FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Self-Adaptation in Evolutionary Strategies

Evolutionary Computation 2016/2017

Gabriel Rodrigues, 2016211454
Gonçalo Pereira, 2008119715

June 4, 2017

Introduction and Contextualization

Genetic algorithms have been used to study problems with enormous spaces of solutions. These algorithms, under the category of heuristic approaches, have shown to be useful and successful in finding near optimal solutions in a reasonable amount of time. However, in order to achieve this, the algorithm has to be well designed for the problem in subject and, also, it has to be properly tuned.

Regarding the tuning of genetic algorithms, there is a main issue that has been studied along the years, this is, which specific values for the parameters of the algorithm result in the best performance possible [1]. Statistic studies have been applied to different genetic algorithms and different problems. The technique is called: “Parameter tuning” [2]. However, the results appear to be problem dependant and not generic.

Due to this incognita, some researchers have proposed different approaches to solve this problem, being parameter tuning the most simple and widely used. Nevertheless, different approaches, called: “Parameter Control” strategies [2], have been studied.

Three categories were discriminated for parameter control: deterministic, adaptive, and self-adaptive [2]. The deterministic parameter control is about changing the values at a specific rate over generations [3]. The adaptive parameter control, uses knowledge about the search to adapt the values of the parameters [4]. The self-adaptive parameter control lets the evolution algorithm to set the values for the parameters and to evolve them [5]. In addition to this, other researchers realized that this issue is also an optimization problem, so they could use genetic algorithms or other optimization algorithms to do parameter tuning [6].

In this project we studied the approach, called “Self-Adaptation”. The premise is that the best parameters will be possessed by the best individuals, and those values will spread over generations, being tweaked by crossover and mutation.

The purpose of this project is to show a comparison between two different parameter selecting approaches. The first one is the “standard” one, having fixed parameters for the whole execution of it. The second one is a “self-adaptive” strategy, in which we let evolution modify the parameters. More specifically, we compared these two approaches using different benchmark functions, a “*float*” representation and the parameter studied is called “*sigma*”, which in the decimal representation for genetic algorithms defines the amount of possible change for a specific gene that is being mutated.

The following sections of this paper describe the functions to be used as benchmark, our implementation, the design of the experiences, its results, discussion and finally, the conclusions of the whole project.

Implementation

In order to make the comparison of the two parameter selection strategies, we had to first implement the genetic algorithm for the decimal (float) representation. The genetic algorithm in subject was implemented using the base code provided by the professor Ernesto Costa for the practical classes. Its representation can be seen in figure 1.

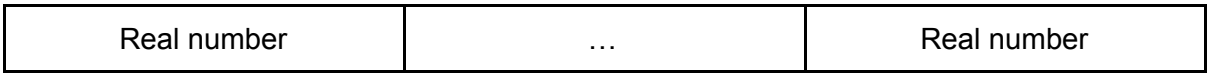

Figure 1 - Real number representation.
A vector with n elements, each element is a real number.

The following step was to build the alternative representation for the “Self-Adaptive” strategy and adapt the standard implementation to support the new representation. In figure 2 we can appreciate the new representation.

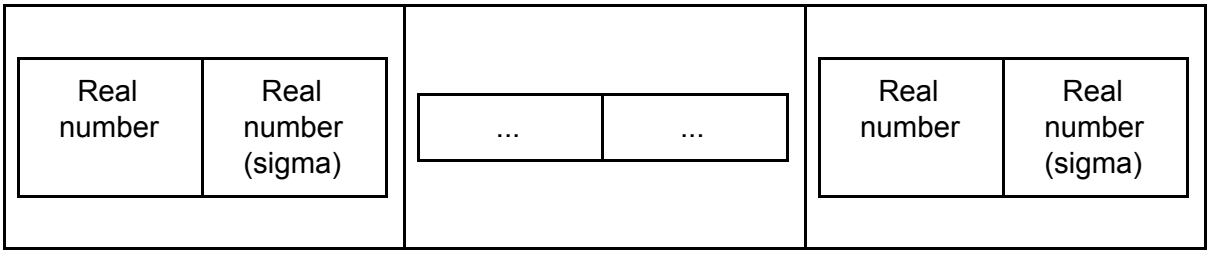

Figure 2 - Self-Adaptive representation.
The second value in each gene represents the sigma value for that specific gene.

