International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 4 Issue: 3 11-Apr-2015,ISSN_NO: 2321-3337

COLLEGE TIMETABLE SCHEDULING USING OPTIMISTIC HYBRID SCHEDULING ALGORITHM

B.S.Yelure[1], Y.D.Chavhan[2]
Assistant Professor, Information Tech Dept., G.C.E.Karad, India [1].
Assistant Professor, Information Tech Dept., G.C.E.Karad, India [2]

.

ABSTRACT— *Scheduling is one of the important tasks encountered in real life situations. Various scheduling problems are present, like personnel scheduling, production scheduling, education timetable scheduling etc. Educational time table scheduling is a difficult task because of the many constraints that are needed to be satisfied in order to get a feasible solution. Education timetable scheduling problem is known to be NP hard. Hence, evolutionary techniques have been used to solve the timetable scheduling problem. Methodologies like Genetic Algorithms (GAs), Evolutionary Algorithms (EAs) etc have been used with mixed success. In this project, we have focus on the problem of educational timetable scheduling and solving it with hybrid scheduling algorithm. We are going to implement time table problem with policies of scheduling within hybrid scheduling algorithm. Hybrid Scheduling Algorithms, a class of evolutionary optimization techniques offer benefits of being probabilistic, requiring no auxiliary knowledge in comparison to conventional search methods such as calculus based, enumerative and random strategies. A Hybrid Scheduling Algorithm satisfying constraints that avoid clash of faculty, class room slots, etc. The automatic course scheduling system proposed to produce course timetables that truly fulfill user's needs and increase teachers' satisfaction. It is effective method for producing high quality solutions to the college course timetabling problem.*

Keywords—Genetic Algorithm (GA), Ant Colony algorithm(ACA),Constrain Based scheduling (CBS).

1, INTRODUCTION

College Time table scheduling (CTTS) represents a difficult optimization problem and finding a high quality timetable is a challenging task. It is widely used in schools, colleges and other fields of teaching and working like crash courses, coaching centers, training programs etc. In early days, time table scheduling was done manually with a single person or some group involved in task of scheduling it with their hands, which take lot of effort and time. While scheduling even the smallest constraints can take a lot of time and the case is even worse when the number of constraints or the amount of data to deal with increases. In such cases perfectly designed time table is reused for whole generation without any changes, proving to be dull in such situations.

Institutions/Schools/Colleges/Universities are the regular users of such time tables. They need to schedule their course to meet the need of current duration and facilities that are available to them. However, their schedule should meet the requirement of new course addition and newly enrolled students to fresh batches. This may result in rescheduling the entire time table once again for its entire batches and to be scheduled in shortest possible time before the batches course start.

2, RELATED THEORY

A Genetic Algorithm (GA) is a famous optimization tool in computer science. It is an intelligent search method that is inspired from biological evolution and survival of the fittest. It operates on a population of solutions, allocating trials to promising areas of the search space. A GA does not depend heavily on the information available from the underlying problem, and it can be easily hybridized to generate knowledge-augmented GA.[1] Using the operations of selection of the fittest, mutation, and crossover, GAs can quickly reach fit individuals (not always the most fit), but who are usually good enough as solutions to problems of a large magnitude. Crossover is considered as the main GA operator, which requires combining two solutions, while the mutation operator performs some small random change on a single solution. Therefore, designing an appropriate crossover operator is often more challenging than developing a mutation operator or a simple neighborhood move. This usually makes Gas implementation more difficult compared to other heuristic or meta-heuristic techniques that gradually improve only one problem solution. Using a GA to solve scheduling and timetabling problems is attractive for researchers in the heuristic and meta-heuristic field, since GAs usually perform well in a variety of hard combinatorial optimization problems.[2][3]

2.1 Ant Colony Algorithm

The Ant Colony System algorithm is an example of an Ant Colony Optimization method from the field of Swarm Intelligence, Met heuristics and Computational Intelligence. Ant Colony System is an extension to the Ant System algorithm and is related to other Ant Colony Optimization methods such as Elite Ant System, and Rank-based Ant System. The Ant Colony System algorithm is inspired by the foraging behavior of ants, specifically the pheromone communication between ants regarding a good path between the colony and a food source in an environment. This mechanism is called stigmergy. Ants initially wander randomly around their environment. Once food is located an ant will begin laying down pheromone in the environment. Numerous trips between the food and the colony are performed and if the same route is followed that leads to food then additional pheromone is laid down. Pheromone decays in the environment, so that older paths are less likely to be followed. Other ants may discover the same path to the food and in turn may follow it and also lay down pheromone. A positive feedback process routes more and more ants to productive paths that are in turn further refined through use. [7][9]

