ISSN No. 0976-5697

Volume 6, No. 2, March-April 2015

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Review of Generation of Time Table Using Genetic Algorithm Implemented in Java

Khan Arman [*1,] Khan Sabir Ali [2,] Choudhary Suhel [3]
Department of Computer Engineering
AIKTC, Panvel, Navi Mumbai,
Mumbai university,
Maharashtra, India

Prof. Kalpana R. Bodke
Assistant Professor
Department of Computer Engineering
AIKTC, Panvel, Navi Mumbai, Mumbai university,
Maharashtra, India

Abstract— An educational timetabling is a multi-dimensional and highly constrained problem. Making educational timetables manually often involves number of rounds of changes before they can be satisfied. In Fact such a process can be time consumings, and often the quality of the timetables is compromised due to pressure to release the TT on time. Automatic generation of timetables then seems to be an attractive to manual approach. But this approach is with problems. In reality, most timetabling problems are NP-complete and most professors and researchers are interested in developing efficient algorithms for solving the time table problem.

In this, a college timetable problem formulation is introduced lowed by recent approaches for solving the problem. After that, a genetic algorithm (GA) is presented to efficiently and effectively solve the problem. The proposed GA has a malleable (flexible) representation that handles all the college timetables at once. It includes repair strategies to always guarantee the creation of a feasible timetable which satisfies constraints that must not be destroyed. The algorithm is implemented and applied to create timetables for the Arab Academy for Science and Technology (AAST) in Egypt and it shows promising results.

Keywords— Genetic Algorithm (GA), Time Table Problem (TTP), Non Polynomial Problem (NP-Problem), Chromosome, Fitness, Mutation, Crossover.

I. INTRODUCTION

An educational timetabling problem can be defined to be the problem of assigning a number of events into a limited number of time slot or time periods.[7] "To allocate the objects in space time of given resources of subject to constraint, to satisfy as nearly as possible a set of desirable objectives"[2]. Holland's original schema was a method of classifying objects, then selectively Merging" those objects with each other to produce new objects to be classified. Created for the purpose of modeling Darwinian theory of natural selection, the programs followed a simple pattern of the birth, mating and death forms[3].

A. Problem Statement:

Scheduling is the arrangement of entities (people, tasks, vehicles, lectures, exams, meetings, etc.) into a pattern in space-time in such a way that constraints are satisfied and certain goals are achieved. Constructing a schedule is the problem in which time, space and other (often limited) resources have to be considered in the arrangement.

Designing a timetable manually, consumes a lot of time. It is tedious work as we have to consider many constraints like the teaching hours, subject scheduling, conducting theory and practical hours and the overall weekly lectures. Genetic algorithm takes care of this problem. It takes the entire iThe collection of soft constraints that are considered in the algorithm is: -

a. Most lecturers and some students do not wish to have empty periods in their timetable.
b. More than two time period per day are not allowed for a class of students.
c. Teaching load for every staff should be considered. Let's say it is maximum 15 hours per week. Input, meets the constraints and generates an optimal solution.

The solution to this problem can be found by using genetic algorithm. It would take the academic course details as input. This will include the subjects, number of days and lectures per day, details of teachers and the load capacity.

In its processing, the algorithm will check all the constraints that need to be followed and will generate such a timetable which is without any conflicts[1].

B. Objective and Scope of Project:

The main goal of this project is the analysing into multidimensional specifics of timetabling and scheduling and finding the most appropriate and efficient algorithm for a software solution.

The timetable scheduling problem is common to all educational institutions. Main algorithm goal is to minimize the number of conflicts in the timetable.

The aim is to fill up all the slots in the timetable.

It also ensures proper distribution of lectures of a particular subject.

Another aim is to ensure that the number of teaching hours does not exceed for any teacher.

C. Scope:

Single bottleneck resource of the educational capacity that is the teaching staff.

It helps to overcome the problems faced during manual execution.

The contribution is two-fold: -

a. The educational capacity analysis is to provide the overall methodology
b. To create the adequate database implementations.

