A Genetic Algorithmic Approach To University Time Tabling Problem

Rohini.V, Dr. A.M.Natarajan

Assistant Professor, Christ University, Bangalore. *rohini.v@christuniversity.in*,
Mob: 9449111413
Chief Executive Officer, Bannari Amman Institute of Technology, Sathyamangalam.

***Abstract*--Scheduling & planning are considered as the most vital events in our life. Course scheduling is a well known constraint satisfaction problem .Course scheduling is a difficult and time consuming job. Here the idea is to use a meta heuristic (genetic algorithm), which is a good solution for the NP hard problems like scheduling. The main objective is to create a novel course scheduling system by using the meta heuristic algorithm. It is cumbersome to solve by the classical algorithms. Optimization techniques are used to solve them and produce optimal or near optimal feasible solutions instead of exact solutions. Genetic algorithms are considered as an efficient approach for solving the type of problem. The paper deals with timetabling of various courses particularly course scheduling and examine the various ways in which meta heuristic techniques might be applied to these sorts of problem. The research starts by defining the problem of scheduling the courses in Christ University, Department of Computer Science and defining the various complicated constraints available. Then the problem model is explained with the set of resources like professors, rooms, labs, student etc. The solution is generated by using a metaheuristic algorithm.**

Keywords: Constraints, Course scheduling, Genetic algorithm, NP hard problem, Phased, Hard constraints Soft constraints

I. LITERATURE REVIEW

In 1975, Holland developed the idea of genetic algorithm idea in his book "Adaptation in natural and artificial systems"[2]. The book describes how to apply the principles of natural evolution to optimization problems and built the first Genetic Algorithms. Holland's theory has been further developed and now Genetic Algorithms (GAs) stand up as a powerful tool for solving search and optimization problems. A genetic algorithm is a problem solving method that uses genetics as its model of problem solving. It's a search technique to find approximate solutions to optimization and search problems [4] .Basically, an optimization problem looks really simple. One knows the form of all possible solutions corresponding to a specific question. The problem consists in finding out the solution that fits the best, i.e. the one with the most payoffs, from all the possible solutions. If it's possible to quickly enumerate all the solutions, the problem does not raise much difficulty.

In 2012, Kumar et. al.[7] explains the use of genetic algorithms in automating the timetabling problem of a university. They have included a fitness function, which is a basic building block for deciding the rank of the chromosome against all other chromosome. Optimal chromosome are allowed to breed and mix their datasets by some specific techniques out of many alternative ways, producing a new generation that will be better one. Karami [8] states that the university course timetabling problem (UCTP) is a multidimensional problem in which teachers, students, and courses are assigned to timeslots and located in suitable rooms. He has taken the Socha's dataset for his experimentation and developed a HGA(Hybrid Genetic Algorithm) for the scheduling problem.

Muller (2012 PATAT) relies on UniTime, which is a university based time tabling system, containing an enrolment based course timetabling solver , which already deals with course sections and configuration of courses.[9] They proposed a curriculum model that is able to tackle an advanced set of real life characteristics of time tabling problem. Vinayak,[10] in his paper exploits the rank based selection scheme to ensure that the time table schedule generated is the feasible global

optima as opposed to the stagnant solution setup associated with roulette selection scheme. An application specific encoding structure, rank based selection to time table schedules and single point crossover to explore new and fitter solutions is used.

Dhande et. al. (2012) illustrates the diversity of approaches that Genetic algorithm provide fairly comprehensible way to address a wide range of difficult engineering and optimization problem producing good if not optimal results, it seems that the technology is finding its way into real world use much more easily than the expert systems did.[11]

In the paper [12], Sanjay presents a novel approach of solving University timetabling which is a NP hard problem using Genetic algorithm. The paper has well defined constraints (hard) towards scheduling and it is incorporated.

Chaudhuri (2010) proposes a Fuzzy Genetic Heuristic algorithm which incorporates Genetic algorithm using indirect representations based on event priorities, Micro Genetic Algorithm and heuristic local search operators to tackle real world time table problem. Fuzzy set model measure of violation of soft constraint in fitness function to take care of inherent uncertainty and vagueness involved in real life data. [13]

Methodology

A Simple Genetic Algorithm

Genetic Algorithm starts by generating an initial population of chromosomes. This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be engendered. Generally, the initial population is generated randomly. In course scheduling the time table is generated randomly. Then, the genetic algorithm loops over an iteration process to make the population evolve.
The time table problem goes thru an cycle to satisfy the constraints of the department.
Each iteration consists of the following steps:

SELECTION:

The first step consists in selecting individuals for reproduction.
This selection is done randomly with a probability depending on the relative fitness of the
Individuals so that best ones are often chosen for reproduction than poor ones. The constraints are considered for selection of the chromosome(subject).

