UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Computational Intelligence

Genetic algorithms/Evolution strategies

Professors

NEBOT CASTELLS, Angela
ALQUEZAR MANCHO, René
ROMERO MERINO, Enrique

Team members

BEJARANO SEPULVEDA, Edison Jair
REYES FERNANDEZ, Grecia

Polytechnic University of Catalonia
C/Jordi Girona Salgado,1-3 08034 BARCELONA Spain
Master's degree in Artificial Intelligence
Barcelona, 2021

The goal of the present work is to use genetic algorithms to find the minimum value of the Rosenbrock function.

Throughout this exercise, multiple parameters and operators are varied in order to determine which values give better results for this specific problem. The options that were varied are: population size, number of generations, initial range, selection and reproduction (crossover and mutation). In the following sections, we discuss the influence of each one of them.

1 Rosenbrock's function:

The Rosenbrock's function, also known as the valey function, is a non-convex function used as a test problem for optimization algorithms, particularly for gradient-based algorithms. It was introduced by Howard H. Rosenbrock in 1960, and finding its optimum is rather challenging because the global minimum is inside a long, narrow, parabolic shaped flat valley. For our case, let the Rosenbrock function by defined as follows:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \tag{1}$$

The location of the minimum as well as its value are well known, and for our case we know that the point that we are looking for in this exercise is at $x = 1$, $y = 1$ and $f(1, 1) = 0$.

2 Finding better parameters

In order to find the best parameters that minimize the Rosenbrock's function, we applied the genetic algorithm, which are optimization and learning algorithms inspired by the processes of natural and genetic evolution. The fitness function is $f(x, y)$ (1), that represents the objective function of our optimization problem, we consider the number of variables is 2. We executed each configuration 1,000 times and compute the mean results to obtain a smoother plots:

2.1 Population size:

We test with a population sizes from 1 to 50 in intervals of 2, to determine the simplest genetic algorithm that gives good results to this problem.

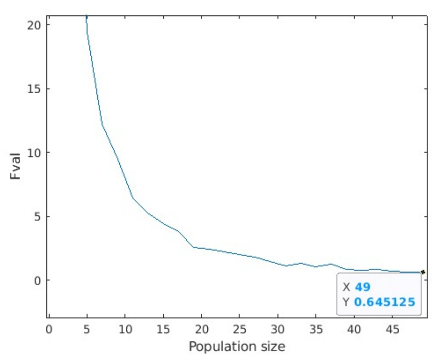

Figure 1 – Evaluating the population size

As shown in Figure 1, there is a clear correlation between the population size and the ability of the algorithm to reach the minimum value of the function: for larger populations, the error decreases. However, because the error plateaus rapidly, there is not a big difference in choosing a population of 50 or greater, but increasing the population means that the model is more complex. Therefore, we chose a trade-off value of 49 for the population size, which yields a value of the objective function of 0.645125.

2.2 Number of maximum generations:

After choosing the population size, we test the number of generations in the range from 20 to 200 in intervals of 10.

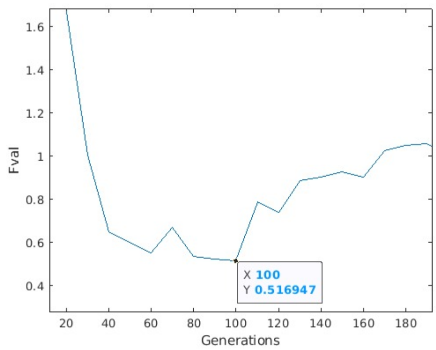

Figure 2 – Evaluating the number of max. generations

As we can see in Figure 2, the results seem to improve as the maximum number of generations increases until a certain point is reached; from then on, the tendency of the results seems to worsen. The number of maximum generations that gave the best results (that is, the lowest value of $f(x, y)$) in our case was 100, and its corresponding value of the objective function was 0.516947.

2.3 Initial Range:

The initial range vector specifies the range of the individuals in the initial population. To test the influence of this parameter on the results, we set the lower limit of the initial test different values for the upper limit of the initial range, range from 1.1 to 3, in intervals of 0.1.

