Nile University

ECEN-501: Machine Intelligence

Professor: Bassem Mokhtar

Genetic Algorithm Report

Fall 2018

Submitted By:
Mohamed Ayman 1510162

i.Contents

ii.List of Figures

iii.Knapsack Problem

The knapsack problem briefly is a collection of N object with various weights and scores, and you need to fit elements in the sack with maximum score and not exceeding the maximum weight. Moreover, in the proposed problem you need to get least weighted car parts with maximum price and least weight.

a.Object List

Firstly, to solve the knapsack problem, a problem scope should be chosen first. Therefore, I chose the field of car parts, furthermore, there is a csv file -proposedProblem.csv -located in the same directory that has a list of all objects used with their respective weight and price.

b.Fitness Function

Basically, the fitness function is used to evaluate each chromosome in the population, the chromosome is a string of 0s and 1s, where 1 means the object is taken into the knapsack. Moreover, each object chosen has a specific weight and price. And the objective of the fitness function to evaluate 0 for overweighed objects and evaluate the price otherwise. For example, if the maximum weight is 50 and the chosen object exceed this limit, the fitness function will output 0.

iv.Implementation & Discussion

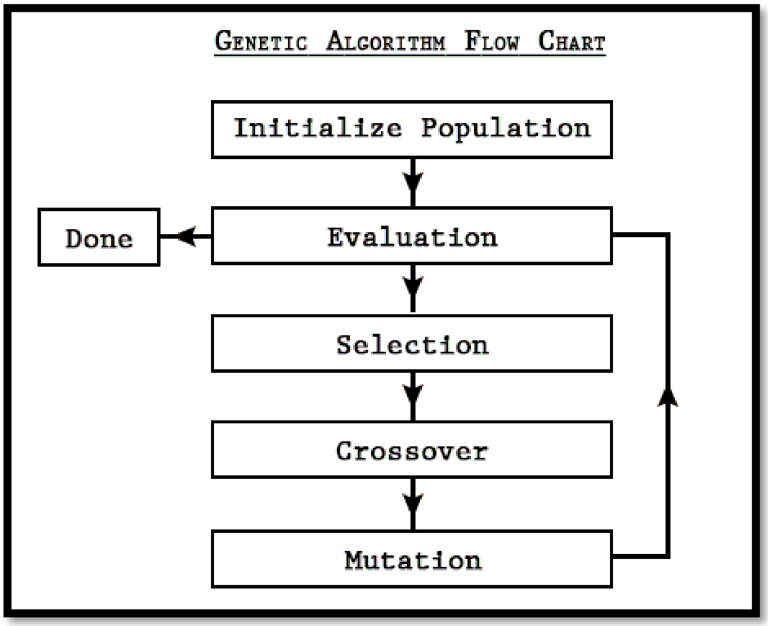

Figure 1: GA Flow Chart

a.Introduction

A genetic algorithm tries to reach a global optimum by a mean of the above algorithm. Firstly, initialize a population to start off, secondly evaluate the fitness of each member, then select 2 random parents for the crossover procedure, crossover between the two parents to obtain 2 children, then a mutation with 0.1% chance and finally these steps keep happening until the number of iterations are over.

b.User Mode

The proposed problem by the developer is found in the csv file mentioned earlier, but if a user would like to choose his own parameters, all he must do is change the definition value found in the beginning of the program.

c. Hyperparameters

Number of Objects

The number of objects used were 8, as to prove a proof of concept of the algorithm with a moderate number of objects for testing and validation.

Population Size

As the number of objects used were 8 and the total combinations are 28, thus 8 is a divisible of 256 and will span the whole set of combinations in a short time.

Number of Iterations

The number of iterations chosen was 500000, as by trial and error the dataset usually converges in this range.

Maximum Weight

Maximum weight chosen was 33, as it is bigger than the maximum weighed element by a bit, which makes the squeeze harder and makes the problem more complex.