II. LITERATURE REVIEW

A. Genetic Algorithm:

The family of artificial intelligence of computer science has Genetic algorithm which is based on evolution or random generation of population, selection, crossover and mutation. Specific domain problems are solved by converting them into models by using a chromosome-like teacher ID and evolve the chromosomes using random generated population, selection, Crossover and mutation.

The genetic algorithm usually starts with a randomly selected sets of chromosomes. The chromosomes that are selected specify the issue that is to be handled. Depending on the properties and problem's characteristic, different positions of each chromosome are encoded as bits, characters or numbers[4]. The generated chromosomes or bits are randomly changed for better fitness. The collection of soft constraints that are considered in the algorithm is shown in figure 1.

a. Most lecturers and some students do not wish to have empty periods in their timetable.
b. More than two time period per day are not allowed for a class.
c. Teaching load for every staff should be considered.

Let's say it is maximum 15 hours per week. The set of chromosomes during a stage of evolution are called as population .For better Fitness an evaluation function is used for calculation. During evaluation calculation, two basic operators,

Crossover and mutation, are used to simulate the re-population or regeneration and mutation of species. The selection of chromosomes for generation and combination is biased towards the fittest chromosomes.

Figure below shows the structure of a simple genetic algorithm. It starts with a randomly generated population, emerge through selection, recombination (crossover), and mutation. Finally, the best individual (chromosome) is picked out as the final result once the optimizations criteria is met.[5]

B. Chromosome Encoding, Fitness, Crossover and Mutation:

A number of properties of binary encoding work to provide simple, effective and elegant GAs. There are, many other ways to represent a creature's genes, which can have their own implicit advantages. In order to get a problem into gene form, the substance of its solution must be represented as a collection of units of information. This is true for many problems.[2]

This can be thought of as not just a list of values but an array of genes[3]. The value in the first row might represent the day and the second row represents the year and so on. Each of these values might be converted from base 10 to base 2 to create a fixed width binary number. Hence the problem of minimizing problem while maintaining your survival is translated into a genetic representation. [3]

A collection of possible fitness could be thus encoded, producing a population of chromosome or creature. Random populations are almost, always extremely unfit. In order to determine which are better than others, each creature must be re-assessed. In order to examine a child, there must be some relevant information about the environment.[2]

Depending on the way, we structure the method of evaluating a chromosome, we can either aim to generate the least costly population or the fitness; it is a question of minimizing cost or maximizing fitness. When discussing optimization techniques, the range of possible solutions is often referred to as the solution space and the cost/fitness of each point in the solution space is referred to as the altitude in the landscape of the problem.[2]

