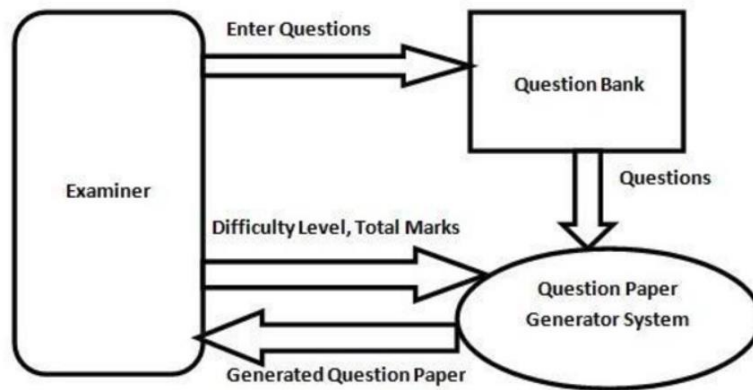


Artificial Intelligence Mini Project



Atharva Marathe

MIS: 111903024

Batch: Div1 T2

maratheas19.comp@coep.ac.in

Avishkar Andhale

MIS: 111903025

Batch: Div1 T2

andhalear19.comp@coep.ac.in

Darshan Shah

MIS: 111903031

Batch: Div1 T2

shahdh19.comp@coep.ac.in

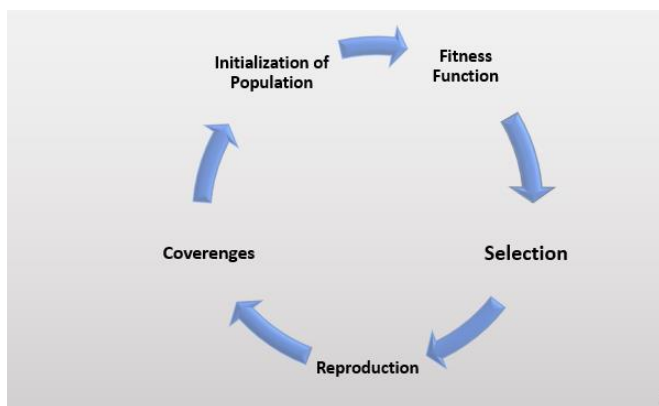
Proposed Topic:

Automated Paper Generation Using Genetic Algorithm

Introduction:

Examination plays an important role in teaching, paperless automation test becoming increasingly prevalent in all walks of life, rich in questions, and Intelligent test paper generation is one of the key steps to implement automated test. [2]

Intelligent test paper auto-generating is of a high-quality paper which satisfy multiple constraints set by professor after selecting questions from the bank of questions. Some algorithms have been considered, such as select-random algorithm, backtrack algorithm, and genetic algorithm. However, they consuming lots of time and space in the condition of much questions. Genetic algorithm is a natural evolutionary process simulation model which have the characteristics of concurrency, intelligent search, robust, simplicity. [2]



Genetic Algorithm

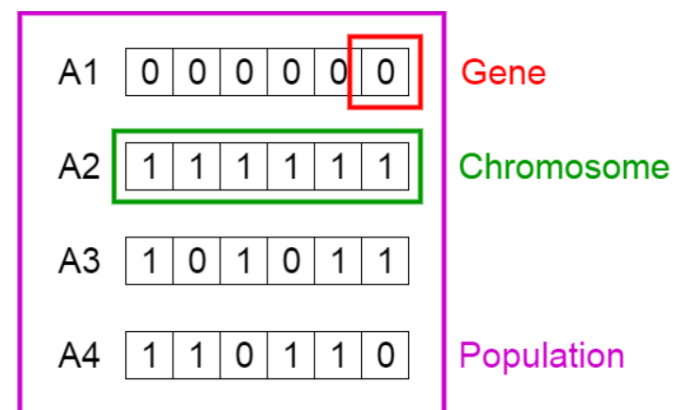
A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.[7]

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 better chance at surviving.

This process keeps on iterating and at the end, a generation with the fittest individuals will be found. This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.[7]

Five phases are considered in a genetic algorithm.

- ❖ Initial Population
- ❖ Fitness Function
- ❖ Selection
- ❖ Crossover
- ❖ Mutation



Dataset:

The dataset required for the question paper generator is huge number of random questions. As the algorithm primarily works on the constraints given on the question paper, which are numbers, a computer program can be written for producing these questions. So as a result, there is no specific requirement to import questions database from any external source.

Initial Population

The process begins with a set of individuals which is called a Population. Each individual is a solution to the problem you want to solve.