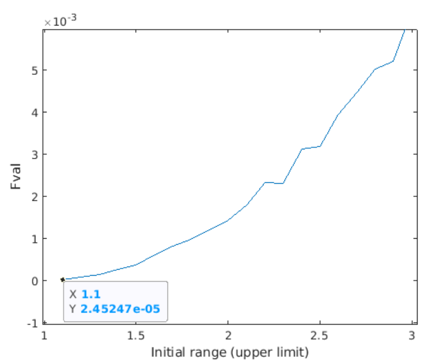

Figure 3 – Evaluating the initial range

In figure 3, the best results are shown for the smallest initial range possible, so we consider that the best range for the initial values of the population is [1, 1.1], and its corresponding value of the objective function was $2.45247e-05$.

2.4 Selection function:

This parameter is used to choose the best adapted individuals of the population. It is a set of rules that serve to choose the parents of the next generation. We chose among the following selection-functions:

- no selection-function
- 'selectionstochunif'
- 'selectionremainder'
- 'selectionuniform'
- 'selectionroulette'

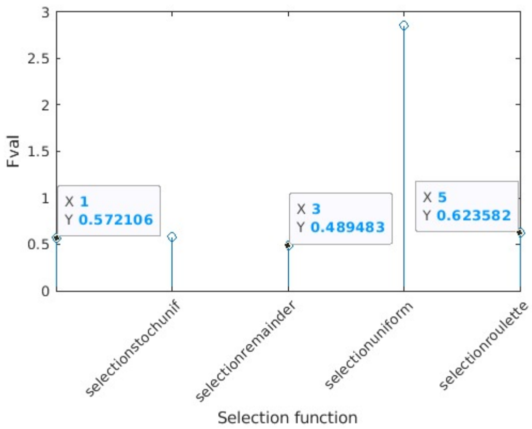

Figure 4 – Evaluating selection

As can be seen in Figure 4, the best performance is achieved with the selection-function 'selectionremainder', for which the value of the objective function is minimum and it corresponds to 0.489483.

2.5 Reproduction (Crossover and Mutation):

We use the crossover function to specifies the fraction of the next generation. It is the most significant phase of genetic algorithm, because chromosomes will be generated which are far better than the previous generation chromosomes. In Figure 5 we present the result in the range from 0 to 1, in this configuration we get the best result between 0.5 and 0.8, there are similar result, however we chose 0.55.

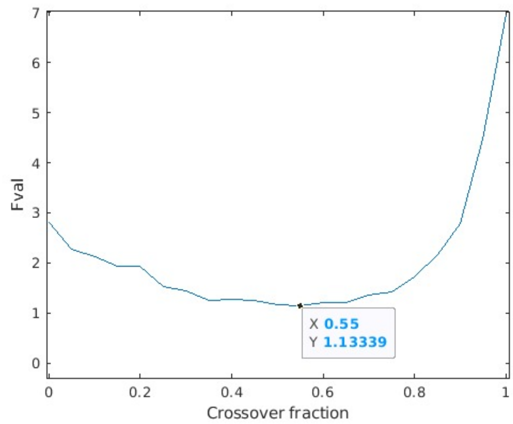

Figure 5 – Evaluating the reproduction

3 Minimum value of the Rosenbrock function

Finally, after testing each parameter with different values, we have the best parameters to configure our genetic algorithm, which are the following:

- Population size: 49
- Number of generations: 100
- Initial Range: [1, 1.1]
- Selection: 'selectionremainder'
- Crossover fraction: 0.55

3.1 Where is the global minimum? Which is the global minimum?

With these parameters, the global minimum of the function found by the genetic algorithm (fval) as well as its position (x and y) are shown in the following table:

Table 1 – Global minimum from the genetic algorithm optimisation with the best parameters.

	Global minimum	**Absolute error**
x_ga	0.9993	6.9878e-04
y_ga	0.9985	1.4723e-03
fval_ga	1.0539e-06	1.0539e-06

In Figures 6 and 7 the location of the global minimum point of the Rosenbrock's function as approximated by the genetic algorithm is shown with a red star, both in the two-dimensional plane and in the three-dimensional space.

