Artificial Intelligence
Genetic Algorithms Coursework

March 8, 2021

1 Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vitae bibendum risus. Vestibulum mattis dui eros, eu tristique orci egestas eu. Maecenas hendrerit mi eget nulla malesuada hendrerit. Praesent egestas dui eget ipsum fringilla, vitae sagittis urna varius. In hac habitasse platea dictumst. In nec magna tellus. Nullam tempus rutrum lectus, nec ornare urna posuere non. Praesent nec arcu tristique nisi elementum luctus nec quis lorem. Cras ullamcorper urna vitae volutpat euismod. Nunc tincidunt lorem et augue interdum sodales. Quisque erat mi, viverra ut quam a, rutrum ornare mi. Donec eget sagittis metus. Vestibulum ante ipsum primis in faucibus orci.

2 Continous Optimisation

Note 1 *All the results for this section were produced with the following parameters.*

- *N=4*
- *Lower bound = -5*
- *Upper bound = 5*

No further mentions will be provided for those hyperparameters.

2.1 Subtask 1.C: Performance

For our algorithm evaluation, we will use two classic optimization testing functions, F1 and F4.

2.1.1 Sphrere

The Sphere(Commonly known as F_1 in the literature[3]) contains a single minimum and is considered an easily solvable function.

$$F_1(X) = \sum_{i=1}^{n} x_i^2$$

2.1.2 Rastrigin's function

Rastrigin's function(Commonly known as F_4 in the literature[3]) is considered a challenging task due to its large number of local minima and its enormous search space.

$$F_4(X) = 10 \cdot n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$$

2.1.3 Optimal hyperparameters

Before evaluating our algorithm's performance, we should first find the optimal hyperparameters for solving our test functions. We need to evaluate the following 3 variables and their affect on the algorithms performance.

- Mutation Rate(MR)
- Crossover Rate(CR)
- Population Size(PO)

We will use the following unbiased metric for measuring the objective performance for a given function(F_1 or F_4) applied on the giver parameters (MR, CR, PO).

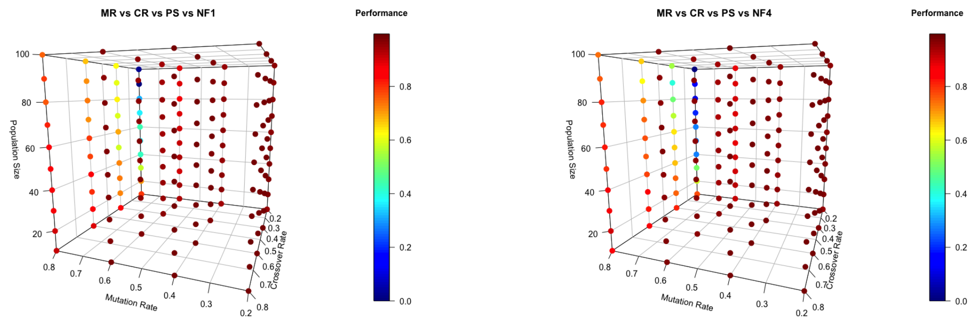
Definition 1 *We call the following metric 'Normalized Performance for function F_x'*[a]

$$NF_x = 1 - \frac{g \cdot PO}{G \cdot max(PO)}, 0 \leq NF_x \leq 1$$

Where the g is the average number of generations untill the solution is found after 10 simulations, for a given MR,CR,PO on a given function F_x, G is the maximum number of generations allowed and $max(PO)$[b] *is the maximum population size allowed.*

[a] Our performance model, for siplicity, assumes that the calculation of a single generation takes constant time (O(1))

[b] We multiply PO and max(PO) here, as can be proven that, without accounting the population size, our metric is biased



(a) Performance for F_1 (b) Performance for F_4

Figure 1: Performance for 160 simulations per function

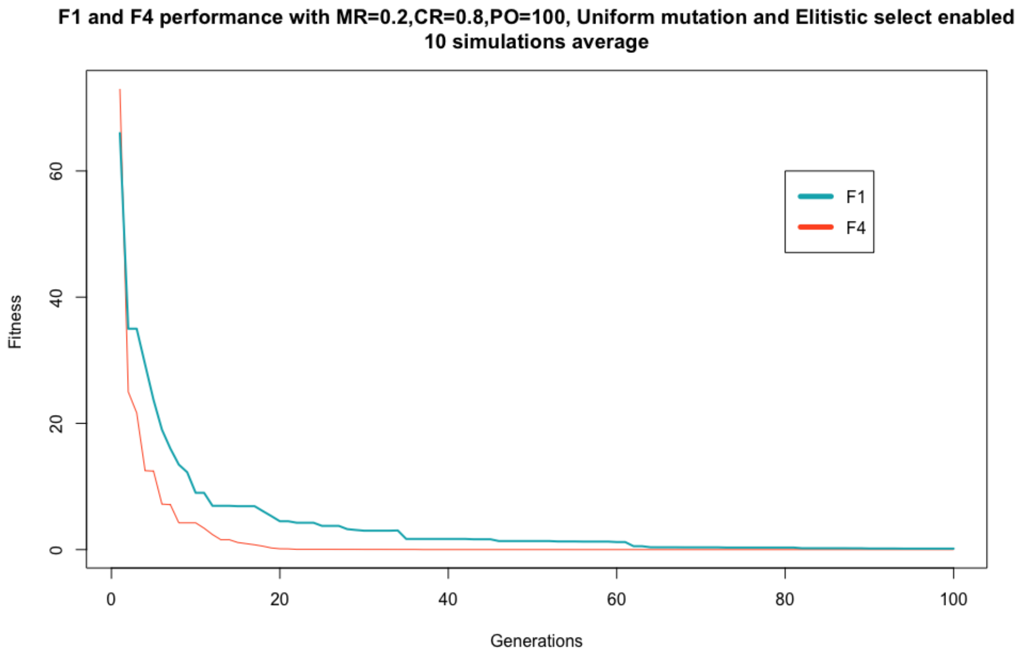
By visualizing the results, we can see that we get a very consistent performance by using the following hyperparameters.

