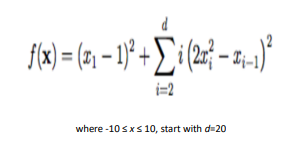
**Assignment Title**

**Student no. 23019866**

**1 INTRODUCTION**

Given two minimization functions and A math equations and formulas

Description automatically generated with medium confidence

Each function is implemented in an evolusionary algorithm which produces the lowest bestfitness depending on the parameters used. Withing my algorithm, I created functions, test\_function() where the minimization functon is implemented, fits() function to calculate the fitness using the test\_function() function, selection() funtion where the parents are randomly chosen from the population and there genes and fitness are copied into the offspring, crossovers() funtion where a random point in schosen in the ofsprings genes and swapped, and the mutation() function where the offspring genes are mutated. Various lists are created to save the fitness and genes of the population. I create a new variable G, which is generation and a list Gen, using a loop, the functions created earlier are called and used to calculate the bestfiness of the generation. The size of the generation changes depending on the number assigned to G, for each generation the bestfitness and average fitness is calculated and the smalles bestfitness value is chosen. The values of the bestfitness is what I used for this experiment and dtermine what produces the smallest bestfitness values.

**2 EXPERIMENTATION**

The aim of my Evolution Algorithm is to calculate the bestfitness value of each generation and find the smallest bestfitness. To calculate the bestfitness function, there are parameters included, MIN, MAX, MUTSTEP, MUTRATE, Population (P), Generation (G), Number of genes (N), Crosspoints. MIN, MAX and N stay constant, their constants differ with each function. Within this algorithm there’s a class (individual) within this class is a function that initialises gene and fitness. The test\_function() is where the minimisation function is implemented, this function is the used with the fits() function to calculate the fitness of the population and within the mutation function to calculate the new fitness value after the mutation. Within the selection() function two parents are selected and two offspring are created using the parents’ genes and fitness, which ever offspring has the smaller fitness values is saved into the offspring list. In the mutation function a random number is generated and if that number is less than the MUTRATE another variable is created, the variable is a random number generated between -MUTSTEP and MUSTEP, and then added to the gene, if the gene is greater than MAX the gene becomes MA, if its less than gene the gene becomes MIN, this process is done bit-by-bit and then the new individual is appended to the offspring list.

This functions are used in a loop to calculate the fitness values of the generation created and the values are added to the generation list.

Starting with the first function, A math equation with numbers and symbols

Description automatically generated with medium confidence

The constant parameters are N = 20, MIN = -10, and MAX = 10



Table 1.

With a decrease in population, the bestfitness values increase. The higher the population the higher the bestfitness values.



Table 2.

Same with the generation, the higher the generation size the lower the bestfitness values.

Giving the MUTRATE a starting values that is 1/N, I started testing out different values slightly higher or lower than 1/N which is 0.05.



Table 3.

Tested numbers higher than 0.05 and noticed the values increased the higher the MUTRATE values, then I tested lower values which resulted in smaller bestfitness values. The values increased when the bestfitness values was 0.002 and 0.001 which are number lower than 0.05, I concluded that MUTRATE values that are 0.01 or approximately 0.01 produces smaller bestfitness values.

For the MUTSTEP, my values for testing MUTSTEP centred around 10% of N, which is 2.



Table 4.

Numbers greater than 2 produced higher bestfitness values, the higher than 2 produced smaller fitness values but the further the values are from 2 the higher the bestfitness values. The smaller fitness values were produced when the MUTSTEP was 0.50, 1.00, 2.00, and 3.00. From that I concluded the smaller fitness are produces + or – 1.00 from 10% of N and the numbers approximately equal to them.

Next parameter is the Crosspoints, comparing the algorithm results when it has a single Crosspoint and when it has two Crosspoints,



Table 5.

The difference between them is very little, based on the table above, the lower the generation, two Crosspoints produce a smaller bestfitness value, but the higher the generation got a single Crosspoint produced smaller fitness values, although the differences are very small, and could be insignificant differences. A graph of a number of people

Description automatically generated

3 COMPARISON

Show comparative performance of other well-known approaches to the optimisation problems provided. You don’t need to implement other algorithms, use of open source, etc. is fine. For the very keen, other benchmark functions can be explored as well.

4 CONCLUSIONS

Concise summary of what you found and learned. Identification of ways you might do things differently next time, and why.

REFERENCES

Author, A. (1970) The best way to do AI ever. *IEEE Transactions on Data Mining* 1(1): 209-238.

**Source code as an appendix**