	30

Figure 1. Comparison of adopted values.

2.3. Predictive Model

As in the original work, a Random Forest model was used for the prediction process with 85% of data for training and 15% for testing. Random Forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large [Cutler et al. 2012].

2.4. Evolutionary process prediction

To verify if the model is capable of predicting the entire evolutionary process of a GA only with the information of its initialization parameters. it is necessary to generate the values of the features of each generation. To do this, as in [Almeida et al. 2021], simply replicate the initialization parameters for all generations, and for feature $CurrentGen$ a sequence of integers from 1 to G is generated, where $G = StopGen$.

To carry out this test, 2 parameter configurations for a GA were generated, maintaining the same parameter rules imposed in the generation of data for the training and validation of the model. The configurations are $C1 = \{PopSize : 183, StopGen : 312, CxPb : 0.79, MxPb : 0.03, Elitism : 4, LowerBound : -6.0, UpperBound : 472.0\}$ and $C2 = \{PopSize : 83, StopGen : 304, CxPb : 0.94, MxPb : 0.02, Elitism : 7, LowerBound : -364.0, UpperBound : 259.0\}$

3. Results and Discussions

In accordance with the original work, the use of the model to predict the targets of a complete evolutionary process proved to be very effective, in the same way it is noticed that the target $MinFitness$ (best individuals fitness) and $AvgFitness$ are the values that the model can more easily predict when compared to $MaxFitness$ (worst individual fitness). This can be seen in figure 2, which presents the results of the predictions for parameter configuration $C1$.

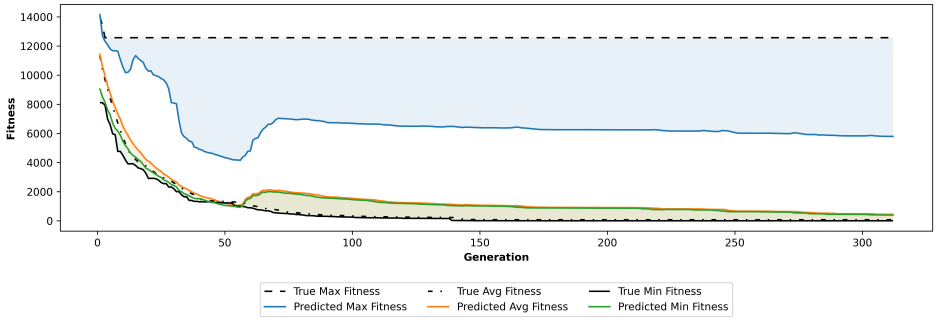

Figure 2. Results of model predictions under parameter configuration $C1$.

Figure 2 makes evident the model's superiority in predicting the $MinFitness$ and $AvgFitness$ values of each generation, but this time, different from the original work, the model does not get close to the $MaxFitness$ real values. The prediction of the evolutionary process of the parameter configuration $C1$ obtained MAE of 2505.26. In fact, the errors generated by the predictions of $MaxFitness$ target end up negatively affecting the error metric. Below, in figure 3, the result of the predictions for configuration $C2$ is shown.

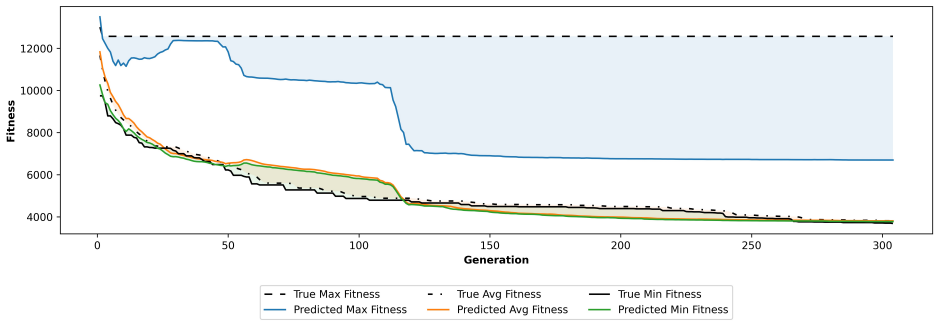

Figure 3. Results of model predictions under parameter configuration $C2$.

4. Conclusions

The results presented in this paper confirm the great potential in this approach. The original work had bad times during the validation of the results mainly because of the problem adopted, which was a very non-realistic problem in the context of optimization through GAs.

The main contribution of this paper complementing the original work, is the confirmation that it is possible to use this approach in the context of more complex problems with many local optimums with a wide search space.

References

Almeida, R. M., Ribeiro, D. M. F., Rodrigues, R. M., and Teixeira, O. N. (2021). *Fitness Value Curves Prediction in the Evolutionary Process of Genetic Algorithms*, page 221–222. Association for Computing Machinery, New York, NY, USA.

Bhattacharya, M. (2013). Evolutionary approaches to expensive optimisation. *International Journal of Advanced Research in Artificial Intelligence*, 2(3).

Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). Random forests. In *Ensemble machine learning*, pages 157–175. Springer.

Eiben, A. E. and Smit, S. K. (2012). Evolutionary algorithm parameters and methods to tune them. In Hamadi, Y., Monfroy, E., and Saubion, F., editors, *Autonomous Search*, pages 15–36. Springer.

Goldberg, D. E. (1989). Genetic algorithms in search. *Optimization, and MachineLearning*.

Holland, J. H. et al. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Hussain, K., Salleh, M., Cheng, S., and Naseem, R. (2017). Common benchmark functions for metaheuristic evaluation: A review. *International Journal on Informatics Visualization*, 1:218–223.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.