Timetable Creation using Genetic Algorithm in Grid Computing

VIJAYASRI A[1]

Asst.Prof. Mrs. J. KALAIVANI

Asst.prof Mr.K. Nirmal, MCA., M.Phil

Krishnasamy College of Engineering and Technology,

Cuddalore.

Abstract: This paper presents Genetic Algorithms (GAs) based schedulers for efficiently allocating jobs to resources in a Grid system. Scheduling is a key problem in emergent computational systems, such as Grid and P2P, in order to benefit from the large computing capacity of such systems. It present an extensive study on the usefulness of GAs for designing efficient Grid schedulers when make span and flow time are minimized The extensive experimental study showed that our GA-based schedulers outperform existing GA implementations in the literature for the problem and also revealed their efficiency when make span and flow time are minimized either in a hierarchical or a simultaneous optimization mode; previous approaches considered only the minimization of the make span. Moreover, It able to identify which GAs versions work best under certain Grid characteristics, which is very useful for real Grids. Our GA-based schedulers are very fast and hence they can be used to dynamically schedule jobs arrived in the Grid system by running in batch mode for a short time.

1. Introduction: A computational grid is a large scale, heterogeneous collection of autonomoussystems, geographically distributed and interconnected by heterogeneous networks.Job sharing (computational burden) is one of the major difficult tasks in a computationalgrid environment. Grid resource manager provides the functionality for discovery and publishing of resources as well as scheduling, submission and monitoring of jobs. The use of Grid infrastructures in solving complex problems from many fields of interest such as optimization, scientific simulation, drug discovery, bio-informatics etc. Therefore scheduling problems in conventional distributed systems. The problem is multi-objective in its general formulation, the two most important objectives being the minimization of makespan and flowtime of the system.

2.Literature survey: Genetic algorithms are general search and optimization algorithms inspired by processes and normally associated with natural world. Genetic algorithm mimics the process of natural selection and can be used as a technique for solving complex optimization problems which have large spaces . They can be used as techniques for solving complex problems and for searching of large problem spaces. Unlike many heuristic schemes, which have only one optimal solution at any time, Genetic algorithms maintain many individual solutions in the form of population.A typical algorithmthen uses three operators, selection, crossover andmutation, to direct the population toward convergence atglobal optimum. It requires a processof initializing, breeding, mutating, choosing andkilling. It can be said that most methods called GAs have at least the following elements in common: Population of chromosomes, Selection according to fitness, Crossover to produce new offspring, and random mutation of new offspring.

3. Proposed approach: In order to deal with timetabling issues we are proposing a system which would mechanically generate timetable for the institute. Course and lectures will be scheduled in accordance with all possible constraints and given inputs and thus a timetable will be generated.

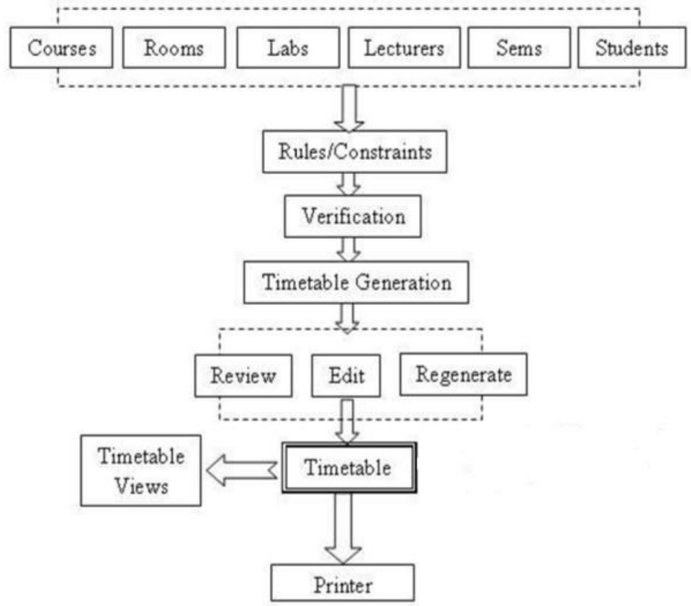

Fig 1: General view of Timetable Generator.

Structure of time table generator consists of input data, relation between the input data, system constraints and application of genetics algorithm.

4. GA Operators:

1) *Chromosome representation:*Chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve. The chromosome is often represented as a simple string. The fitness of a chromosome depends upon how well that chromosome solves the problem at hand.

2) *Initial population:*The first step in the functioningof a GA is the generation of an initial population. Eachmember of this population encodes a possible solution to a problem. After creating the initial population, eachindividual is evaluated and assigned a fitness valueaccording to the fitness function. It has been recognizedthat if the initial population to the GA is good, then thealgorithm has a better possibility of finding a goodsolution and that, if the initial supply of building blocks isnot large enough or good enough, then it would be difficultfor the algorithm to find a good solution.

3) ***Selection:***This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected toreproduce .

4) ***Crossover:***In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Crossover is a process of taking more than one parent solutions and producing a child solution from them. There are methods for selection of the chromosomes. This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two Single-chromosome organisms.

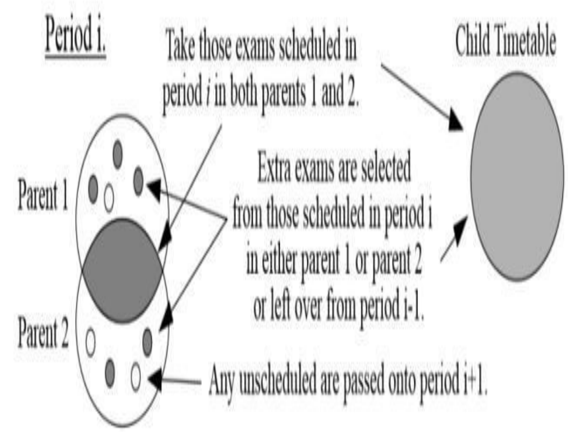

Fig 2: Crossover Operator.

