College Timetabling Information System

Husain Gholoom (Instructor)
Department of Computer and Information Science
College of Business Studies
Public Authority for Applied Education and Training
huaingholoom@yahoo.com

Abstract

Course timetabling is an important operational and time consuming task in many colleges and universities. Every lecture room must be assigned to instructors and their classes , day and time for each lecture without violating any constraints [1]. The purpose of this paper is to examine timetable generation problem with regards to college classrooms , course subjects, start and end time for each subject. Several popular techniques used on course timetabling problem are discussed. The paper will introduce a brief description of the timetabling problem at college of business studies at Public Authority for Applied Education and Training in Kuwait. An automated timetabling information system framework will also be presented.

1. Introduction

Timetabling is usual problem of each academic community and it was very often tried to be solve by researchers. Although a lot of scientific and commercial work was made in this area, however , in many educational institutions, timetable is still scheduled manually and often if a computer is used, it is needed only for data presentation or checking constraint validation [2] . The term scheduling is generally described as constrained allocation of resources to objects being placed in space-time with minimal cost of set of resources used, however , timetabling is the allocation subject to constraints of giving resources to objects being placed in space-time which satisfies or nearly satisfies set of possible objectives [3] . Example of scheduling is the airplane or a train route

traveling from point ***a*** to point ***b*** with objective of minimizing the total cost function, where class timetabling and exam timetabling is examples of timetabling which all hard constraints and few soft constraints must be satisfied. Constraints are restrictions of some type such as number of lecture rooms, number of computer labs. Constraints are either hard or soft. Hard constraints are conditions that can not be violated under any circumstances and soft constraints, on the other hand are constraints that are desirable, may be violated, but should be satisfied as much as possible [4] . Many techniques has been used in order to automated timetabling problem . In the early days and still some institutions carry out the construction of time tabling manually every semester . The manual task is complicated and time consuming since the educational system became more dynamic and complicated. The need for automated timetabling system became a must. New techniques were presented such as integer and linear programming , graph coloring , genetic algorithms , memetic algorithms , local search, and constraint logic programming . It should be noticed that once the timetable is generated , the timetabling team members may interactively change the timetable that is produced by the automatic tool without violating any of the hard or soft constrains . In this paper, we present college timetabling problem and overview of automatic and interactive techniques and tools applied in solving timetabling . The paper will also presents a brief description of timetabling problem at college of business studies at public authority for applied education and training. Finally, a broad information system architecture for automated timetabling problem will also be presented.

2. Course Timetabling Problem Description

Course timetabling problem , which is one classification of university timetabling , is well known problem for researchers for some time. The timetabling problem is dealing with generally series of meetings between students, instructors in classrooms or labs (space) for a specific period of time per week (timeslots) during a semester according to satisfied constraints [5]. It is mapping between instructors meeting students in a specific location for a period of time in specific days of the week during a semester.

College of business studies at public authority for applied education and training consist of two separates main campuses, all male and all female campuses. Course timetabling is done manually each semester and each

department traditionally leads it by select few faculty members each academic year to perform the task. Since teams changes every academic year, each team may have their own different methods of timetabling depending on their professional background. Each department is assigned its own classrooms and labs by the college administration. Each department uses their own policies when constructing timetable. There are ten different department covering four primary disciplines . Once the timetable is completed by each department, it is handed to the regenerator's office where it is entered into a centralized computer for students to view and register. This task is very complex , time consuming , and it carries high potential for errors and time conflicts . It usually takes anywhere between 2 – 4 days for each department to complete the course timetabling for their faculty members. Students at College of Business Studies have no influence on how the timetable at the end should be constructed. Student have to attend courses covering all their major and minor fields according to the proposed schedule.

Courses vary when it comes to number of hours for each course. Courses carry anywhere between 1 hour to 7 hours depending on the department. For example computer science department has classes that hod anywhere between 3 to 7 hours per week. While English department have classes that carry anywhere between 1 to 4 hours per week. Each course has a department id, course id, course name, section number, number of hours. Date , time, room number, location (Male / Female college) , and a professor's name will be added during timetable construction . The following table shows some of the allowed times for scheduling classes at the computer science department.