REPRODUCTION:

In the second step, offspring are bred by the selected individuals.
For generating new chromosomes, the algorithm can use both recombination and mutation. For allotment of the time slots, the subjects are checked.

EVALUATION:

Then the fitness of the new chromosomes is evaluated. The allotment of the staff, hour and resource is done.

REPLACEMENT:

During the last step, individuals from the old population are killed and replaced by the new ones. In timetabling, the constraints, like different subjects should be handled in the first time slot(9-10), can be considered and the allotment rescheduled.
The algorithm is stopped when the population converges toward the optimal solution. The time table will be fixed when the hard constraints are satisfied and some of the soft constraints are experimented.

The basic genetic algorithm is as follows:

[Start] Genetic random population of n chromosomes (suitable solutions for the problem) *
[Fitness] Evaluate the fitness f(x) of each chromosome x in the population
[New population] Create a new population by repeating following steps until the New population is complete
[Selection] select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
[Crossover] With a crossover probability, cross over the parents to form new offspring (children).

If no crossover was performed, offspring is the exact copy of parents.

[Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome)

[Accepting] Place new offspring in the new population.

[Replace] Use new generated population for a further sum of the algorithm.

[Test] If the end condition is satisfied, stop, and return the best solution in current population.

[Loop] Go to step2 for fitness evaluation.

In Genetic algorithm cycle, Reproduction is the process by which the genetic material in two or more parents is combined to obtain one or more offspring. In timetabling system, , based on the slot and subjects, the allotment falls. In fitness evaluation step, the individual's quality is assessed. The constraints are evaluated and allotment is done accordingly. Mutation is performed on one individual to produce a new version of it where some of the original genetic material has been randomly changed. Some of the constraints are evaluated and satisfied. Selection process helps to decide which individuals are to be used for reproduction and mutation in order to produce new search points.

Before implementing GA, it is important to understand few guidelines for designing a general search algorithm i.e. a global optimization algorithm based on the properties of the fitness landscape and the most common optimization method types:

1. Determinism A purely deterministic search may have an extremely high variance in solution quality because it may soon get stuck in worst case situations from which it is incapable to escape because of its determinism. This can be avoided, but it is a well-known fact that the observation of the worst-case situation is not guaranteed to be possible in general.
2. Non determinism: A stochastic search method usually does not suffer from the above potential worst case "wolf trap" phenomenon. It is therefore likely that a search method should be stochastic, but it may well contain a substantial portion of determinism, however. In principle, for a course scheduling system, it is enough to have as much non determinism as to be able to avoid the worst-case wolf traps.

II. EMPIRICAL STUDY

In this section, a real world university course scheduling system is represented and is solved by the genetic algorithm described above. The Department of Computer Science, Christ University, Bangalore is a large department, consisting of various courses in Computer Science like BCA, B.Sc., MCA, M.Sc. and consists of thirty professors. Scheduling the subjects to the various courses in the allotted time slot and resources is a complex hard problem. The job involves scheduling the various subjects to the course, various staff and resources adhering to the constraints stated.

The problem consists of the following entities:

Days, Timeslots, and Periods. :

We are given a number of teaching days in the week (typically 6). Each day is split in a fixed number of timeslots of 6 (9 am to 4 pm, except 1 – 2 pm for lunch) and Saturdays only 4 hours (9 am – 1 pm). A period is a pair composed by a day and a timeslot. The total number of scheduling periods is the product of the days times the day timeslots.

Courses and professors:

Each course consists of a fixed number of lectures to be scheduled in distinct periods; it is attended by given number of students, and is taught by a teacher. For each course there is a minimum number of days that the lectures of the course should be spread in, moreover there are some periods in which the course cannot be scheduled.

Rooms.:

Each room has a capacity, expressed in terms of number of available seats. All rooms are equally suitable for all courses (if large enough). Rooms will have the resources like board (white / black), Projector etc.

The timetables are feasible if and only if the following hard constraints are satisfied:

- No instructor teaches more than one course at a given time.

