Name: Jonas de Deus Guterres (1301183615)
Class: IF-42-INT
Subject: Artificial Intelligence Observation Parallel Assignment 1

Report

1. **Observation of the Genetic Algorithm**

Genetic algorithm that I have built in the program is using python Language because it is easy to implement. The problem solve strategy that I use to solve this genetic algorithm is in the following steps:

- Chromosome Design and Decoding Technique:

  In the chromosome design I decide to use is the integer number as a value of the list and the value of chromosome will randomly choose with size of list which is 8. In the function of get chromosome will append to the list that will be encoded and calculated in the encoding genotype function.

  Decoding Technique is integer encoding that will take a half of the size of chromosome to calculate with the genotype formula:

$$x = r_{min} + \frac{r_{max} - r_{min}}{f \sum_{i=1}^{N} 9 * 10^{-i}} (g_1 * 10^{-1} + g_1 * 10^{-2} + \ldots + g_N * 10^{-N})$$

  So the number of range or limits of x1 and x2 that will calculate in the genotype formula are $-1 \leq x_1 \leq 2$ and $-1 \leq x_2 \leq 1$. After we got the result of the phenotype of x1 and x2, we will use the function h in the programming to find the fitness of the chromosome with –h because the purpose is finding the minimum value. The function that going to use is:

$$h(x_1, x_2) = \cos(x_1) \sin(x_2) - \frac{x_1}{(x_2^2 + 1)}$$

- Population size:

  The population size will be determine by the user because the population size can be change by user if the optimum value has not reached yet. In the function create population will be generated the population from the function of get chromosome before calculate the fitness of chromosome.

- Parent Selection Scheme:

  The method which has been used is tournament selection as long as it easiest to be implemented and fastest in the program. In parent selection there are two parents that will be selected from the population with the first and second highest in the fitness by the result of the encoding that implement in function best fitness. The tour size is 4 to make it faster.

- Genetic operation techniques and options (Crossover and Mutation), and Probability value in each Genetic operation (Pc and Pm):

  From the parent selection process and the result, crossover will be proceeded by using the probability less than equal 70% to exchange the chromosome. In the function of crossover, I used random number of getting the probability number with condition less than 0.71 to generate the children 1 and children 2 by swapping. There is two point that will be random also to make a boundaries before making the crossover. At the end it will return the children that will proceed to the mutation.

Name: Jonas de Deus Guterres (1301183615)
Class: IF-42-INT
Subject: Artificial Intelligence Observation Parallel Assignment 1

In the mutation process, the result of the crossover will be mutated using probability less than equal 1%. In the function of mutation, there is similar way to crossover to get the random probability with condition less than equal 0.01. So the random index number of child will assign the new random number random number.

- New Generation Selection technique (Survivor Selection) and Stopping Criteria (Generation termination):

  In the selection technique is using generational replacement for generating the new generation. In this function, there is list of new generation that will append the best fitness of chromosome and it used for the parent selection as well. The condition will repeat of creating the new generation based on the size of population in twice (I+=2). The process start from parent selection, crossover and mutation that I have explain in above. After the mutation the children will added to new generation for the next process.

  The stopping criteria will be inputted by the user until they get the optimum value. As I implemented in the program is 500 for stopping criteria and it will change from generation 20 to 300, and from 300 to 500 is stable with the optimum value.

2. Genetic Algorithm Parameter: (My case to get Optimum value)

- Chromosome Design : Integer
- Decoding Technique : Integer Encoding Genotype
- Chromosome Size : 8
- Population Size : 50
- Parent Selection scheme : Tournament Selection
- Crossover : two-point, 70% probability
- Mutation : 1% probability
- New Generation Selection technique : Generational Replacement
- Stopping Criteria (Generation termination) : 500 generations

3. screenshot of the best running results based on the optimum parameter values

```
    population = gen_replacement(population,bf_chrom,size_chrom)
print("\n\nThe last Generation")
print("************************")
print("Best Chromossome:",population[bf_chrom])
print("Fitness: ",best_fitness)
print("X1,x2: ",encodingGenotype(population[bf_chrom],size_chrom))
```

```
Enter the number of population: 50
Enter the number of generation: 400
 {Generation - 400 | Best Chromossome:  [9, 9, 9, 9, 5, 0, 0, 0] | Best Fitness:  199960001.0003731 | Decode Chromosome (x1,x2): [2.0, 0.0001000100010001711] }

The last Generation
************************
Best Chromossome: [9, 9, 9, 9, 5, 0, 0, 0]
Fitness:  199960001.0003731
X1,x2:  [2.0, 0.0001000100010001711]
```

Explanation: The output will be in every generation in a tuple that will display the best chromosome, fitness and decode x1 and x2 with result 2.0 and 0.0 respectively.