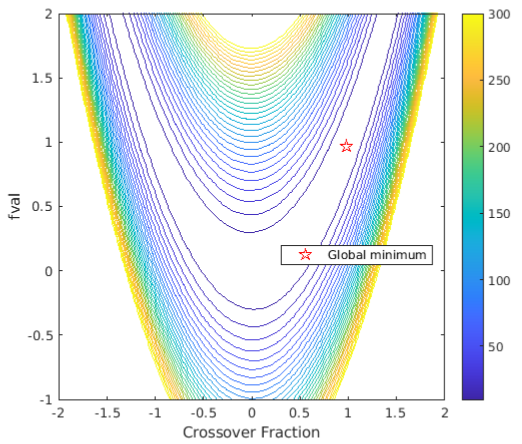

Figure 6 – Location of global minimum (2D)

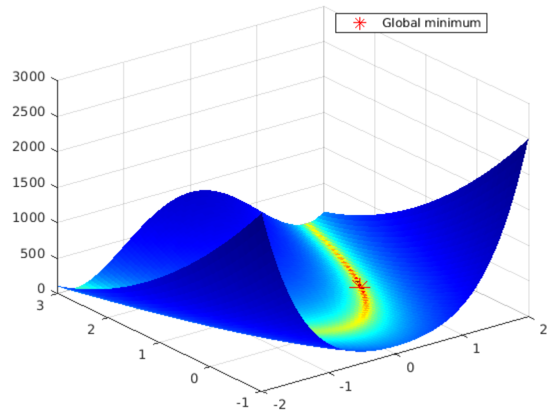

Figure 7 – Location of global minimum (3D)

Additionally, Figure 8 shows three subplots: first, the stopping criteria that forced to end the iterative process of the algorithm (the maximum generation limit is reached); second, the evolution of the best and the mean values through the iterations; and third, the current best individual, or in other words, the x and y coordinates of the minimum.

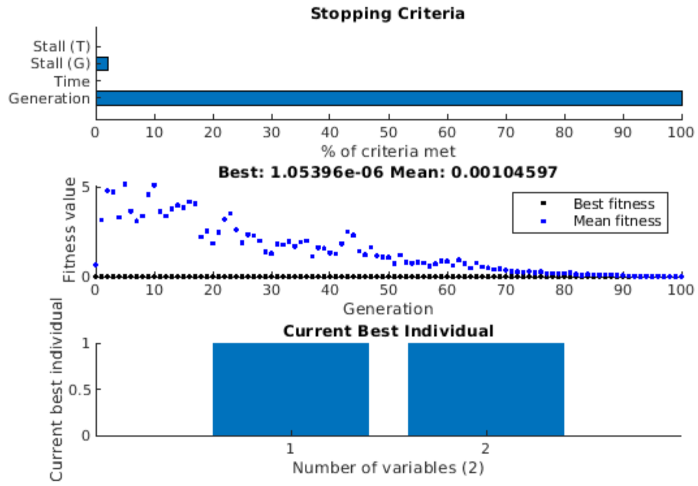

Figure 8 – Results from the genetic algorithm optimization with the best parameters.

4 Conclusions

In this laboratory, the optimization of the Rosenbrock function is performed using a genetic algorithm. Based on the findings, this algorithm has been shown to provide good results in the optimization problem. It is observed that it was necessary to increase the number of iterations when finding each parameter to obtain a stable result.

On the other hand, we can see that the parameters have a strong correlation between them, so it was necessary to test many configurations.

On the other hand, for us, the most important parameter is the mutation and the crossover, in the first parameter, the individuals with the greatest variation can be operated. The second parameter generates a new generation through the mutation of the parents of the previous generation, the offspring of the new generation comes from carrying genes from both parents. These two are the ones that have the greatest impact on the algorithm for us and it is important to select correctly to find a high-quality solution.

5 References

[1] https://www.mathworks.com/matlabcentral/fileexchange/36883-rosenbrock-function

[2] https://la.mathworks.com/help/gads/gaoptimset.html#mw_4bef4408-20fc-4fe2-9311-2b122025a7b8