0	S T Th 8:00 - 9:00	9	S T Th 8:00 - 10:00
1	S T Th 9:00 - 10:00	10	S T Th 10:00 - 12:00
2	S T Th 10:00 - 11:00	11	S T Th 12:00 - 14:00
3	S T Th 11:00 - 1200	12	S T Th 14:00 - 16:00
4	S T Th 12:00 - 13:00	13	M W 8:00 - 9:30
5	S T Th 13:00 - 14:00	14	M W 9:30 - 11:00
6	S T Th 14:00 - 15:00	15	M W 12:30 - 14:00
7	S T Th 15:00 - 16:00	16	M W 14:00 - 15:30
8	S T Th 16:00 - 17:00	17	M W 15:30 - 17:00

Observing the foregoing table, it is safe to assume that, a slot can be divided into half-hour. Therefore, a class scheduled on M W 8:00 - 9:30, will reserve 6 slots. Only five classrooms and nine computer labs are assigned to the computer science department.

3. Courses and Constraints

The timetabling problem is dealing with generally series of meetings between students, instructors in classrooms or labs (space) for a specific period of time per week (timeslots) during a semester according to satisfied constraints . constraints can be classified as hard or soft [5] . the following are examples of hard and soft constraints that appears at College of Business Studies at Public Authority for Applied Education and Training :-

a. Female instructors can only teach at the female college . They have high priority in timetabling for all female college [hard] .
b. Practical courses that require hands on and practice must be taught in labs [hard].
c. Instructor must teach one class at any given time and a class is taught by one instructor only[hard].
d. Conflicts : Lectures of courses in the same curriculum or taught by the same teacher must be all scheduled at different times [hard].
e. Limited number of classrooms and labs of particular type [hard].
f. Classrooms Occupancy : classrooms can not be shared in a specific time by more than one class or instructor [hard].
g. Length of classes can be scheduled for 60 , 90 , 120 minutes only [hard] .
h. Male instructors are allowed to teach in all female college [soft].
i. Instructor's teaching load varies. Department head , Phd holders, non PHD holders , and Lab Instructors all have different teaching loads [Soft]
j. Number of hours for each class : class could carry 1 . . 7 hours per week [soft]
k. Room Capacity : The number of students that attend a course must be less or equal than the number of seats of all the rooms that host its lectures [soft].

l. Classroom / Lab utilization : The daily schedule of a classroom or a lab should be as much compact as possible, avoiding unused timeslot [soft].

4. Timetabling Automation Techniques

There has been a large amount of research carried out on university and college timetabling problem. Since the size and the complexity of current universities has increased, basic domain-specific heuristic methods no longer can construct good timetables. The trend shifted toward constructing more general and automated algorithms solving timetabling problems, such as genetic algorithms, memtic algorithms , evolutionary or meta-heuristics algorithms represented in simulating annealing and tabu algorithms. Constraints Logic are also popular approaches in the area of artificial intelligence, operational research and logic programming. It seems that from the following brief description of automated mythologies, the most popular technique is usaing constraints approaches followed by the Evolutionary and Genetic algorithms. One can first use heuristics in order to constraint the search for an initial timetable solution. Then , using Genetic Algorithm as an enhanced technique on the initial solution in order to develop the solution and arrive to a final and optimal solution. Bare in mind that there is not an exact technique for an optimal timetable that is generated by an automated general timetabling system.

4.1 Genetic Algorithms

Genetic algorithms (GAs) are based on finding greatest numbers of feasible solutions then selecting the best one with the least number of constraints violations. Long bit string encoding regarding when and where each class is to take place is the most common genetic representation for a timetable. When selecting pairs of timetables they might be crossed over and as a result , the bit string is sliced and new timetables are created. Researchers have implemented intelligent systems that uses a smarter mutation operator than the cross over mythology. Other GA timetabling systems represent timetabling as an order of events and the order of events are inputted into special program which uses the orders in order to produce the timetable. GA has been implemented that deals with infeasible solutions as well. Population of

individual solution was selected and uniform cross over and transformation operators was applied. Detailed description of this method found in [1 , 6 , 10]

4.2 Memetic Algorithms

An one extension of GA's is Memetic algorithms. Memetic algorithms based on a representation of how ideas progress. The basic units of ideas are memes, which can be improved in order to reach a local optima during their lifetime unlike gene which can not improved during their life time. Thus to ensure that all population members of the timetable are at local optima, a local search method in the form of hill climbing or repairing strategy function is used at set intervals [1]. One draw back of this particular search technique is the time that the search takes in order to reach a solution. Paechter and Cumming implemented Memetic lecture scheduling system that has been used at Napier University [12, 13]. The system is based on set of events . Time slots are associated with each event. This system employed several types of transformation methodologies on these timeslots and events successful timeslots are removed and placed on top of the list allowing improvement of memetic material.

