Timetable Scheduling using Genetic Algorithm

Submitted by:

Name: Nimra Iqbal **Student ID: Sp17-bsse-0055**

Name: Abdullah Waseem **Student ID: Sp17-bsse-0060**

Name: Sheikh Haris Khalid **Student ID: Sp17-bsse-0046**

Section: A(M)

Submitted to: Sir Kashif Khan

Date: 10 Jan 2020

DEPARTMENT OF COMPUTER SCIENCE

MUHAMMAD ALI JINNAH UNIVERSITY

22-E BLOCK 6, PECHS, KARACHI-PAKISTAN

Index

Acknowledgement

We would like to thank, Sir Kashif khan for teaching us Artificial Intelligence and expanding our domain knowledge. We would always be grateful of him giving us the opportunities and solving our queries.

We would also like to thank Dr.Khitab Gul, teacher of bio-science for helping us and make us understand about the things clearly.

We have done a lot of effort in researching the things related to the project and this couldn’t be done without the universities resources.

Abstract

Day by day the field of Artificial intelligence is expanding, everyday new things are being introduced in this field. If we see closely around ourselves, we see Artificial intelligence is being implemented everywhere either in educational sector or in the hospital. Especially in educational sectors AI performs a great roll either in bio-metric attendance, online quiz or arrangement of classes. Back in the days they were solves this things manually with taken lots of time and efforts.

In this report, we have work on the knowledge that we have done to make class scheduling that will sort out the conflicts of teacher and student and give them the optimal solution.

Introduction:

Problem statement: "allocation of classes and lessons per week at a time, allocating courses to teacher and timing issue was the great challenge to do. Solving this manually was too haptic because it take so much time and effort and in the end the result is full of conflicts**"**

How do we solve it?

In the issue of tabulation of courses, in principles includes allocating lessons per week holding the periods and room class, if conditions such as failure to allocate one class at a time to several lessons or lack of time in courses of a teacher are added to this issue, it becomes clear that the problem of designing this time is CSP (constraints satisfaction problem). Before going to further CSP lies down in the Variable Based Model because we need to holds the value of our previous and current state for the product and calling it Variable because it can vary the values.

Before AI came into being it was done manually but after AI it has been done with different methods such as Graph coloring problem, use of heuristic functions or ant colony algorithm but they failed for not giving the optimal solution. For example: Graph coloring problem it failed because number of the least possible colors to be used for GCP is called chromatic number. As the number of vertices or edges in a graph increases, the complexity of the problem also increases. Because of this, each algorithm cannot find the chromatic number of the problems and may also be different in their executing times.

Ant colony algorithm:

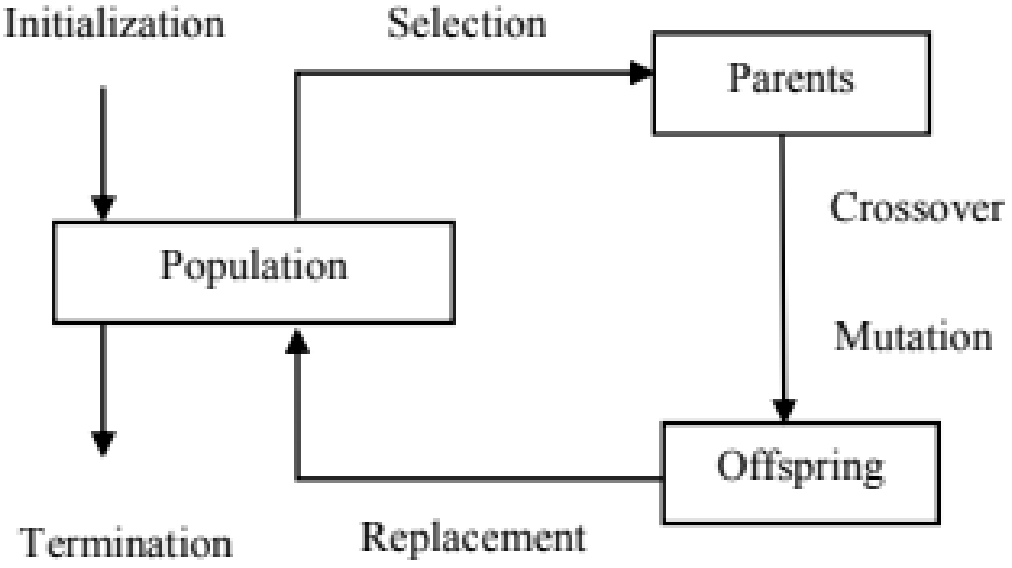
Probability distribution changes every time, difficult theoretical analysis, dependent sequences of random decisions more experimental than theoretical research uncertain time to convergence.

In this Project we are going to solve the Class scheduling problem with the Genetic Algorithm. It is one of the strongest and most widely used algorithm in search and optimization problem, one of the reason of its popularity it is it does not require any high level mathematical model all you need is to know the little background of the Genetics. Genetic algorithm idea is taking by the nature or law of development. It produces the results that best fits in the environment. It starts with some set of data and then it start shuffling, and by every time shuffle it's getting better than before until it reaches the global optimum level.