A new genetic algorithm based on the standard one was adapted to support this new representation. The sigma values for each gene were recombined the same way as the normal values, and its mutation was done using the domain given for sigma and a normal distribution. As is the purpose of the “Self-Adaptive” approach, the values associated to the solution were mutated using the sigma value associated to the gene they were on.

To fulfil the objective, we had to implement the benchmark functions to test the performance of the different parameter selection strategies. The three selected benchmark functions were the following:

The Rastrigin function. It is a multimodal function with many local minima. This function was chosen due to the amount of local minima, fact that is going to be useful to assess the capability of the algorithm to escape local minima. See figure 3.

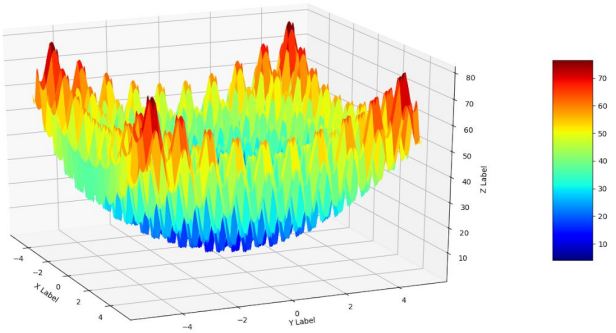

Figure 3 - Rastrigin function in 3D.

The Schwefel function. It is also a multimodal function with many local minima, which was chosen for the same reasons as the rastrigin function. See figure 4.

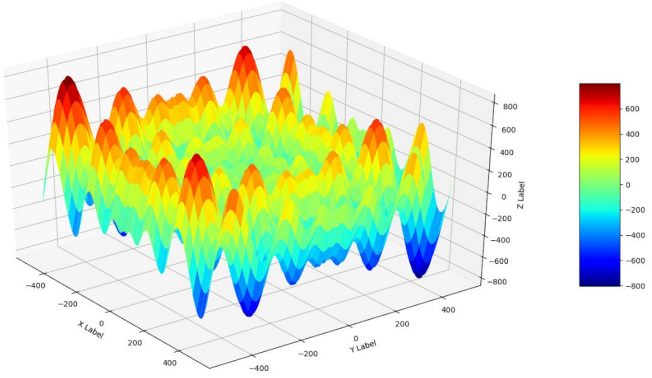

Figure 4 - Schwefel function in 3D.

The Sphere function. It is a continuous, convex and unimodal function. This particular function, because of its convexity, can be easily solved by a hill-climber algorithm. It was used to compare the performance of the strategies with a very simple problem. See figure 5.

The Griewangk function. Although at simple sight this functions appear to be similar to the previous one, it is a multimodal, non-convex function. For the Griewank, there is also many local minima, similarly as in the Rastrigin and Schwefel, however, these local minima are closer together. See figure 6 and 7.

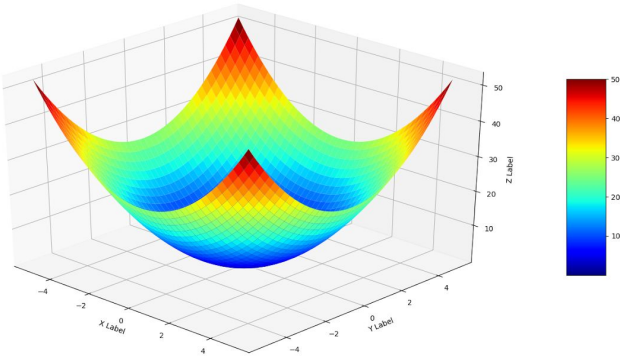

Figure 5 - Sphere function in 3D.

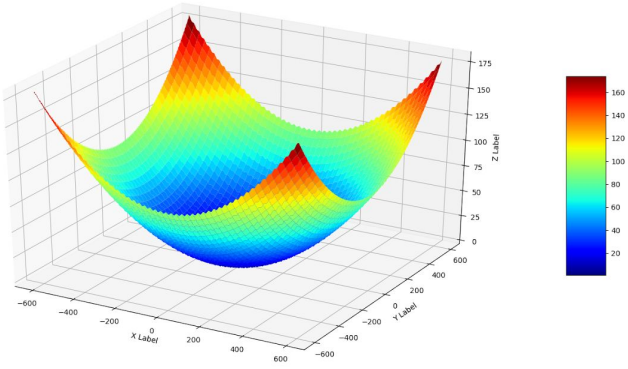

Figure 6 - Griewangk function in 3D.