4.3 Simulated Annealing

Simulated Annealing (SA) is arbitrary search technique for finding solution to optimization problems. It is based on correlation with the physical development of annealing, which involves the aggregation of many elements in a physical system as it is cooled. Initially ,It was first used as a technique to solve hard non linear optimization problems . In this method , optimization is performed without former knowledge of problem structure or of any particular solution approach. Most of the time, this method is easy to apply because changes or moves in the solution space of the problem that is being solved are easy derived. Changes results to either uphill or downhill arrangements in the solution space. The idea is representing college timetabling by placing teachings into periods of days of the week so that number of conflicts for resources such as classes, teachers , classrooms are minimal. The algorithm is identified as geometric , adaptive , and adaptive with reheating cooling

scheduling systems are applied to college timetabling problem. Syracuse university has tested this method with real test data and the initial outcome showed that SA with adaptive cooling and reheating algorithm surpasses other methods. A complete description of this method and how it is applied to college timetabling problem is found in [6 , 10 , 14] .

4.4 Tabu Search

Similar to Simulated annealing, tabu search starts in the same way as common local search advancing iteratively from one point (solution) to another until a chosen termination criterion is satisfied. Local search are based on the notion of neighborhoods. There is a tabu list that is maintained by tabu search which represent timetables that have been visited recently and are forbidden. The purpose of this list is prevent cycling moves and prevent the search from staying in the same area and thus escape from the best possible solution. Usually tabu list is a fixed size list, When a new move is added to the list the oldest one is removed from that list so that the moves will not be revisited periodically. In some cases, tabu moves may prevent the search to proceed in order to fine new improved solution. thus, If tabu timetable best solution is reached " aspiration level " , the solution might me removed from the tabu list. More detail are found in [6 , 15 , 16 , 17]

4.5 Constraint Logic Programming

One way to solve timetabling problem is by using constraint logic programming (CLP) over finite domains. Constraints in timetabling problem are formulated as set of distinct variables in a declarative manner within the domain. The general idea is to assign values to variables satisfying all the constraints. Constraint Logic Programming can also be viewed as logic programming in which unification if being replaced by constraints handler in a constraint system. The power of this approach clearly demonstrated in languages such as Prolog , CLP , CHIP , CHR (Constraint Handling Rules) , and DOMLOG . For example, clauses in Prolog are equivalent to set of constraints in CLP , satisfying these constraints are verified during execution of the program. In constraint Logic Programming CHIP for example, the starting time for each course is represented by a domain variable and can represented by numbers. Attributes such as classroom location of a course may also be a

domain variable and can be represented as numbers. All constraints of the timetabling problem can be symbolized by built-in constraints . Search is conducted in finding solution and backtracking is performed when conflict is detected. Many software tools have been developed for automated timetabling. More details regarding these methods are in [1 , 6 , 8 , 9 , 10] .