Decade shows that genetic algorithm is the strongest method inspired from the nature of genetics and natural selection phenomenon which the best form of optimization problem in science and engineering. The components of this algorithm is like natural such as chromosomes, genes genetic population fitness function and genetic operators.

Genetic Algorithm	**Natural Algorithm**
Variables	Chromosome
Fitness Function	Environment
Breeding	Natural selection
Crossover	Proliferation
Mutation	Genetic mutation
Result	Reproduction

Flow of Genetic Algorithm:



How Genetic Algorithm Works?

Genetic Algorithm based on the ideas of natural selection and genetics. Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate **"survival of the fittest"** among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and everyone represents a point in search space and possible solution. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Foundation:

It is based on 4 steps:

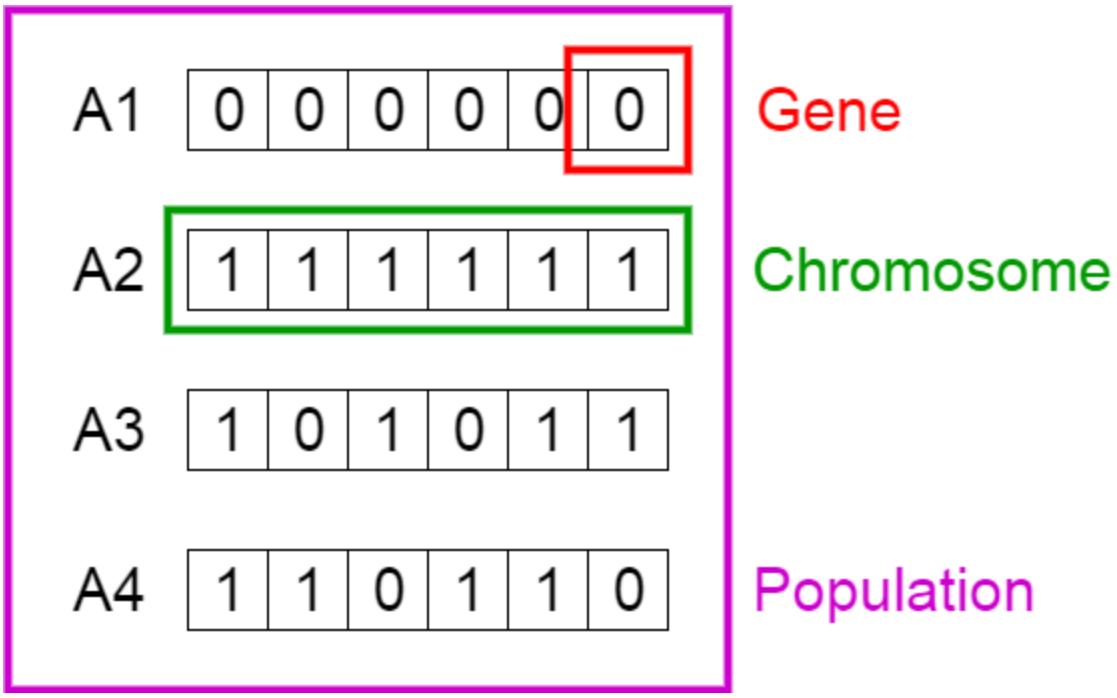
- Individual in population compete for resource and mate.
- Those individuals that are fittest then mate to create more offspring.
- Genes from fittest parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
- Thus, each successive generation is more suited for environment**.**

Five phases are considered in a genetic algorithm:

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation

Search Space:

The population of individuals are maintained within search space. Everyone represents a solution in search space for given problem. First, we create individuals and then we group them and call Population. An individual is distinguished by set of variables known as Genes. These Genes are combined into a string to form Chromosome, which is basically the solution.



Fitness Function:

Fitness function determines the fitness of individuals, in general terms, its ability to compete with other individuals and be the final desired output. It gives fitness scores to everyone and the probability that an individual will be selected for reproduction is dependent on its fitness score. This way, individual with higher fitness score is a specimen genetically closer to success than the others. Our fitness function will precisely sort the population.