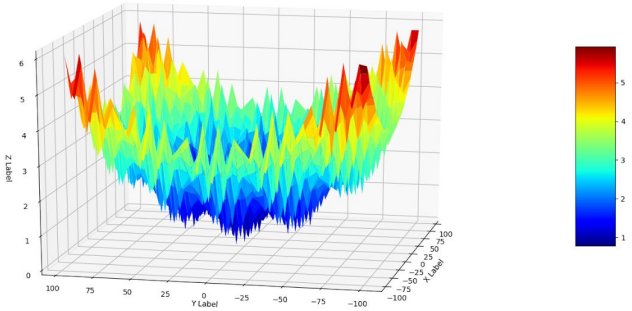

Figure 7 - Griewangk function in 3D zoomed in.

Experimental Setup

Having two approaches, Standard and Adaptive, to experiment on and four different benchmark functions, we saw ourselves with a total of 8 experiments to perform. All experiments were performed by running the algorithms with the same configuration of parameters for a total of 100 independent runs. Additionally, since the initial populations were not equal for both approaches, we considered the experiments to be totally independent.

In order to compare the two approaches, we ran each pair of algorithms with the same parameters. Firstly the dimensionality of all problems were set to 20, the number of generations to 1000 and the population size to 100. For operators, mutation was performed by a Gaussian Mutation operator and crossover by a Heuristical Crossover operator. The parents selection was done using Tournament (size 5) and the survivors selection by Elitism (top 10%). For the other parameters, values had to be tuned in order to counter differences between benchmark functions. In the following table we present a summary of differences in configuration for each experiment:

	Rastrigin		Schwefel		Sphere		Griewangk	
Param	Stand.	Adapt.	Stand.	Adapt.	Stand.	Adapt.	Stand.	Adapt
Problem Domain	-5.12 to 5.12	-5.12 to 5.12	-500 to 500	-500 to 500	-5.12 to 5.12	-5.12 to 5.12	-600 to 600	-600 to 600
Mutation Probability	0.01	0.01	0.1	0.1	0.01	0.01	0.1	0.1
Sigma Domain	0.5	0 to 1	100	0 to 200	0.05	0 to 1	100	0 to 200
Cross. Probability	0.9	0.9	0.8	0.8	0.9	0.9	0.8	0.8

Table 1: Genetic Algorithm parameters

Having the experiments ready the only thing left was to decide the procedure for comparing and analyse results. We chose the Shapiro-Wilk test in order to decide if the results of each experiment followed or not a normal distribution. For this test the null hypotheses was that the distribution of the population of each experiment is normal and the alternative hypotheses that the distribution did not follow a normal distribution. To compare results of the 2 different algorithms for each benchmark function we decided to use the Mann-Whitney U test. This test is appropriate for two class, independent and nonparametric tests (as we will see ahead) and uses as the null hypotheses that “the results’ population have equal distributions”, or, by other words, that the two results are statistically equal.

Results

Rastrigin Function

With the two algorithms executed 100 times, the results were obtained for the Rastrigin function (summarized in table 2). Additionally the normality of the distributions was tested giving a significance value of 2.8e-10 for the standard algorithm and 3.2e-15 for the Adaptive. Given this results we reject the null hypotheses and conclude with a significance level of 1% that both samples do not follow a normal distribution. With this result we opt for the non-parametric test.

	Standard	Adaptive
Min	0.00046	0.0000005
Max	0.01594	5.97055
Mean	0.00371	0.55066
Median	0.00322	0.00110
Std	0.00284	0.98838
Skew	2.11362	2.82332
Kurtosis	6.01361	9.92851

Table 2: Results metrics for the Rastrigin function

Following the decision for the parametric test, the Mann-Whitney U test was used to find if there was significant differences in both samples' populations distributions. With a U value of 4355, z-score of -1.576 and a significance value of 0.058, we conclude that we cannot reject the null hypotheses and thus we cannot find significant differences in both distributions.

Schwefel Function

This time for the Schwefel function we repeated the procedure and tested the normality of the distributions. Similar to the Rastrigin function we could not prove the normality of the distributions given that the Adaptive algorithm gave a significance value of 0.472 (although irrelevant given the Adaptive result, in this function the normality test for the standard algorithm was proven to follow a normal distribution with a significance value of 0.009).