5. An Automated Timetable Information System Framework.

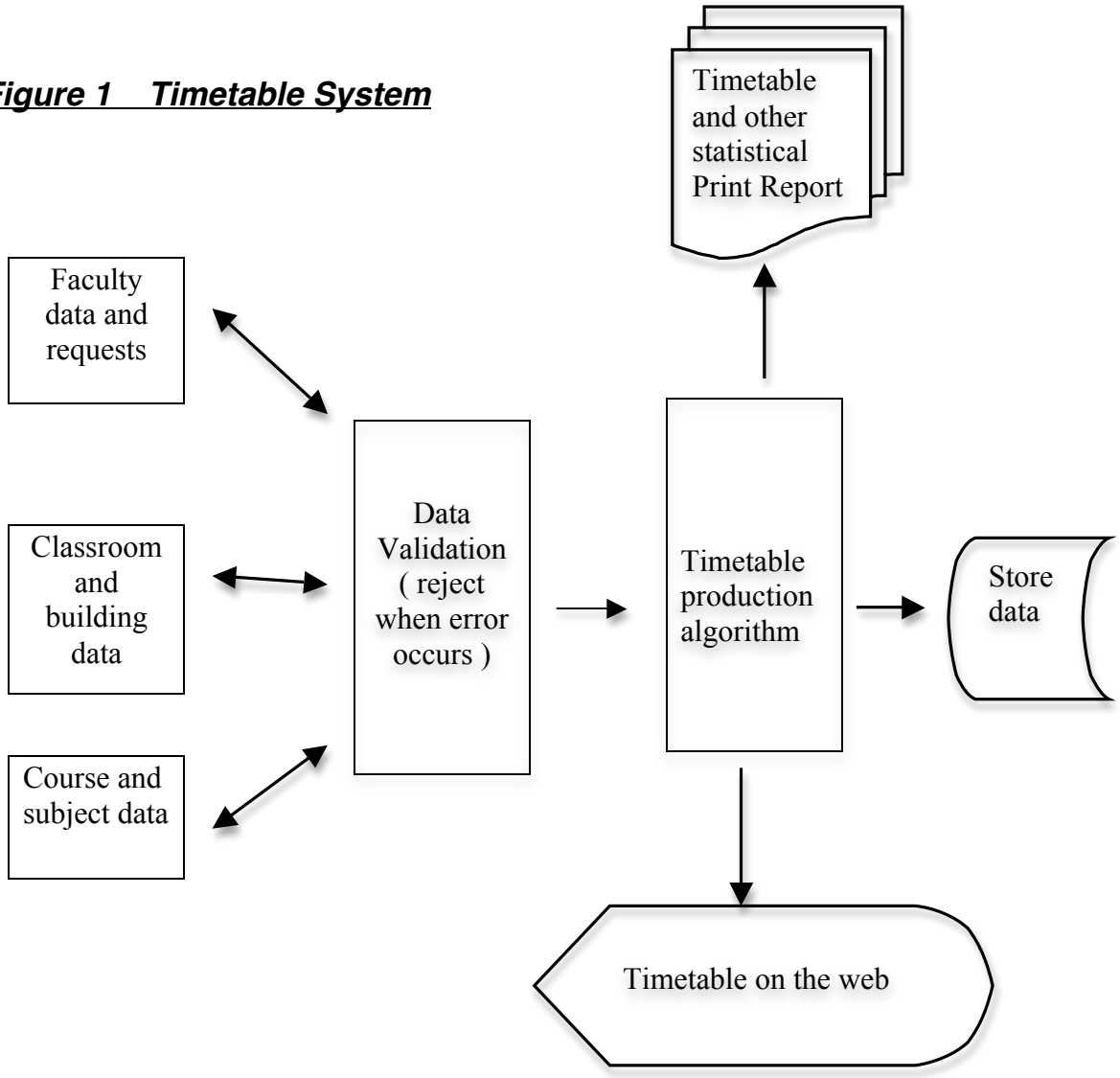
Timetabling system production is very complex task and consist of many steps such as data gathering, data entry, data verification. One of the steps in timetable production system is timetable creation. Since timetable creation is a very complex activity and time consuming, the use of software helps carrying it out and reduces time . The automated ideal timetabling production system should consist of databases to store data, web site for data entry , timetable view and maintenance, timetable algorithms for data entry, data verification and timetable construction. Files are used in order to store input data and output results. When dealing with files as input, problems dealing with data entry are encountered. Once data are entered incorrectly, unexpected behavior of the timetable creation algorithm software might be encountered or the software might terminated abnormally. Incorrect data are very hard to detect and sometime it requires to be checked manually. One way to solve this problem is automation checking and data validation upon data entry. Bad data should be rejected by the automation data verification software. Another problem is system modifications to constructed system software due to change in user requirements. The problem exist when the system is gigantic and closed. In order to avoid this problem, The system should be adaptable and easy to expand [18].

In the following sections, an open and easy to expand system workflow will be presented. The system performs data verification in the early stages, databases are used for data storage , website is used for data entry , data validation according to the specified business rules before it is stored in the data base. Website can be used as well for information retrieval.

5.1 The timetabling Production System

This system might contain two main components. The first component is the system for data handling such as data entry , verification, saving .. etc. This subsystem might contain more than one module. Each module responsible for specific task and deal with specific file or database. For example, instructors data instructors are saved separately from classes data. The second component of the system is the timetable construction algorithm. This also might contain several modules and each module is responsible for specific task. For example, printing hard copy of timetable schedule module is different from constructing the actual timetable schedule module which is also different from posting the schedule on the web module. (See figure 1 shows each department module)

Figure 1 Timetable System



5.2 Data Gathering and preparation

the process of timetable production start when at the beginning of each semester, the instructor is handed the class request form to indicate the classes that he / she would like to teach next semester. The process ends when the instructor received his / her class schedule for the next semester. The entire process takes about 2.5 to 3 weeks to complete. various data are entered into the system , for example data about the course that is offered such as time, location, number of hours , days of the week , maximum number of students allowed to enroll … etc. this information is entered into the system each semester. data about instructors are also entered into the system such as teaching status , class preference, days and time, class to be taught .. etc. external instructors are not fed into the system. Classes without assigned instructors are posted as staff and will be entered into the manually at later stage. Most of this information remains unchanged during each semester . This data must be verified and stored in data files (databases). The choice of databases are implementation decisions. Exams timetables are not part of this system.

