Artificial Intelligence
Genetic Algorithms Coursework

March 5, 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

2.1 Task 1:Continous Code

Note 1 *Please find extensive documentation on how to run the following examples on /continous/README.md*

2.2 Subtask 1.C: Performance

For the evaluation of the algorithm, we will use two standarized functions,

2.2.1 Sphrere

the Sphere(Commondly known as F_1 in the literature[3]) contains a single minima and its considered a easily solveable function.

$$f(x) = \sum_{i=1}^{n} x_i^2$$

2.2.2 Rastrigin's function

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

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

2.2.3 Results

We will evaluate against, Mutation and Crossover rates as well as population.

The following results are the averages after 10 runs for each function, on 4 dimensions.

MR	Avg F1	Avg F4
0.2	36	35
0.4	96	104
0.6	-	510

(a) Mutation rate (CR=0.8,Population size=10000)

CR	Avg F1	Avg F4
0.2	125	127
0.4	82	66
0.6	53	42
0.8	42	38
1	30	29

(b) Crossover rate (MR=0.2,Population size=10000)

Population size	Avg F1	Avg F4
100	536	344
1000	88	103
10000	43	31
100000	-	-

(c) Population size (MR=0.2,CR=0.8)

Figure 1: Various hyperparameters and their respective affect on performance

With the results in mind i can conclude the following

- High mutation rate creates an oscillation effect around the minima, worsening the performance, 0.2 seems to be the best choice
- Low crossover rate worsens the performance, as it enforces the algorithm to multuple extra generations to converge into the minima.
- High population size seems to increase the performance[1]

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).*

[1] When the metric used is 'Number of generations'