Sememster Project Report

Time Table Scheduling using Genetic Algorithms in Artificial Intelligence

Muhammad Faisal
i14-0154-E
i14-0154@nu.edu.pk

Abstract

Almost all education institutes have problem concerning with scheduling, especially university. Many things have to be considered in order to arrange schedule. One of them is availability of lecturers. Not all lecturers are available at any time. Some of them are just available in some time. Therefore, when schedule is arranged, this thing has to be considered. The other things are number of classes and courses offered. Number of classes and courses in university timetable are many. Room availability is other thing, budgeting and many others.
A **genetic algorithm** is a search heuristic that is inspired by Charles Darwins theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.
Genetic Algorithms have two parts **Selection and Population**. The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found. In this project, we will implement the Genetic Algorithm to solve the process of Time Table Scheduling in a compact way!

Keywords Genetic Algorithm, Scheduling, Population, Selection, fitness

1. Introduction

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.
Optimization is the process of making something better. In any process, we have a set of inputs and a set of outputs. Optimization refers to finding the values of inputs in such a way that we get the best output values. The definition of best varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.

2. Related work and Failures

Traditional calculus based methods work by starting at a random point and by moving in the direction of the gradient, till we reach the top of the hill. This technique is efficient and works very well for single-peaked objective functions like the cost function in linear regression. But, in most real-world situations,like Scheduling a time table, which include limited Teachers, Courses, and Rooms of an intitution, which causes such methods to fail, as they suffer from an inherent tendency of getting stuck at the local optima as shown in Figure 1 above.

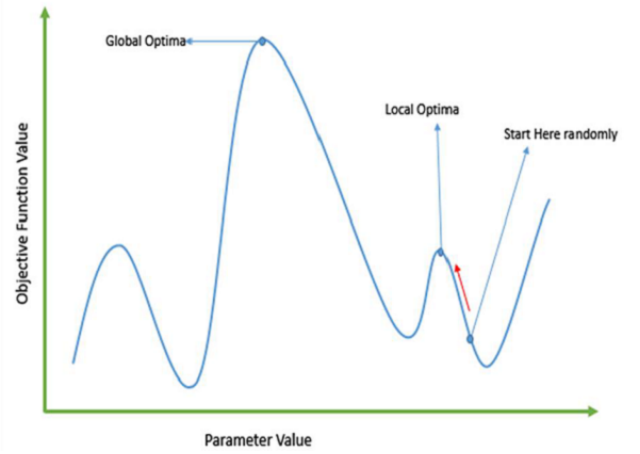

Figure 1. Local Maxima Optima Graph

Some difficult problems like the Travelling Salesperson Problem (TSP), have real-world applications like path finding and VLSI Design. Now imagine that you are using your GPS Navigation system, and it takes a few minutes (or even a few hours) to compute the optimal path from the source to destination. Delay in such real world applications is not acceptable and therefore a good-enough solution, which is delivered fast is what is required.

3. My Approach:Genetic Algorithms

Some of the terminology we used in this project, are as follows.

- **Population** It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings.
- **Chromosomes** A chromosome is one such solution to the given problem.
- **Gene** A gene is one element position of a chromosome.
- **Fitness Function** A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem.

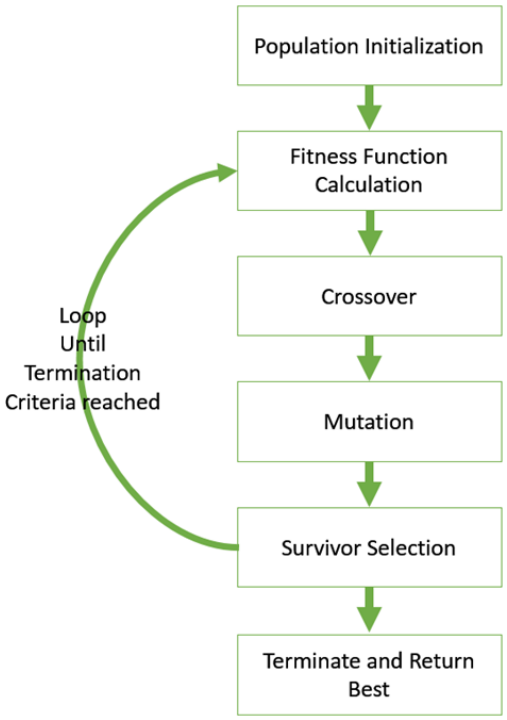

Figure 2. Basic Genetic Algorithm working

- **Population and Selection** Population and selection are randomly chosen values. Firstly some random values are picked and handed over to the gene to make a solution. Selection then works to use the fittest gene for further process.