- Number of courses taught during any time slot should not exceed the maximum number of classrooms and labs available at that specified time.
- Restrictions on the starting time, ending time, lunch time.
- Lectures and labs of each group of students should not overlap.
- Practical courses should run on two continuous time slots.
- Class timings is 9 -4 with a break of lunch from 1-2 pm.

The following soft constraints should be satisfied (if possible):

- There should be a gap of at least one hour in the schedule between courses taught by the same instructor.
- Course should be spread over evenly throughout the week (Monday to Saturday).
- All lectures of a particular course should be assigned to the same room.
- All first hour (9-10 am) should be allocated to different subject/faculty.
- Repetition of the same subject more than once per day should be avoided.

The problem is solved in a phased approach as follows :[5]

Phase 1: Allotment of subjects to the professors.

Phase II : Allot the labs to the courses.

Phase III: Assigning lectures to time slots.

Phase IV: Assigns labs and tutorials to days and available time slots in the days.

Subject allocation of the various courses to the professors are done by the HOD and Coordinators and given to the committee. The scheduling committee is responsible for allotting the time slots, resources to the various courses satisfying the above mentioned constraints. According to the meta heuristic algorithm described, the relationship of those events and resources, a course scheduling system is going to be developed. Finally, a solution which has the best fitness is obtained. Through analysis of the time table prepared, the constraints are satisfied and some soft constraints are not able to satisfy. It will be proved that the meta heuristic algorithm for course scheduling will be an effective one adhering to all the constraints given by the department.

III. CONCLUSION

The system developed a time table for a particular department of a university. It follows the constraints(hard) and some of the soft constraints. It assumes the rooms are free and they satisfy the needs of the course scheduled. The room allotment can be included in the future.

REFERENCES

[1] Bratković, Z., Herman, T., Omrčen, V., Čupić, M., Jakobović, D.: University course timetabling with genetic algorithm: A laboratory exercises case study. In: Cotta, C., Cowling, P. (eds.) EvoCOP 2009. LNCS, vol. 5482, pp. 240–251. Springer, Heidelberg (2009)

[2] J.H. Holland , Adaptation in Natural and Artificial systems, Cambridge, The MIT press, 1992

[3] D.Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning", Addison Wesley, 1989

[4] S.N.Sivanandam and S.N.Deepa , Introduction to Genetic Algorithms, Springer, 2007

[5] Minhaz Fahim Zibran, A Multi-phase Approach to University Course Timetabling, University of Lethbridge (Canada), 2007

[6] S Yang, S.N. Jat, "Genetic algorithms with guided and local search strategies for university course timetabling. " IEEE transaction on system , man and cybernetics – Part C : Applications and reviews, vol 41, no.1,pp.93-106,January 2011.

[7] Kumar Kuldeep, Srikander et al. , Genetic Algorithm Approach to Automate University Timetable, International Journal of Technical Research, vol 1, Issue 1, pg 33- 37, Mar- Apr 2012.

[8] Amir Hossein Karami, University Course Timetabling Using a new Hybrid Genetic algorithm, Intl. conf. on Computer and Knowledge engineering, October 2012.

[9] Muller Thomas, Rodova Hana, Real life curriculam based timetabling, PATAT,2012.

[10] Vinayak Sapru, Kaushik Reddy,B Sivaselvan, Time table scheduling using Genetic algorithms employing guided mutation ,IEEE 2010.

[11] Dhande A S, Dr.Ladhake, Dhande S S, Genetic – Algorithm – An effective approach to solve real application problems , International Journal of Advanced Research in Computer Science and Software Engineering , Vol 2, Issue 2, Feb 2012.

[12] Sanjay R Sutar, Rajan S Bichkar, University timetabling based on Hard constraints using Genetic algorithm, Internation journal Computer Applications, Vol 42,no.15, March 2012.

[13] Arindham Chaudhuri, Kajal De, Fuzzy Genetic heuristic for University Course Timetable problem,Intl. Journal of Soft Computing Applications, Vol.2,No.1, March 2010.

[14] E K Bruke , J P Newall, Solving Examination timetabling problems through adaptation of Heuristic orderings, Annals of Operations Research, Vol. 129,(2004), PP.107-192.

[15] N D Thanh, Solving Timetable Problem using Genetic and Heuristic Algorithms, Proceedings of 8 th ACIS International Conference ,2007.