An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution).[7]

Fitness Function

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.[7]

Selection

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Cross-Over

Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes.

Mutation

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped.[7]

Pseudo-code

START

Generate the initial Population

Compute fitness

REPEAT

Selection

Crossover

Mutation

Compute Fitness

STOP when Population converged.

Project Feasibility:

Need of project:

Question papers are the most prevalent method for assessing a student's academical excellence. Designing them correctly hence is one of the most crucial steps in our education system. Curating question papers for subjects having very large number of topics or having many subjects, is a very tedious and time-consuming task.

Constraints on Question Paper: -

- 1) No. of marks for each question
- 2) Total number of questions
- 3) Overall Difficulty of Exam
- 4) Course Outcomes
- 5) Chapter Distribution

Significance of Project:

All constraints are needed to be satisfied. In addition to it, as it a MCQ based exam, to ensure no malpractices are used questions in each paper should be randomised and the overall difficulty of the exam should be the same for each student (In addition to the constraints mentioned above).

This type of question paper generator is particularly helpful during the design of question papers of competitive examinations such as GATE, JEE and others. The size of the question bank available is in thousands, the number of chapters are also large in number. In such type of situations designing randomized question papers for thousands of students while ensuring that the overall difficulty remains the same requires efficient Computer algorithms to generate the papers which should resemble the given structure.

Feasibility of project:

As the data can be generated using computer programs there is no specific requirement for external dataset for designing the algorithm. Also, for GUI development the packages and dependencies required are available on the internet.

Literature Survey:

The use of genetic algorithms (GA) for problem solving is not new. The pioneering work of J. H. Holland in the 1970's proved to be a significant contribution for scientific and engineering applications. Since then, the output of research work in this field has grown exponentially although the contributions have been, and are largely initiated, from academic institutions world-wide.[5]

GA is inspired by the mechanism of natural selection, a biological process in which stronger individuals are likely to be the winners in a competing environment, here, GA uses a direct analogy of such natural evolution. It presumes that the potential solution of a problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as fitness value, is used to reflect the degree of "goodness" of the chromosome for solving the problem, and this value is closely related to its objective value.[5]

According to the basic idea of the genetic algorithm, the test paper generating algorithm which can solve the multiple constraints combination optimization problem in the test paper generating, and with a high success rate of test paper generating in a relative short time has been designed. There are a lot of researches in this area.

The paper: *Application of Improved Genetic Algorithm in Automatic Test Paper Generation* Kui Zhang [6] states that "Automatic test paper generation problem is a multi-objective optimization problem under certain conditions. Its constraint conditions cannot be described in mathematical form, so it is very difficult to use the traditional mathematical method to solve it. It can be proved that GA is one of the best approaches to dealing with this problem efficiently." Hence Genetic algorithm and its variants is proved to be very useful in the field of automatic paper generation.

Research Gaps:

At present, the commonly used method of test paper generating has the following three kinds.