2.2 Constraint Based Algorithm

Constraint-based scheduling has become the dominant form of modeling and solving scheduling problems. Recently, due to ever more powerful embedded processors, it has become possible to embed and run constraint-based schedulers on-line even for fast processes such as product assembly sequencing. This makes constraint-based scheduling interesting to the control community as a new tool for system control, distributed and reconfigurable control, and the integration of various planning, scheduling, and control tasks. constraint-based scheduling (CBS) has become the dominant form of modeling and solving scheduling problems . CBS separates the model the description of activities, description of activities, resources, constraints, and objectives from the algorithms that solve the problem. This allows one to deal with a wider variety of constraints, facilitates the changing of the model, even dynamically, without changing the algorithms, and enables the re-use of the model for other tasks, such as simulation, planning, and diagnosis.

Constraint solving methods such as domain reduction, constraint propagation, and backtracking search have proved to be well suited for many industrial applications. Building on the CP representations and techniques introduced above, various variable and constraint types have been developed specifically for scheduling problems. Variable domains include:

1.interval domains where each value is an interval (e.g., start and duration)

2. resource variables for various classes of resources.

In scheduling applications, integer variables might be used to represent timings, interval variables to represent tasks, logical variables to represent mutual dependencies or exclusions, resource domains to denote classes of resources etc. Higher-level domains may also be defined in terms of lower- level domains; for example, interval variables are often represented as tuples of integer variables that denote start and duration of the interval.

Scheduling-specific constraints include

1. interval constraints for interval variables (e.g., $t1 \leq t2$ to express that task 1 has to occur before task 2)
2. resource constraints for timing (integer or interval) variables (e.g., allocate(r,t) for resource r and interval t to express that the task occupies resource r during interval t).[7]

2.3 Tabu Search

Tabu Search is one of the most popular met heuristics used for finding solution of combinatorial optimization problems. We now briefly describe the most basic components of Tabu Search. Starting from an initial solution, Tabu Search iteratively moves from one solution to another by searching through different parts of search space to try to find a solution that minimizes the value of the objective function i.e. a function that evaluates the cost of a solution. At each iteration, only one solution is chosen to be the current solution, and the part of the search space examined at this iteration is generated from it, this part is called the neighborhood of the current solution. The modification that makes a solution become it neighbor is a move. To prevent cycling, a Tabu list is used to store information of recent applied moves. Moves that are currently stored in Tabu list are called Tabu moves. These moves are forbidden to be used as long as they're still in Tabu list.

However, some Tabu moves might be good enough to highly improve the current solution and should be considered to be used, that means their Tabu status should be dropped, that the reason why aspiration criteria are used. At each iteration, Tabu moves are checked if they satisfy aspiration criteria, if they do, their Tabu status will be dropped immediately. In many cases, aspiration criteria that are often used is that "if the new solution created after applying a Tabu move to the current solution has the objective function's value smaller than the best solution found so far, this Tabu move will be removed from the Tabu list".[11]

3, PROPSED METHODOLOGY

3.1 Proposed System

Course scheduling system mainly includes the following six modules: security authentication module, user management module, data import module, course scheduling management module, data change module, and information query module. Its system framework as shown in figure1.

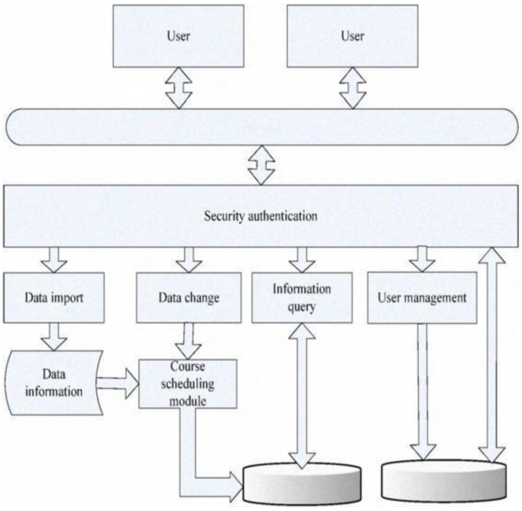

Fig 1: Course scheduling system framework diagram.