A population having P chromosomes called individuals is generated by pseudo random generators whose individuals represent a feasible solution. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space. The whole Process is described in Figure 2 .
Individual members and chromosomes of the population represented by a string are evaluated to find the objective function value. This is exclusively problem specification. The objective function is mapped into a fitness function that computes a fitness value for each chromosome. This is followed by the application of GA operators OR Models. I have used a reproductive/generation Model.
Reproduction or generation is usually the first operator applied on a population. It is an operator that makes more copies of better chromosomes in a new population. Thus in reproduction operation the process of natural selection causes those chromosomes that encode successful structures to produce copies more frequently. To sustain the generation of a new population the reproduction of the chromosomes in the current population is necessary. For better chromosomes these should be generated from the fittest chromosomes of the previous population. Thats why **Initialize population** function returns the fittest chromosomes to **GA** function.
A generalized working of crossover function is shown in Figure 3 . These values of generations are passed to the crossover function for further processing. A crossover function is used to recombine two chromosomes to get a better one. In the crossover operation, recombination process creates different chromosomes in the successive generations by combining material from two chromosomes of the previous generation. In reproduction, good chromosomes in a population are probabilistically assigned a larger number of copies

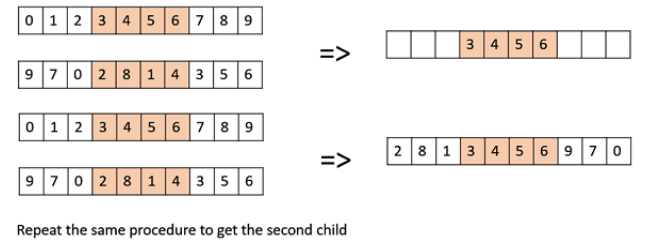

Figure 3. Cross over function working

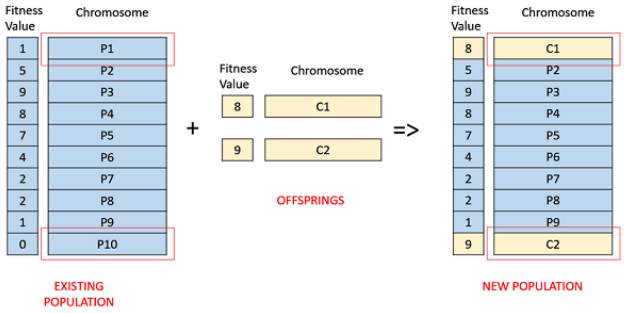

Figure 4. Cross over function working

and a mating pool is formed. It is important to note that no new chromosomes are usually formed in the reproduction phase. In the crossover operator, new chromosomes are created by exchanging information among strings of the mating pool.
P1 and **P2** are passed to the **crossover** function.

Mutation function adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes to be different from those of their parent. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level. When the bits are being copied from the current string to the new chromosomes, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability. The need for mutation is to create a point in the neighborhood of the current point. The mutation is also used to maintain diversity in the population. The Mutation function general working is shown in Figure 4.

4. Fitness Evaluation

Fitness Evaluation criteria for this program is easy. At the end of mutation, the mutated groups are passed to fitness function for evaluation. At first every group has 0 fitness. Then keeping the view of limited space, we try to find out which of the population group have performed best. And by performing best, I mean that weather or not they pass almost all of the constraints mentioned in Statement. Some of the things that include in calculating fitness is teacher preference, free days taken, batch groups, and duration do not conflict. if they have conflicts, the fitness value is subtracted from tunable variable. For mine it is 10. And at last the system return the group solution which have highest score.