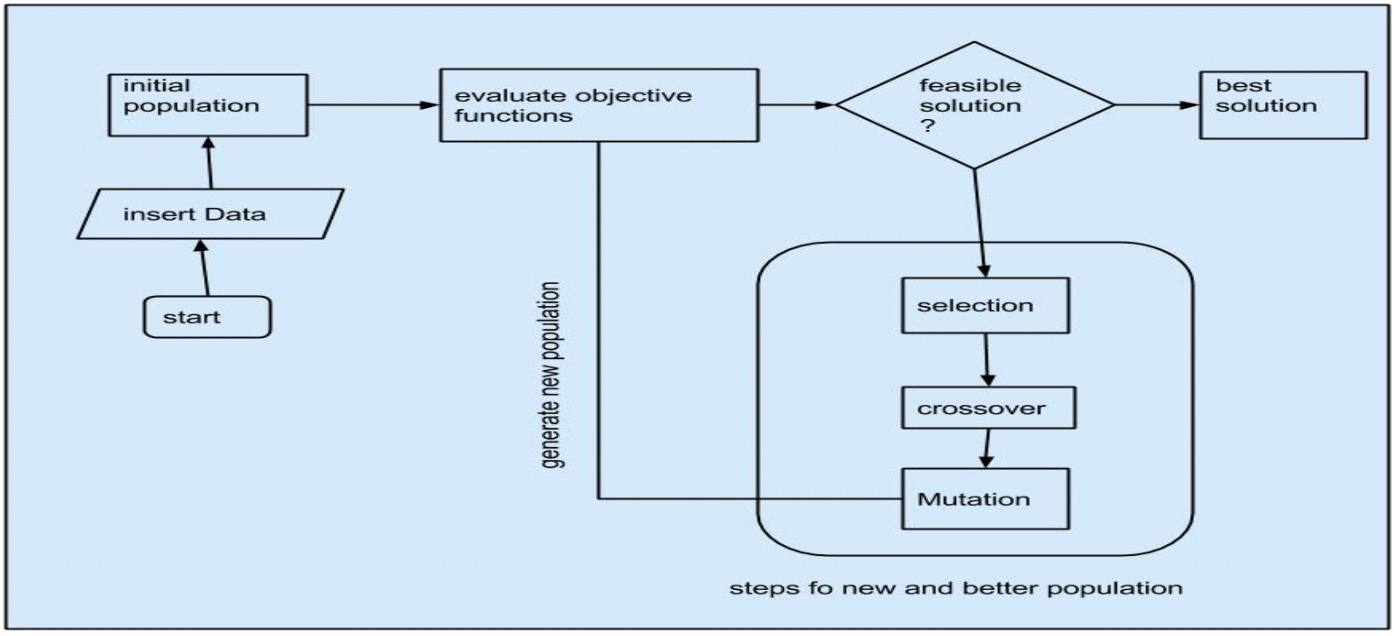

Figure1. Structure of a simple genetic algorithm

Looking for the global minimum of the cost is also to look for the lowest point in the lowest valley of the cost landscape. Similarly, to find the global maximum fitness is to look for the highest point of the highest mountain in the fitness portrait. In many GA's, the process of choosing child for breeding is handled in multi-ways. Holland's developed model uses a methodology where the chromosomes must be fittest and healthiest. Other methods select any two creatures at random for breeding. Selective breeding can be used in conjunction with or in the absence of an Elitist Natural Selection Operator- in either case the GA can perform evolution. Once parents have been considered, Reproduction can then be carried out. A new child is produced by considering each gene in the chromosome, which holds a similar characteristic either from the father or mother. [1]

The process of combining the genes can be performed in a number of ways. One of the easiest process of combining chromosomes is termed as one point crossover. This can be best demonstrated using genes encoded in binary, though the process is translatable to almost any gene representation. A child chromosome can be produced using one point crossover, as shown in figure 2. A crossover point is randomly chosen to occur somewhere in the set of genes.One Crossover point is selected ,genetic characteristic before the selected crossover point is taken from one parent and all genetic characteristic after the selected crossover point is taken from another parent.

After crossover is performed and before the child is released into the world, there is a chance that it will undergo mutation. The chance of this happening is referred to as the rate of mutation. This is usually kept quite small. The purpose of mutation is to put some distortion, and, in general, newly specified chromosome, into the population. This is useful in escaping local minima as it helps explore new regions of the multi-dimensional solution space. Once a gene has been choose for mutation, the mutation itself can work in many different ways. This again, depends on the implementation of the GA. In the case of a binary string representation, simple mutation of a gene causes that genes value to be complemented- a 1 becomes a 0 and vice versa.[1]

a) In Holland's founding work on GA's he mentioned another operator, besides selection, breeding, crossover and mutation which takes place in biological reproduction. This is defined as the inversion operator. An inversion is where a portion of chromosomes detaches from the rest of the chromosome, then changes its path and recombines with the chromosome.[6]