Security authentication module is responsible for security authentication to the user login information, while assigning different permissions for legitimate users. If a student user can only carry out query operation, the administrator user can have the data changes and user management permissions. That is, different user roles have different permissions. Data import module is to collect the data information of teaching plans, teacher information, time model, and then through the data format conversion, to convert them into a unified interface file data for the course scheduling module processing. Data change module can adjust the curriculum, classes, teachers, classrooms, and other information, and through the course scheduling module processing, it will achieve the addition, deletion and modification for the course scheduling database data.Information query module is to query the course schedulinginformation database based on user request information to provide the curriculum, classes, teachers, classrooms, and other information to users. User management module can add, delete and modify the user information to ensure that the information in the user information database is complete, the latest data. Course scheduling module is to generate reasonable course schedule information according to the genetic algorithm presented above, and at the same time to update the information to the course scheduling information database.

4, PROPOSED ALGORITHMS

4.1 Hybrid Algorithm for CTTS:

a) Input for the CTTS:

- Enter college name and academic year.
- Enter periods per day and number of day.
- Select period then enter start and end time also break position.
- Enter class name and number of batches e.g.I1,I2,I3.
- Enter lecture hall details :lecture hall name.
- Enter lab room detail:lab name.
- Enter tutorial room detail: tutorial name.
- Enter teacher information:teacher name.
- Enter theory subject:subject name , class to which that subject allocated , number of lecture per week.
- Enter practical detail:practical subject name, class name , number of practical.
- Enter tutorial detail:tutorial subject name, class name , number of tutorial per week.
- Enter teacher workload: name , subject , location , availability.

b) Hybrid Algorithm:

- Start.
- First truncate the entire transaction table which are required for storing temporary result.
- Read no. of days and lectures per day.
- Check teacher’s availability.
- Check teachers hour count and lecture, practical and tutorial,
- Check teachers workload on the all SE, TE and BE.
- Insert all these values i.e. teachers availability ,teachers hour count and teachers distribution in teachers priority table and give the priority to all the staff depending on above factors.
- Select the teachers having highest priority first and distribute the load into the time table column and rows.

5. CONCULSION AND FUTUREWORK

College Timetable Scheduling will simplify the process of time table generation smoothly which may otherwise needed to done using spread sheet manually possibly leading to constraints problem that are difficult to determine when time table is generated manually. In this project we primarily include the basic algorithm which firstly get all the information required like staff information, classroom information, student information etc. After we set priority for the teacher using soft and hard constraints.This project presents improvement of effectiveness in automatic college timetabling arrangement by using optimized hybrid algorithm. The algorithm was developed in enabling to obtain more optimized results.

References

[1]. Branimir Sigl, Marin Golub, Vedran Mornar," Solving Timetable Scheduling Problem Using Genetic Algorithms" 2012 IEEE.

[2]. Sehraneh Ghaemi, Mohammad Taghi Vakili, Ali Aghagolzad,"Using a Genetic Algorithm optimizer tool to solve University Timetable Scheduling Problem"2007.

[3]. Salwani Abdullah, Hamza Turabieh,"Generating University Course Timetable Using Genetic Algorithms and Local Search" 2008

[4]. E. Aycan and T. Ayav ,"Solving the CourseScheduling Problem Using Simulate Annealing" 2009

[5]. Algorithm Pariwat Khonggamnerd, Supachate Innet, "On Improvement of Effectiveness in Automatic University Timetabling Arrangement with Applied Genetic algorithm" 2009

[6]. M.V. Padmini, Member ACM, and K. Athre."Efficient Design of University Timetable"2010

[7]. Thatchai Lutuksin and Pupong Pongcharoen. "Best-Worst Ant Colony System Parameter Investigation by Using Experimental Design and Analysis for Course Timetabling Problem"2010

[8]. Negin Najdpour and Mohammad-Reza Feizi-Derakhshi "A two-phase evolutionary algorithm for the university course timetabling problem" 2010

[9]. Adalet Oner, Sel Ozcan, Derya Dengi "Optimization Of University Course Scheduling Problem With A Hybrid Artificial Bee Colony Algorithm"2011

[10]. Amir Hossein Karami1 and Maryam Hasanzadeh "University Course Timetabling Using a New Hybrid Genetic Algorithm"2012 .