ASSIGNMENT-3

Subject: Linked list implementation

Advisor: Res. Assist. Selim YILMAZ

Due Date:01.01.2020 – 23:59:59

Student: Mucahit Veli CUMART -- 2160589

INTRODUCTION

In this experiment , we will design an implementation for Generic Algorithm ,or GA,is a well-known evolutionary-based heuristic search algorithm used in artificial intelligence (AI) and computing.

GA makes use of a population which is composed of individuals, also called chromosome, to solve both constrained and unconstrained optimization problem. Each chromosome is composed of genes, the smallest unit containing a genetic value.

GA is an iterative algorithm and, at every step, it employs generic operators involving selection, crossover, and mutation to evolve new, hopefully better, individuals that represent the solution toward the problem at hand. The quality of each evolved chromosome is evaluated through a given fitness function. At the end of the evolutionary process, it returns the best solution found for the problem.

FIVE STEPS

***INITIALIZATION:** Simply creates a new population.

Functions of First Step

-void Append(): This function is useful for push the genes to linked list of chromosome.

-void Append_dna(): This function is useful for push the chromosomes to linked list of population.

*

****Fitness evaluation:** Reveals how well a chromosome represents a solution toward a problem.

Functions of Second Step

-long fitness_eva(): This function is useful for evaluation of rank of chromosomes.

*****Selection:** Selects two chromosomes as parent with respect to some rules:

******Crossover:** Combines the selected parent to evolve offspring for next generation.

Functions of Fourth step

-void Xover(): This function is useful for combining the selected parent to evolve offspring for next generation.

*******Mutation**: Applies a change on genes of evolved offspring.

Function of Fifth Step

-void Mutate(): This function is useful for appliying a change on genes of evolved offspring.

The other Functions of my Implementation

- İnt power(): This function is for converting binary to decimal.
- Float rank_eva(): This function is for evaluating ranks of chromosomes.
- Void display(): This function is for displaying genes of a chromosomes.
- Vodi display_populations(): This function is for displaying all of the chromosomes in the Population.
- Void findBestChromosome():This function is for finding best chromosome in the population.
- Void findWorstChromosome(): This function is for finding worst chromosome in the population.
- Void BubbleSort(): This function is for sorting the chromosomes.
- Void swap(): This function is for changing the features of two chromosomes each other:

ALGORİTHM OF MY IMPLEMENTATION

Read command line arguments;

Load all the parameters including selection, mutation, crossover;

Initialize population w.r.t. population file;

Evaluate the fitness and ranking of each chromosome

Sort the population in ascending order;

Store the best chromosome;

Print the population with the best chromosome found so far;

Repeat

- Apply selection;
- Apply crossover;
- Apply mutation;
- Evaluate the fitness and ranking of each chromosome
- Sort the population in ascending order;
- Update the best chromosome if found better;
- Print the population with the best chromosome found so far;

```
until MAX GEN is reached;
```