	Standard	Adaptive
Min	-7787	-7544
Max	-6602	-5821

Mean	-7317	-6857
Median	-7313	-6843
Std	251	322
Skew	0.45275	0.34229
Kurtosis	-0.05315	0.28059

Table 3: Results metrics for the Schwefel function

Again, given the non-normality of the data, we used the Mann-Whitney U test this time with a different outcome. With a U value of 1276, z-score of -9.099 and significance of 4.6e-20 we could reject the null hypotheses (with 1% of significance) and assess which algorithm was best. With a big effect size of r -0.643, and looking at the mean and minimum values for each algorithm we can say that the Adaptive algorithm performs worst than the standard one.

Sphere Function

Proceeding in the same fashion, for the Sphere function, we again could not prove the normality of the samples' population distribution. With a significance value 1.4e-09 for the standard and 7.8e-017 for the Adaptive we could not reject the null hypotheses and thus, one more time, opted for the nonparametric test.

	Standard	Adaptive
Min	4.38732e-08	1.20976e-09
Max	1.66759e-06	5.51235e-05
Mean	3.14228e-07	3.91852e-06
Median	2.27366e-07	1.81591e-06
Std	2.60620e-07	6.96375e-06
Skew	1.99901	4.67114
Kurtosis	6.19685	27.99937

Table 4: Results metrics for the Sphere function

Using the Mann-Whitney U Test the result was the opposite of the Schwefel case. With a U value of 926, z-score of -9.954, and significance of 1.2e-23 we rejected the null hypotheses and concluded that results were different. This time with a higher effect size of -0.704 and looking at the values in table 4, we concluded that, opposite to the Schwefel case, our Adaptive algorithm was performing better.

Griewangk Function

Finally, the Griewangk function gave results similar to the Schwefel function overall. We could not prove the distributions' normality given the standard results (significance of 0.051 for the standard and 1.9e-06 for the adaptative).

	Standard	Adaptive
Min	0.03215	1.02270
Max	0.27709	1.30395
Mean	0.11435	1.09777
Median	0.11401	1.08468
Std	0.04399	0.05137
Skew	0.60120	1.33020
Kurtosis	0.80010	2.16795

Table 5: Results metrics for the Griewangk function

On the nonparametric Mann-Whitney U test, with a U of 0, a z-score of -12.217 and significance of 1.2e-34 we could reject the null hypotheses with a significance of 1%. Additionally, given an high effect size of -0.8 and the values in table 5, we can clearly state that the Standard algorithm performs better.

Conclusions

The parameter control strategy is meant to evolve the values for the sigma parameter. However, the understanding of its evolution is difficult to comprehend.

Given the results obtained we conclude that the adaptive approach implemented could not improve on the standard approach. Although in one of the functions the adaptive was indeed better, that function was the simplest of the four. We attribute this outcome mainly to an effect that was seen in the evolution of the sigma in the population of the genetic algorithm, where the value's behaviour wasn't consistent between generations or runs, sometimes rising and other times dropping to lower values. In light of this conclusion and the behaviour observed we leave ourselves wondering if our implementation missed on some key aspect of the self adaptive approach and leave, as future work, the idea of placing some selective pressure on the evolution of the sigma values.

On the note of future work, we also propose the experimentation with the other approaches mentioned in the introduction of this work, mainly the non-self adaptive and deterministic. Finally, we also leave the justification of the chosen parameters using statistical methods for later work, mainly due to time constraints.

Despite the results, we found this work invaluable to build knowledge on the multiple aspects of the parameter control techniques associated to genetic algorithms and on the statistical process involved with experimentation and result analysis.

References

[1] Kramer, O., & Ting, C.-K.(2004). *Self-adaptive Evolutionary Algorithms*. Germany: University of Paderborn.

[2] Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). *Parameter Control in Evolutionary Algorithms*. IEEE.

[3] Back, T. (1992). *Self Adaptation in Genetic Algorithms*. University of Dortmund.

[4] Nakajima, H., Nakadai, K., Hasegawa, Y., & Tsujino, H. (2008). *Adaptive Step Size Parameter Control for Real-World Blind Source Separation*. Honda Research Institute Japan Co., Ltd.

[5] Spears, W. (1995). *Adapting Crossover in Evolutionary Algorithms*. Proceedings of the Fourth Annual Conference on Evolutionary Programming.

[6] Smit, S., & Eiben, A. (2009). *Comparing Parameter Tuning Methods for Evolutionary Algorithms*. Norway: IEEE.