**CHAPTER 2: LITERATURE SURVEY**

**2.1 Survey Existing System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No. | Paper Title | Authors | Summary | Conclusions |
| 1 | Solving Timetable Problem by Genetic  Algorithm and Heuristic Search Case Study:  Universitas Pelita Harapan Timetable | Samuel Lukas, Arnold Aribowo and Milyandreana Muchri | Steps involved in GA are thoroughly explained in this paper. The operations performed in each step that is Selection, Crossover and Mutation are given in detail.  Input format is presented well in the paper. The target matrix is the data stricture used for input. | Soft Constraints are not included in the scope along with constraints of classrooms. |
| 2 | Electronic Lecture Time-table Scheduler using Genetic Algorithm | Jumoke Soyemi, John Akinode and Samson Oloruntoba | Compares MTTS with GA, and shows flowchart of how GA is implemented | Preference of the University to use Automatic Scheduling with GA. However Soft constraints are not taken care of. |
| 3 | Solving Timetable Scheduling  Problem  Using Genetic Algorithms | Branimir Sigl, Marin Golub and Vedran Mornar | Shows formation of Chromosomes and Genes which are used as the basis for GA | Alternate approach to the data structure being used to optimised. |
| 4 | Solving Timetabling Problem Using Genetic and Heuristic Algorithms | Nguyen Duc Thanh | Incorporates the use of a hybrid algorithm consisting of Genetic and Heuristic Algorithm. Using this Algorithm Timetabling problem is converted into a way of optimally solving a 2D Matrix. | Crossover and Mutation operations of the given system will be used. |
| 5 | Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem | Theodore W. Manikas and James T. Cain | A simplified model of the placement problem circuit partitioning was tested on three circuits with both a genetic algorithm and a simulated annealing algorithm When compared with simulated annealing the genetic algorithm was found to produce similar results for one circuit and better results for the other two circuits | The study concluded its findings based on a very small dataset rather than considering a large data set to give an approximate percentage as on how much is genetic algorithm more efficient than simulated annealing |

**Summary and Conclusions of Surveyed Papers:**

Table 2.1 Summary and Conclusions of Surveyed Papers

**Detailed Summary of the Surveyed Papers:**

1. Samuel Lukas, Arnold Aribowo and Milyandreana Muchri, “*Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable*”:
   1. Methodology employed:

The paper describes in detail the Steps involved in GA. The operations performed in each step that is Selection, Crossover and Mutation are given in detail. The Architecture Design of the System makes use of a Target Matrix. Six sets are applied to this Matrix. The description of each set are as follows:

1. Course Code set **M** = {M1, M2, …}
2. Type Course Class set **T** = {T0, T1, …}
3. Lecturer Code set **L** = {L1, L2, …}
4. Class Name set **C** = {C1, C2, …}
5. Day set **D** = {D1, D2, …}
6. Hour set **H** = {H1, H2, …}

The columns of this matrix are made up by the close relationship of Li, Mj and Tk which indicate Lecturer ‘i’ teaches Course ‘j’ of the type ‘k’.

The rows are made of the relationship between Dx, Hy and Cz which indicate Slot on Day ‘x’ at Hour ‘y’ with Class ‘z’

Each combination between the row and column indicates a unit of the timetable, i.e., for Target Matrix ‘V’, Vij being the element on ith row and jth column gives us d, h, c, l, m, t which is on day ‘d’ at hour ‘h’ for class ‘c’, Lectures ‘l’ teaches course ‘m’ of type ‘t’.

Genetic Algorithm steps as mentioned above are then applied to this Target Matrix.

* 1. Result Achieved:

An experimental setup for 7 courses and 10 lecturers with 2 type course classes (lecture and lab), 2 class names (A and B), 5 days lecture a week and 10 hours lecture a day. The system ran with 10 chromosomes and 10 generations are set in the experiment.

An error free Schedule was achieved.

* 1. Scope for future improvement:

Room set has not been explicitly defined in the target matrix; however, the system is still able to manage and allot rooms. The author/s note that if room plays a key role in the scheduling, then it should be included in the target matrix.