During timetable construction, hard and soft constraints will be taken into considerations. The algorithm might run more than one time in an iterative nature until an couple of satisfying solutions are found. Once a satisfied timetable is constructed, instructors will be able to view them on paper or on the web . Modifications are done via the timetable responsible team only members. Data entry and database modifications are done via administrative staff with the timetable team authorization.

6.0 The Software Design

software design will describe the elements of the system and how these elements interacts with each other. Since the system is very complex, We are going the break down the elements of the system into small parts. This allows the system to be easier to be solved. Once each part is implemented, the combination of these parts will represent the solution to our complex system. the system will consist of three major layers, interface , logic, and data persistence layers. More example of actual implementation of such systems found in [3 , 19 , 20 , 21] .

6.1 The Interface Layer

First layer is the interface layer which presents data to the user. This layer allow the user to enter data and the data is exchanged with other layers. One typical type of data in this layer is basic data that belongs to the institution. This data must be secured and the users are not allowed to modify any part of it such name of departments, department codes, class codes, lecture room numbers .. etc. Another type of data is direct input data which is performed by administration staff. This data includes instructors preference classes to teach, preference time .. etc. The data is validated as it is entered into the system. Any error must be corrected before it is stored. Last type of data involves the timetable responsible team which monitor the system in order to produce a solution without violating any constraints. This layer only interacts with the logic layer. for example, If database change is required , it must go thru the logical layer first then the data resolution layer.

6.2 Logical Layer

The second layer is the logic layer. This layer is for data validation and allowing the data to be exchanged with other layers. The layer also responsible for enforcing the business rules of the system. business rules do vary and are defined by each institution. These rules are coded in this layer. Examples of business rules are female instructors are not allowed to teach at boys college , starting time of first class in the morning, the starting time of the last class in the afternoon . etc.

6.3 Data Resolution

This is the physical storage of data in files and databases in a specific format after it has been generated such as text files, databases format, html format .. etc.

7.0 Technology Resources

In order to implement such a system the following technology needs are required. First , A language to implement the data entry forms such as Microsoft access , oracle forms ,visual basic , html, php, and perl. Most of these languages are widely used in creating web pages and web applications. Second, the form in which data are interchanged between layers such as text files , sql , and xml. Third, language to implement the algorithm such as java . bear in mind that the language must be platform

independent. Java is widely used in many different platforms with various operating system. Finally, a tool to manage the databases such as mysql and sql database. These tools should be used by any programming languages as long as the programming language has a capability of connecting to the database via ODBC (Open DataBase Connectivity). Java has such capability by using JDBC (Java DataBase Connectivity)

When it comes to the interface layer, One must a browser that is available in every platform, a web browser such internet explorer , safari or fire Fox are found in most platforms and could be used in order to input or modify data.

In the logic layer , since web is going to be used a server is required. Apache HTTP as web server and Jakarta Tomcat as server to provide the business logic and are widely used in systems such as UNIX , Windows and MAC. Since this is a web application, security is very important.

In the data resolution layer, JDBC (Java DataBase Connectivity) could be used in order to store and retrieve data from and to database via SQL language. Sql is standard database language that is used by most database programming languages such DB2 by IBM and Oracle.

8.0 Conclusion.

The long and demanding task of creating a timetable for a college can be considerably eased by the use of computers and the automation process . However, the substantial complexity and variety of the problem means that there is much room for improvement on current systems. In this paper , we presented timetable generation problem with regards to college classrooms , course subjects, start and end time for each subject. We started by showing several popular techniques used on course timetabling problem. Such techniques were genetic algorithms, memtic algorithms , evolutionary or meta-heuristics algorithms represented in simulating annealing and tabu algorithms. Constraints Logic which is also popular approach in the area of artificial intelligence, operational research and logic programming was explored. The paper introduced briefly the timetabling problem at college of business studies at Public Authority for Applied Education and Training in Kuwait. A timetabling information system framework in attempt of solving timetabling problem was presented as well. We proposed a general framework of a timetable production system for courses. The general framework was proposes around two fundamental qualities of software development: openness and

extensibility. Full range of activities such as data gathering , data preparation, and timetable construction software system was introduced. The suggested implementation required technological resources such as languages to facilitate the data, set of business rules, programming languages to implement the rules, and finally tools to manage the data. It was clear that automation can be accomplished by using web applications and browsers such as internet explorer or firefox , databases management systems such as MySql or Oracle, and modern programming language such Java to store , process , and retrieve data and present it on paper or on the web.