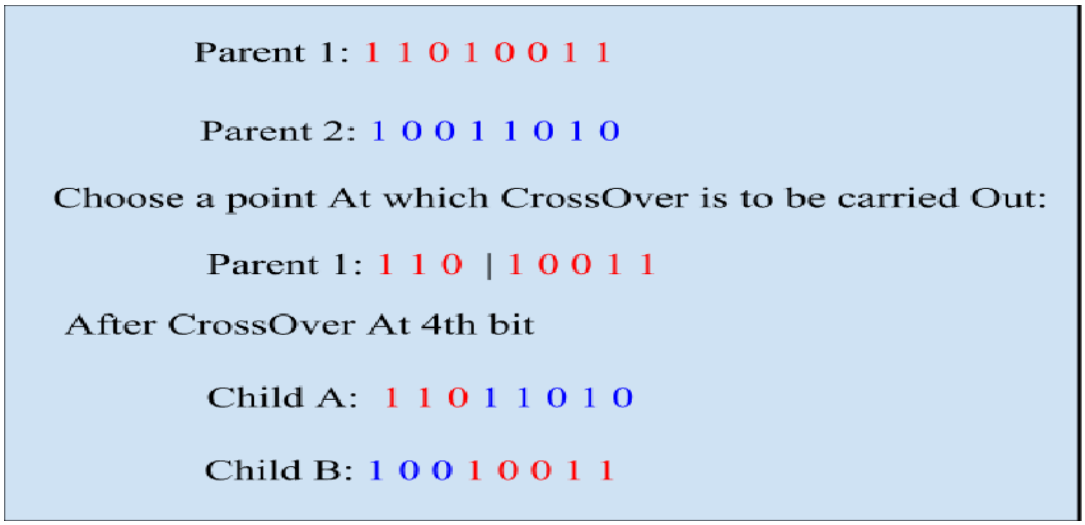

Figure2. An Example of Crossover with Fully Encoded Genes

a. Hard Constraints:

The collection of hard constraints that need to be satisfied in designing the proposed Genetic algorithm is: -

(a). A course must not have two bookings simultaneously.
(b). Classrooms must not be double booked.
(c). Some courses require particular rooms. For example, experiments might be held in particular laboratories.
(d). A teacher must not be assigned two lectures simultaneously.

b. Soft constraints:

The collection of soft constraints that are considered in the algorithm is: -

(a). Most lecturers and some students do not wish to have empty periods in their timetable.
(b). More than two time period per day are not allowed for a class of students.
(c). Teaching load for every staff should be considered. Let's say it is maximum 15 hours per week.

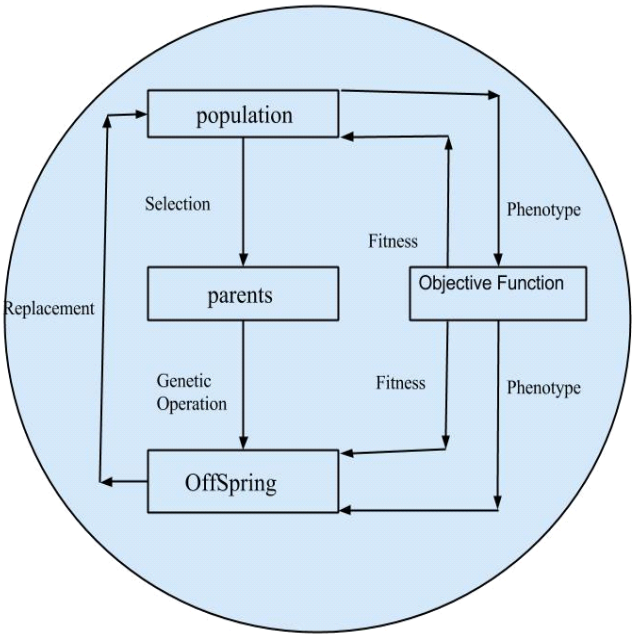

Figure3. Life-cycle of a particular genetic algorithm.

III. PSEUDO CODE

Explanation of algorithm.

Step1 initialize the population

Step2 checks for generated population if generated population i.e g is less then new generation i.e n-Generation go inside the while loop..

Step3 In this randomly generated chromosome are carried out.

Step4 we select any two parents(generated Chromosome).

Step5 in this we create offspring or child as new generation and defiend the crossover point.and performs the crossover on selected parents.

Step6 in this we check for mutation can be done or not.If this is true then go to Step7 perform mutation.

Step7 after this search for exists offspring.

Step8 evaluate offspring.expands the offspring.and passes to next step

Step9 evaluated offspring then added in P variable which holds the population

Step10 now increase the i by 1 and repeat from step3 otherwise go to next

Step11 now select chromosome and update new generation repeat the steps from step2.