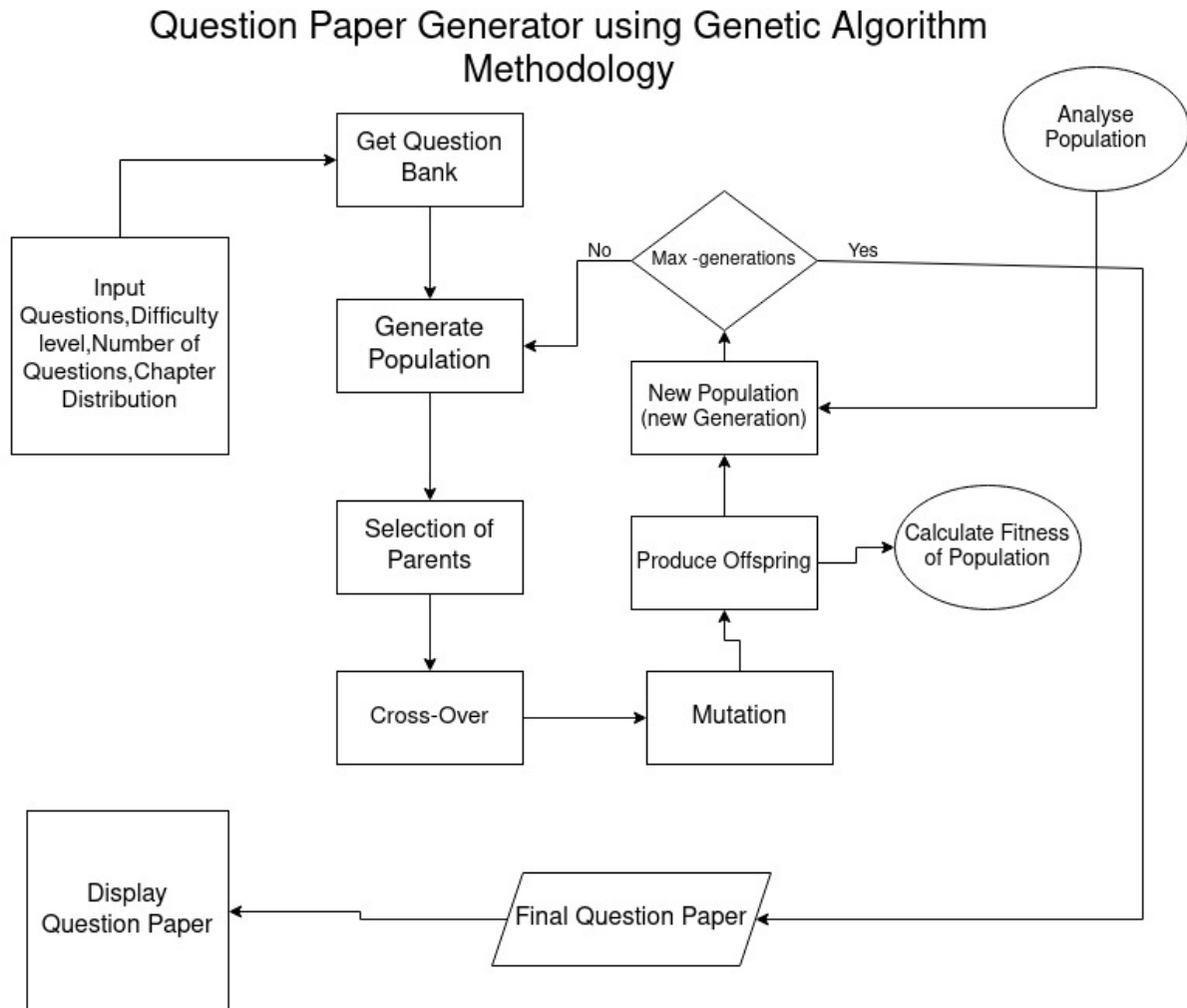
Random extraction method: In the process of the test paper generation, one question is selected randomly from the Item bank according to the current constraint, check the questions whether meet the constraints of the objective function, if not satisfied, then choose a new question for new trial, until the test paper that meet the user's requirement was generated. If unable to produce a test paper to meet user demand, the test paper generation process failure. The research in this area can be found in the literature [4]

Retrospective testing method: The states of each step of the random selection method are recorded. When the search fails, release the state of the last record, then transform to a new state according to a certain rule. Through continuous retrospective testing until the test generation is finished or returned to the start point. If the state and question quantity are less, the success rate of this conditional depth-first algorithm is better.[4]

Backtracking algorithm can reduce the probability of invalid exam questions, but it faces the same problem as the random algorithm which is time consuming to search or backtracking when the questions volumes is huge. Therefore, researchers started to adopt artificial intelligence in their researches to improve on the performance of Automated Exam Questions Generator and the quality of exam questions. Examples of the artificial intelligence algorithm are ant colony algorithm, simulated annealing algorithm and Genetic Algorithm. [3]