	Lower Bound	Higher Bound
MR	0.2	0.4
CR	0.6	0.8
PO	80	-

Figure 2: Optimal hyperparameters

2.1.4 Performance Results

Using the optimal hyperparameters, as well as our optimal select and mutate operators (see 2.2)



We can see that the algorithm converges close enough to the global minima of each function in the first few generations, then it spends the majority of its time oscillating around the desired result. This can be solved using more sophisticated mutation and crossover operators that are outside of the scope of this coursework.

2.2 Subtask 1.D, algorithm tuning

The initial version of the algorithm was improved in the following 2 areas

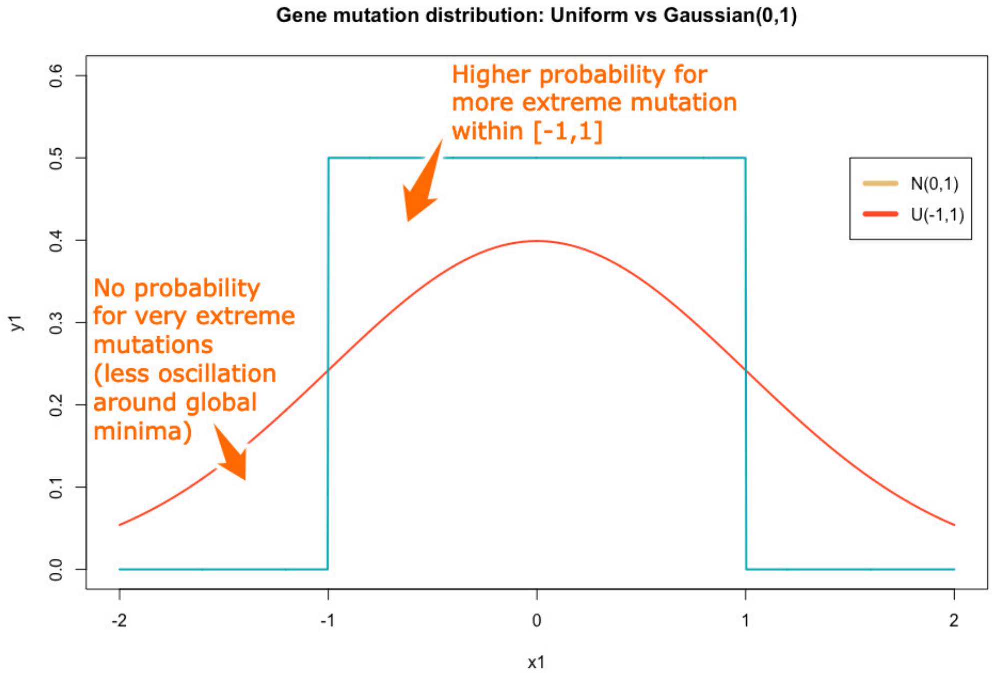
- Balanced and Elitistic selection operator
- Mutation step distribution

2.2.1 Balanced and Elitistic selection

The initial implementation of the selection operator involved a balanced wheel selection based on the relative fitness of the individual on the current generation, as well as an individual probability of each individual to mate. Using both metrics, we determined if an individual was allowed to mate or not. This implementation allowed less ideal candidates for mating with some probability, as this approach allowed the algorithm to avoid local minima traps efficiently. After some refinement, this approach was abandoned for a more straightforward elitistic wheel selection. In this selection strategy, relative fitness is the only factor that affects an individual's probability to mate. The data below shows the improvement using the more straightforward method.

2.2.2 Gaussian and Uniform mutation step

The mutation operator's initial implementation used a step taken from a Gaussian distribution with $\mu = 0, \sigma = 1$. The desired effect was to select a tiny step as a mutation operation most of the time while leaving a small probability for a more significant step to avoid local minima in case of a stuck. The performance analysis showed that the gaussian mutation operator was not 'wide' enough, and the algorithm spent an enormous time stuck on local minima. This was improved significantly by choosing a uniform distribution within the range [-1,1]. As the probability for a more considerable step increases, the probability of sticking into some local minima for too long decreases, and overall we can observe a performance gain as we reduce the number of generations to converge significantly



> **Note 2** *The following results was produced with the following hyperparameters*
>
> - *Population 1000*
> - *Crossover Rate 0.8*
> - *Mutation Rate 0.2*

	Balanced	Elitistic
F1	124	98
F4	107	87

(a) Balanced vs Elitistic(Gaussian Mutation)

	Balanced	Elitistic
F1	133	90
F4	94	67

(b) Balanced vs Elitistic(Uniform Mutation)

	Balanced Gaussian	Elitistic Uniform
F_1	-6%	+8%
F_4	+12%	+22%

(c) Performance differences between the methods

Figure 3: Performance results on 10-Round averages

References

[1] *Bäck, Thomas, Evolutionary Algorithms in Theory and Practice (1996), p. 120, Oxford Univ. Press*

[2] *Holland J.H. (1984) Genetic Algorithms and Adaptation. In: Selfridge O.G., Rissland E.L., Arbib M.A. (eds) Adaptive Control of Ill-Defined Systems. NATO Conference Series (II Systems Science), vol 16. Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-8941-5_21*

[3] *Carvalho, D. B. et al. "The Simple Genetic Algorithm Performance: A Comparative Study on the Operators Combination." (2011).*