End the while loop

```
Step1:- Initialize population p
Step2:-While(g<n-Generation)
Step3:-While(i<n-Individuals)
Step4:-Parents ->select any 2 individuals from P
Step5:-Offspring -> crossover(parents)
Step6:-If(Mutation Acceptation Criteria is valid) //no
two lectures of same   subject in one day
Step7:-Mutation(Offsprings)
Step8:-Offsprings -> Local-search(offspring)
Step9:-Evaluate(offspring)
Step10:-Add offspring in P
Step11:- i-> i+1
Step12:- End while
Step13:- P -> select (P)
Step14:- g -> g+1
Step15:- End while
Step16:- End
```

IV. EXPECTED OUTCOME

AIKTC

Regenerate Timetable

COMPUTER BRANCH

DAY	CLASS	9:30-10:30	10:30-11:30	11:30-12:30	1:00-2:00	2:00-3:00	3:00-4:00	4:00-5:00
MON	S.E.	EDLC {1,1,1,4} Prof. Aamir	DSGT TUTORIAL	DSGT TUTORIAL	Maths 3 {1,1,4,1} Prof. Tabrez	DSF {1,1,5,3} Prof. Shalini	PCT SEMINAR	PCT SEMINAR
	T.E.	ADBMS {1,2,1,9} Prof. Prashant	WE {1,2,2,1} Prof. Tabrez	CN {1,2,3,8} Prof. Farhaan	WE-P /MP-P /ADBMS...	WE-P /MP-P /ADBMS...	EVS {1,2,6,3} Prof. Shalini	MP {1,2,7,7} Prof. Kalpana
	B.E.	DSIP {1,3,1,5} Prof. Salaam	DSIP-P /Elec-P /SS-P	DSIP-P /Elec-P /SS-P	Project 1 {1,3,4,9} Prof. Prashant	RAI {1,3,5,7} Prof. Kalpana	SS {1,3,6,4} Prof. Aamir	Elec {1,3,7,8} Prof. Farhaan
TUE	S.E.	DSF {2,1,1,3} Prof. Shalini	COA {2,1,2,5} Prof. Salaam	DLDA {2,1,3,6} Prof. Sameer	Maths 3 {2,1,4,1} Prof. Tabrez	DSGT {2,1,5,2} Prof. Ashraf	DLDA-P /COA-P /DS...	DLDA-P /COA-P /DS.
	T.E.	WE {2,2,1,1} Prof. Tabrez	MP {2,2,2,7} Prof. Kalpana	ADBMS {2,2,3,9} Prof. Prashant	MP-P /ADBMS-P /CN...	MP-P /ADBMS-P /CN...	TCS {2,2,6,2} Prof. Ashraf	CN {2,2,7,8} Prof. Farhaan
	B.E.	Project 1 {2,3,1,9} Prof. Prashant	MC-P /RAI-P /DSIP-P	MC-P /RAI-P /DSIP-P	RAI {2,3,4,7} Prof. Kalpana	SS {2,3,5,4} Prof. Aamir	DSIP {2,3,6,5} Prof. Salaam	MC {2,3,7,6} Prof. Sameer
WED	S.E.	DSF {3,1,1,3} Prof. Shalini	Maths 3 {3,1,2,1} Prof. Tabrez	DSGT {3,1,3,2} Prof. Ashraf	DLDA {3,1,4,6} Prof. Sameer	EDLC {3,1,5,4} Prof. Aamir	DSF-P /EDLC-P /COA...	DSF-P /EDLC-P /COA
	T.E.	MP {3,2,1,7} Prof. Kalpana	TCS TUTORIAL	TCS TUTORIAL	ADBMS-P /CN-P /TC...	ADBMS-P /CN-P /TC...	CN {3,2,6,8} Prof. Farhaan	EVS {3,2,7,3} Prof. Shalini
	B.E.	Elec {3,3,1,8} Prof. Farhaan	Elec-P /SS-P /MC-P	Elec-P /SS-P /MC-P	DSIP {3,3,4,5} Prof. Salaam	Project 1 {3,3,5,9} Prof. Prashant	MC {3,3,6,6} Prof. Sameer	RAI {3,3,7,7} Prof. Kalpana
THU	S.E.	EDLC {4,1,1,4} Prof. Aamir	DSGT {4,1,2,2} Prof. Ashraf	DLDA {4,1,3,6} Prof. Sameer	Maths 3 {4,1,4,1} Prof. Tabrez	COA {4,1,5,5} Prof. Salaam	COA-P /DLDA-P /ED...	COA-P /DLDA-P /ED.
	T.E.	EVS {4,2,1,3} Prof. Shalini	ADBMS {4,2,2,9} Prof. Prashant	WE {4,2,3,1} Prof. Tabrez	TCS-P /WE-P /MP-P	TCS-P /WE-P /MP-P	CN {4,2,6,8} Prof. Farhaan	TCS {4,2,7,2} Prof. Ashraf
	B.E.	MC {4,3,1,6} Prof. Sameer	SS-P /MC-P /RAI-P	SS-P /MC-P /RAI-P	RAI {4,3,4,7} Prof. Kalpana	CN {4,2,6,8} Prof. Farhaan	Project 1 {4,3,6,9} Prof. Prashant	SS {4,3,7,4} Prof. Aamir
FRI	S.E.	SS-P /MC-P /RAI-P	Maths 3 {5,1,2,1} Prof. Tabrez	EDLC {5,1,3,4} Prof. Aamir	DSF {5,1,4,3} Prof. Shalini	DSGT {5,1,5,2} Prof. Ashraf	EDLC-P /DSF-P /COA...	EDLC-P /DSF-P /COA
	T.E.	TCS {5,2,1,2} Prof. Ashraf	ADBMS {5,2,2,9} Prof. Prashant	EVS {5,2,3,3} Prof. Shalini	CN-P /TCS-P /WE-P	CN-P /TCS-P /WE-P	MP {5,2,6,7} Prof. Kalpana	WE {5,2,7,1} Prof. Tabrez
	B.E.	Project 1 {5,3,1,9} Prof. Prashant	RAI-P /DSIP-P /Elec-P	RAI-P /DSIP-P /Elec-P	DSIP {5,3,4,5} Prof. Salaam	SS {5,3,5,4} Prof. Aamir	MC {5,3,6,6} Prof. Sameer	Elec {5,3,7,8} Prof. Farhaan