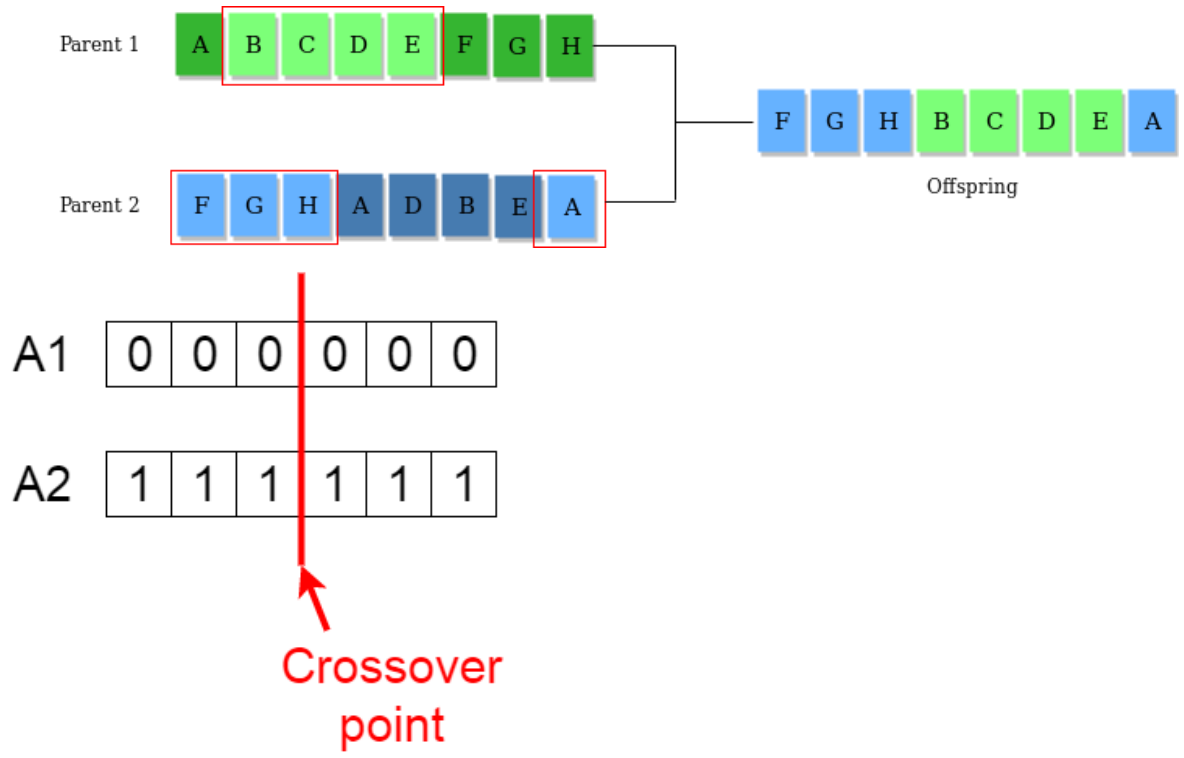
Selection:

As we now have the fitness score, in this phase we select the individuals with the highest scores and let them pass their genes to the next generations. After selecting individuals, we now group them into a pair of two(parents) based on their fitness score because obviously it's not possible to reproduce without two entities.

There are many ways to do this, however; we should keep two things in our mind. Firstly, to generate the best solution of the previous generation but not keeping others aside. And secondly, also the hazard, if you select only good solutions at the beginning, you will quickly reach towards the local minimum, which is the limitation of Genetic Algorithms, and not towards the best solution.

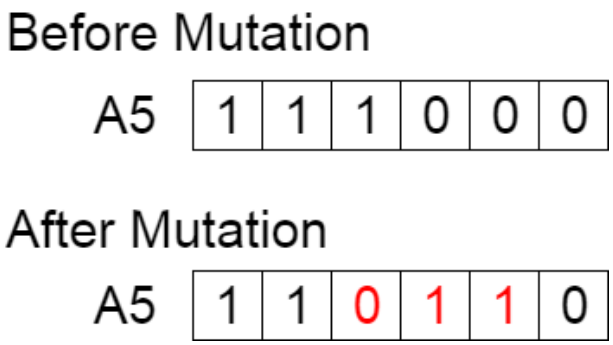
Crossover:

The goal of reproduction is to mix the DNA of two individuals. So, we will do the same thing here. Let's take two individuals, their DNA is defined by their alleles (value of each letter). Therefore, in order to mix their DNA, we just must mix their letters.



Mutation:

The primary goal of this step is to prevent our algorithm to be blocked in a local minimum. After the crossover, each individual must have a small probability to see change in their DNA.



Algorithm:

1) Randomly initialize populations p

2) Determine fitness of population

3) Until convergence repeat:

   a) Select parents from population

   b) Crossover and generate new population

   c) Perform mutation on new population

   d) Calculate fitness for new population

Advantages:

- They are Robust
- Provide optimization over large space state.
- Unlike traditional AI, they do not break on slight change in input or presence of noise

Application of Genetic Algorithms

Genetic algorithms have many applications, some of them are:

- Recurrent Neural Network
- Mutation testing
- Code breaking
- Filtering and signal processing

References:

1. https://mafiadoc.com/automatic-generation-of-university-course-_5bae258a097c47e0058b473b.html

2. https://www.researchgate.net/publication/273776640_Class_Timetable_Scheduling_with_Genetic_Algorithm

3. https://pdfs.semanticscholar.org/8f8f/cf040b50ea7c76eb45042aa6aaa98b29ee49.pdf

4. https://www.intechopen.com/books/real-world-applications-of-genetic-algorithms/solving-timetable-problem-by-genetic-algorithm-and-heuristic-search-case-study-universitas-pelita-ha

5. https://www.academia.edu/8306247/University_Timetable_Automation_using_Genetic_Algorithm

6. https://en.wikipedia.org/wiki/Genetic_algorithm

7. https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications

8. https://towardsdatascience.com/genetic-algorithm-explained-step-by-step-65358abe2bf