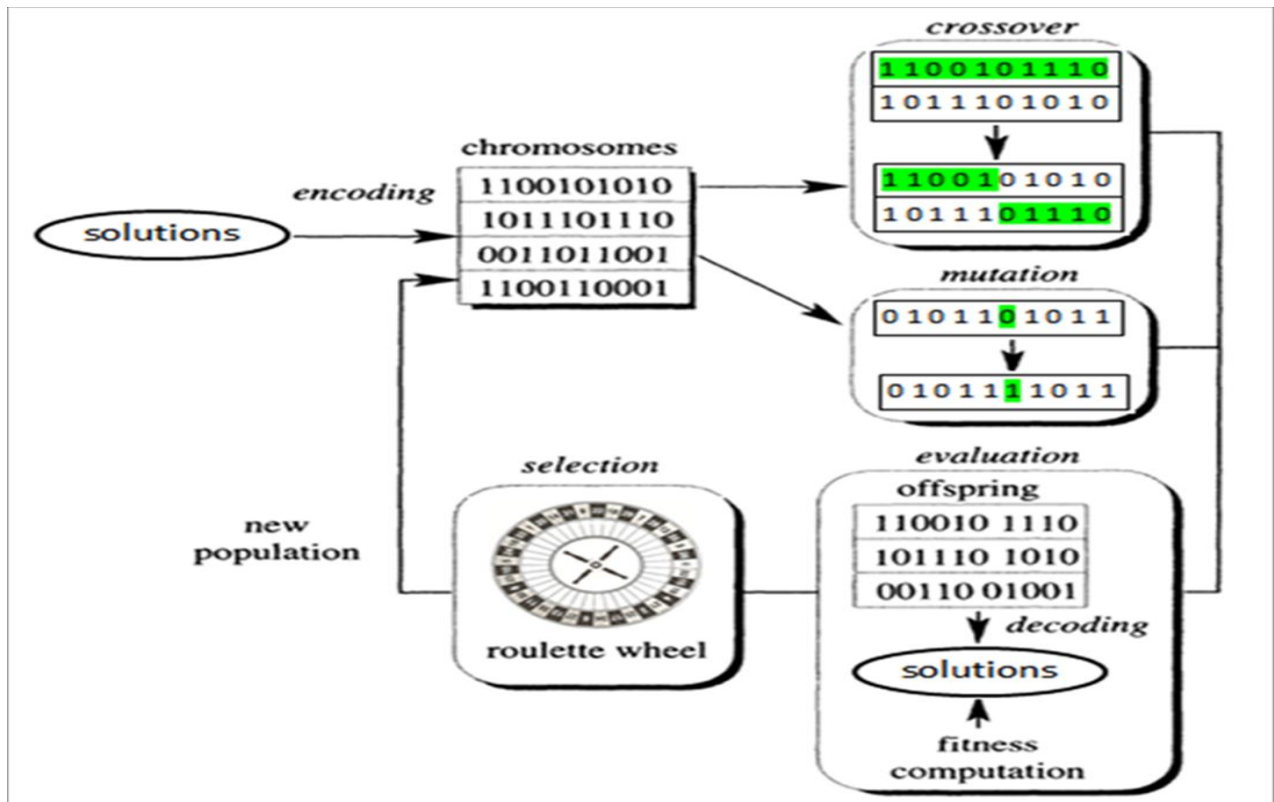
But for the practical application, the randomness of questions selected is lack, and the test paper generation time is too long. The memory usage of this method is relatively large, and the program structure is relatively complex. However, retrospective testing can guarantee that a feasible solution can be found as long as the solutions to meet the conditions existing in the item bank.[4]

Methodology /Planning of work:



- Generating dataset required for designing and testing the algorithm.
- Making population and selection of parents for creating new generations.
- Exploring the fitness functions given in literature and designing a custom version of it suiting the needs of our application.
- Iteratively generating several generations of the population and analysing the results of them.

- Creating front-end GUI which can be used with our model in-order to use it for practical application.



Facilities for proposed work:

1. JavaScript
2. React (for Front-end GUI)
3. MongoDB (as database of questions)
4. NodeJS
5. Dependencies required for the above packages.

Timeline:

Week	Work
Week 1	Designing the Question bank generator
Week 2	Generating population and cross-over Of parents
Week 3	Designing the fitness function and analysis on new generations
Week 4	Testing and further analysis of question papers generated
Week 5	GUI development for the application

Bibliography:

- [1] Huang Ming, Tang Ling, Liang Xu “Research on intelligent test paper generation based on improved genetic algorithm”
<https://ieeexplore.ieee.org/document/5191822/>
- [2] An Improved Genetic Algorithm for Intelligent Test Paper Generation:
<https://ieeexplore.ieee.org/document/7003488>
- [3] Tengku Nurulhuda Tengku Abd Rahim “Automated Exam Question Generator using Genetic Algorithm “IEEE Conference on e-learning, e-management and e-services 2017
- [4] Song Yan, Yang Guoxing “A Genetic Algorithm of Test Paper Generation” The 8th International Conference on Computer Science & Education (ICCSE 2013)
- [5] K. F. Man, K. S. Tang and S. Kwong “Genetic Algorithm: Concept and Applications” IEEE Transactions on Industrial Electronics Vol 43

[6] Kui Zhang, Lingchen Zhu "Application of Improved Genetic Algorithm in Automatic Test Paper Generation" IEEE 2015

[7]<https://towardsdatascience.com/introduction-to-genetic-algorithms-includingexample-code-e396e98d8bf3>

[8] Genetic Algorithms: The Nature of Code, Coding Train
<https://youtu.be/9zfeTw-uFCw>

[9] <https://www.sciencedirect.com/topics/engineering/genetic-algorithm>