Reference:

[1] Michael W. Carter , Gilbert Laporte , and Sau Yan Lee . Examination Timetabling : Algorithmic Strategies and Applications . Journal of Operational Research Society , 47, 373 – 383 (1996).

[2] Legierski, W. Widawski, R. , System of automated timetabling . Proceedings of the 25th International Conference on Information Technology Interfaces 2003 , 495 - 500 (2003)

[3] Keith Murray and Tomas Muller. Automated System for University Timetabling . PATAT 2006, pp. 536–541.

[4] Slim Abdennadher, Michael Marte .University course timetabling using constraint handling rules . Journal of Applied Artificial Intelligence , 14 : 311 – 325 , 2000.

[5] Lee, J. Yong-Yi Fanjiang Lai, L.F. , A software engineering approach to university timetabling . Proceedings. International Symposium on Multimedia Software Engineering . pages 124-131 , 2002

[6] Edmund Burke, Kirk Jackson Jeff Kingston Rupert Weare . Automated University Timetabling: The State of the Art . The computer journal , vol 40, no 9 , 1997 , pages 565 – 571.

[7] Simon Geller Timetabling at the University of Sheffield, UK – an incremental approach to timetable development , Patat 2002 .

[8] Martin Henz and Jorg Wortz , Constraint-Based Timetabling – A Case Study , Applied Aritifical Intellegence , 10:439-453, 1996

[9] Hans-joachim Goltz, Dirk Matzke , University timetabling using constraint logic programming . Practical Aspects of Declarative Languages, LNCS 1551 , Springer Verlag 1990 , pp. 320 - 334

[10] A. Schaerf. A survey of automated timetabling . Computer Science / Department of Interactive Systems , Technical Report CS-R9567 , Centrum voor Wiskunde en Informatica , 1995

[11] Juan Jose Blanco, Lina Khatib , Course scheduling as a constraint satisfaction problem (1998) by In Proc. PACT98

[12] Ben Paechter, R.C. Rankin, Andrew Cumming, Terence C. Fogarty. Timetabling the Classes of an Entire University with an Evolutionary Algorithm. Lecture Notes in Computer Science. Springer Berlin / Heidelberg . Volume 1498. Page 865 . (1998).

[13] Ben Paechter , Andrew Cumming , Michael G. Norman, and Henri Luchian . Extensions to a memetic timetabling system . Lecture Notes in Computer Science. Springer Berlin / Heidelberg. Volume 1153 , pages 251 – 265 . (1996)

[14] David Abramson , Mohan , Henry Dang . Simulated Annealing Cooling Schedules for the School Timetabling Problem . Asia-Pacific Journal of Operational Research, Volume 16 , pages 1 – 22 , (1999).

[15] A. Schaerf, Issn -x, Andrea Schaerf. Tabu search techniques for large high-school timetabling problems. In Proceedings of the Fourteenth National Conference on Artificial Intelligence (1996) .

[16] Fred Glover . Tabu search fundamentals and uses . US West Chair in Systems Science ,Graduate School of Business, University of Colorado , Boulder, Colorado . Technical Report : June 1995

[17] F Glover, E Taillard, A user's guide to tabu search. Annals of Operations Research . Springer Netherlands. Volume 41, Number 1 pages 1 – 28 , March, 1993.

[18] Michael W. Carter , Gilbert Laporte . Recent development in practical course timetabling., Second International Conference , PATAT 97 , Lecture Notes in Computer Science – 1408 , pages 2 - 19.

[19] Barry McCollum , The Implementation of a Central Timetabling System in a Large Brithish Civic Univeristy. Second International Conference , PATAT 97 , Lecture Notes in Computer Science – 1408 , pages 237 – 253.

[20] Tim B. Cooper and Jeffrey H. Kingston The Solution of Real Instances of the Timetabling Problem . Tim B. Cooper and Jeffrey H. Kingston, The Computer Journal 1993 36(7):645-653

[21] John van den Broek, Cor Hurkens, and Gerhard Woeginger . Timetabling Problems at the TU Eindhoven , European Journal of Operational Research , 2008