Figure4. Life-cycle of a particular genetic algorithm

V. CONCLUSION

The problem of finding the global optimum in a space with many local optima is a classical problem for all systems that can be perceived and learned. GA gives a descriptive search functionality for feasibility. GA can be applied to both continuous and distinguishable feasible problems .Globally in optimization cases, GA's generally display their power: efficiency, parallelizable searching technique; the ability to generate solutions with multiple objective criteria; and a controllable process of innovation.

a. Genetic algorithms are vitally parallel.
b. Genetic algorithm does not leave any empty slots.
c. Genetic algorithm produces multiple equally good solutions.
d. Genetic algorithm drags the population away from the local optimum and ensures global optimum is reached.
e. Genetic algorithm is independent of domain-specific information.

VI. REFERENCES

[1]. Sujit Kumar Jha "Exam Time Tabling Using Genetic Algorithm",IJRET: International Journal of Research in Engineering and Technology, June 2014

[2]. Chetan Kale, Sarfaraj Hodekar, Avinash Pawar, Prof.V.S More,"Time Table Scheduling Using Genetic Algorithm",International Journal of Research in Advent Technology (IJRAT) Vol. 1, No. 3, October 2013.

[3]. Leon Bambrick,"Lecture TimeTabling Using Genetic Algorithm",Thesis Bachelor of Computer System Engg. www.researchgate.net/publication/239927876_Lecture_Timetabling_Using_Genetic_Algorithms.

[4]. Ehab Talal Abdel-Ra'of Bader, Hebah H.O. Nasereddin," Using Genetic Algorithm in Network Security",IJRRAS 5 (2) November 2010.

[5]. Bahaa Mohsen Zbeel,"Using Genetic Algorithm for Network Intrusion Detection ",www.iasj.net/iasj?func=fulltex&ald=74365,Nov 2013.

[6]. Sandeep Singh Rawat, Lakshmi Rajamani, "A time TAble Prediction for Technical Educational Systems Usnig Genetic Algorithm",Journal of Theoretical and Applied Information Technology, 2010.

[7]. Edmund Kieran Burke and Sanja Petrovic, "Recent Research Directions in Automated Timetabling", European Journal of Operational Research – EJOR, 2002.