5) ***Mutation:***Mutation is agenetic operator used tomaintain genetic diversity from one generation of apopulation of genetic algorithm chromosomes to the next.

It is analogous to biological mutation. Mutation alters oneor more gene values in a chromosome from its initial state.In mutation, the solution may change entirely from theprevious solution. This operator randomly flips some ofthe bits in a chromosome. For example, the string00000100 might be mutated in its second position to yield01000100. Mutation can occur at each bit position in astring with some probability, usually very small.

6) ***Fitness Function:***The fitness function is definedover the genetic representation and measures the quality ofthe represented solution. The fitness function is alwaysproblem dependent In particular, in the fields of geneticprogramming and genetic algorithms, each design solutionis commonly represented as a string of numbers referred toas a chromosome. After each round of testing, or

simulation, the idea is to delete the 'n' worst designsolutions, and to breed 'n' new ones from the best designsolutions. Each design solution, therefore, needs to beawarded a figure of merit, to indicate how close it came tomeeting the overall specification, and this is generated byapplying the fitness function to the test, or simulation,results obtained from that solution.

5. Methods used in Proposed System:

A. Input Data

The input data contains:

1) ***Professor***:Data describes the name of lecturersalong with their identification number.

2) ***Subject:***Data describes the name of courses inthe current term.

3) ***Room:***Data describes the room number and theircapacity.

4) Time intervals:It indicates starting time alongwith duration of a lecture.

B. System Constraints

System constraints are divided into 2 categories:

1) ***Hard Constraint:*** The timetable is subjected tothe following four types of hard constraints, whichmust besatisfied by a solution to be considered as a valid one:

a. A student should have only one class at a Time.

b. A Teacher should have only one class at a time.

c. A room should be booked only for one class at atime.

d. Some classes require classes to have particularequipment. For example, audio visual equipment,projectors etc.

2) Soft Constraints:These are the constraints that are of no great concern but are still taken into contemplation. They don't need to be satisfied but the solutions are generally considered to be good if they are satisfied.

a. Courses must be eventually distributed.

b. Students should not have any free time between two classes on a day.

c. Scheduling of teachers should be well spread over the week.

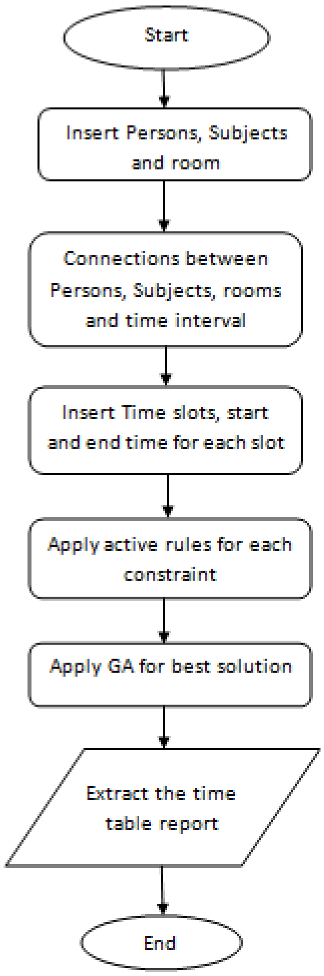

Fig 3: The structure of time table generator.

6. Future Scope: To generate timetable for the institute which will be lesstime consuming and free of human errors along with highlevel of efficiency and precision. Moreover improve theoverall process of timetable generation with help ofgenetics algorithm along with the assistance ofTechnology and Genetic is a static one in futureto work is targeted on dynamic scheduling.

7. Conclusion: As discussed, an geneticalgorithm for time tabling has been proposed. Theintention of the algorithm to generate a time-tableschedule automatically is satisfied. The algorithmincorporates a number of techniques, aimed to improve the

efficiency of the search operation. By automating this process with the help of computer assistance timetable generator can save a lot of precious time of administrators

who are involved in creating and managing various timetables of the institutes. Also the timetables generated are much more accurate,precise than the ones created manually. Used real data of variousdepartments of our institute to test the method and howeffectively it is functioning.The project reduces time consumption and the pain inframing the timetable manually. The benefits of thisapproach are simplified design and reduced developmenttime.

8.References

[1] David A.Coley, "An Introduction To Genetic Algorithms ForScientists And Engineers",World*Scientific Publishing Co. Pvt. Ltd.*,1999.

[2] DilipDatta, Kalyanmoy Deb, Carlos M. Fonseca, "Solving ClassTimetabling Problem of IIT Kanpur using Evolutionary Algorithm".*KanGAL* 2005.

[3] Sandeep Singh Rawat, Lakshmi Rajamani, "A Time tablePrediction for Technical Educational System using GeneticAlgorithm", *Journal of Theoretical and Applied InformationTechnology*,2006-2010JATIT.

[4]M. Doulaty, M. R. FeiziDerakhshi, and M. Abdi, "Timetabling: AState-of-the-Art Evolutionary Approach", *International Journal ofMachine Learning and Computing,*Vol. 3, No. 3, June 2013.

[5]AnujaChowdhary, PriyankaKakde, ShrutiDhoke,SonaliIngle,RupalRushiya, Dinesh Gawande, 'Time tableGeneration System", *International Journal of Computer Scienceand Mobile Computing*, Vol.3 Issue.2, February-2018.