* 1. Platform Dependency and Experimental Setup requirement:

The proposed system has no platform dependency and the setup requirement would be based on the technology used.

1. Jumoke Soyemi, John Akinode and Samson Oloruntoba, *“Electronic Lecture Time-table Scheduler using Genetic Algorithm”:*
   1. Methodology Employed:

The given paper describes timetable generation to be an NP hard problem (a problem where there isn’t any specific approach or algorithm that can be used to solve it).The problem of timetable generation was defined as a problem of assigning the various resources to the meetings in a consistent manner. Timetable involves a set of incident E = {e1, e2,..…en} meeting a set of time to the said incident T = {t1, t2, …..ts}. The set of places where the incident occurred P = {p1, p2..,.pm} and a set of agents to conduct the incident A = {a1, a2, ….an} for example, lecturers. The paper describes the system design for an automatic timetable generation system using genetic algorithm. All the steps of genetic algorithm were clearly explained in the paper.

* 1. Result Achieved:

A study was conducted to adopt the electronic timetable generation system against the manual system that was used in most of the Nigeria Tertiary Institutions of learning. Around 215 questionnaires were distributed, and their response was recorded. Opinion of respondents towards their perception on the two systems based on time consumption, efficiency, convenience and overall performance were analysed on a Likert scale of three thereby resulting in the test of degree of relationship and formulation of hypothesis. Based on the empirical analysis of the study it was concluded that electronic system was more appropriate and would be the best method to adopt in tackling the lapses of its manual process.

* 1. Scope of future improvement:

The given system designed for approved by 93% of the people involved in the study. However, the system doesn’t provide the system specifications and modifications to handle soft constraints appropriately

* 1. Platform dependency and Experimental setup requirements:

Java was the programming language used for the development of the front end and Microsoft SQL Server was employed for designing the back end. Window 7 operating system is the minimum operating system required to run the application. Also, Adobe PDF reader or any other PDF reading software was used.

1. Branimir Sigl, Marin Golub and Vedran Mornar, *“Solving Timetable Scheduling Problem Using Genetic Algorithms”:*
   1. Methodology employed:

The papers approach to solving the Timetable problem is using a 3D structure.

The dimension of this structure is:

* + - 1. Days on the x axis
      2. Time Slots on the y axis
      3. Rooms on the z axis

The scheduling process is placing those cubes into a timetable such that conflicts are there. Conflicts in a timetable are a room being scheduled to two different classes at the same time, a lecturer teaching two or more classes at the same time or a class having two classes at one time.

* 1. Result Achieved:

The proposed system was tested on both large and small instances of problem. The large problem consisted of almost 770 classes with 41 rooms and 114 groups and the smaller problem was 70% less than this with 227 classes, 27 rooms and 55 groups. The smaller problem was solved with no conflicts. The larger problem with 95 conflicts which was later brought down to 20 conflicts using intelligent operators

* 1. Scope for future improvement:

The algorithm can be improved in certain ways. One way is when generating constraints, sign each one so that no constraint is checked twice. Individuals should be generated in such a way that classes which are difficult to schedule occupy the front genes and the easier ones behind.

* 1. Platform Dependency and Experimental Setup requirement:

The proposed system has no platform dependency and the setup requirement would be based on the technology used.

1. Nguyen Duc Thanh, *“Solving Timetabling Problem Using Genetic and Heuristic Algorithms”:*
   1. Methodology employed:

Like the *Samuel Lukas, Arnold Aribowo and Milyandreana Muchri, “Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable*”, this paper too makes use of Target Matrix with the same sets as described in the summary of the paper. Along with this a detailed set of rules for filling the target matrix is included. The Gene representation employed is using the column id which represents (c, o, i) where c is classes, o is courses and i is instructors. The flowchart of the algorithm is explained in detail. The initial population is generated randomly. Crossover is implemented using a two-point crossover and mutation is randomly changing the Gene.

* 1. Result Achieved:

In the experiment, the size of population, and the ratio of choosing parents and children for the next population are 100, 0.4 for parents and 0.6 for children. The probabilities of crossover, mutation and reversion are set to 0.8, 0.05, and 0.05 respectively. These values were choses experimentally through testing. In all cases an error free timetable was achieved.

* 1. Scope for future improvement:

However, in order to solve the general scheduling problem, further research needs to be done. For example, solving timetabling problems that allow one combination of classes, courses, and instructors takes more than one time per week.

* 1. Platform Dependency and Experimental Setup requirement:

The proposed system has no platform dependency and the setup requirement would be based on the technology used.

1. Theodore W. Manikas and James T. Cain, *“Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem”:*
   1. Methodology Employed:

The paper compares genetic algorithm and simulated annealing for its implementation in circuit design is placement where components are assigned to physical locations on a chip. A popular contemporary method for placement is the use of simulated annealing. While this approach has been shown to produce good placement solutions recent work in genetic algorithms had produced promising results. Genetic algorithm and simulated annealing algorithms were explained and implemented for the above problem and the results were compared and studied. A genetic algorithm is an iterative procedure that maintains a population of individuals these individuals are candidate solutions to the problem that had to be solved. Each iteration of the algorithm is called a generation. During each generation the individuals of the current population are rated for their effectiveness as solutions Based on these ratings a new population of candidate solutions is formed specific genetic operators. Simulated annealing is an iterative procedure that continuously updates one candidate solution until a termination condition is reached.

* 1. Result Achieved:

Both a genetic algorithm and simulated annealing approach were tested on a set of circuits Three circuits were selected for data sets. For the genetic algorithm the population size P and swing value W were varied during testing for simulated annealing the starting temperature T cooling factor number of move state M and stopping value ts were varied during testing Each set of parameter combinations forms a treatment. There were approximately 20 trials per treatment. Genetic algorithm produced a smaller average cut size than simulated annealing for two circuits and while no significant difference was found between the methods for the third circuit. Based on the results of the study the genetic algorithm was shown to produce solutions equal to or better than simulated annealing when applied to the circuit partitioning problem.

* 1. Scope of future improvement:

The study concluded its findings based on a very small dataset rather than considering a large data set to give an approximate percentage as on how much is genetic algorithm more efficient than simulated annealing. However, the paper explains both the algorithms in detail with the procedure and finding for all the test cases considered.

**2.2 Literature Survey Summarization:**

In two of the Surveyed papers, namely, Samuel Lukas, Arnold Aribowo and Milyandreana Muchri, “*Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable*” and Nguyen Duc Thanh, “Solving Timetabling Problem Using Genetic and Heuristic Algorithms” the system makes use of a target matrix on which Genetic Algorithm is applied to find the most suitable/optimised positions of slots in the matrix. Also, in both these papers Encoding of information into chromosome and Selection and Mutation Operators described are of the same type.

One gap in the Research is that none of the papers surveyed consider soft constraints. The papers only consider that classrooms, lectures, student groups and labs should not clash, i.e. the hard constraints.

The approach to solving the timetable problem using Genetic Algorithm is straight forward, but one aspect that is consistent in all the research papers is that each system implements certain optimization techniques to allow the solution to reach a global optimum. Those operations may include heuristic algorithms along with Genetic Algorithms or even modified Genetic Algorithm Stages.

**2.3 Problem Statement:**

The timetable generator aims to create the best optimised timetable for any given institution, consider various constraints while generating a timetable and apply the genetic algorithm approach to produce the best optimal timetable based on the organization's preferences and specified constraints.

**2.4 Objectives:**

* To generate an optimized timetable.
* To use genetic algorithm to generate the timetable
* To satisfy all the hard constraints.
* To satisfy soft constraints as much as possible.
* Final timetable is the timetable with the best fitness.

**2.5 Scope of the Project:**

Project Goals:

* Make a comprehensive “Automatic Timetable Generation” Software that Schedules lectures, lab, tutorials, workshop, etc. efficiently.
* Make use of Genetic Algorithm for its implementation.
* Software handles all the relevant constraints.

Project Deliverables:

* Timetable generated automatically using Genetic Algorithm
* Exporting Timetable for each lecture, lab, tutorials, workshop into suitable formats.