Development of Web based Timetabling System

Olivia Kembuan, Gladly Caren Rorimpandey, Parabelem Tinno Dolf Rompas
and Julyeta Paulina Amelia Runtuwene
Department of Informatics, Universitas Negeri Manado, Minahasa, North Sulawesi, Indonesia

Keywords: Web, Timetabling, Genetic Algorithm, Php, Mysql.

Abstract: This paper presents a case-study which applies genetic algorithm to Department of Informatics, Manado State University timetabling system. The currently timetabling method is not efficient to solve the problem. The aim of this research was to develop a web based timetabling system to optimize the resources using genetic algorithm. The algorithm was tested with the real data containing 47 combined lessons data to be scheduled into 40 timeslots and 8 rooms. The research method used is a design or experimental method. Rapid Application Development (RAD) in system development life cycle (SDLC) model was used as the system development methodology of this Timetabling system. PHP Programming language and MySQL were used in this timetabling application. The results showed that the proposed timetabling system was successfully minimize processing time and provide the optimal solution for the problem.

1 INTRODUCTION

Timetabling scheduled courses or examinations is a fundamental activity in educational institutions, these must be assigned into appropriate timeslots for students, lecturers, and classrooms subject to constraints (Burke et al, 2007).

There are several categories of timetabling in educational institution and these include master timetabling, lecture timetabling, faculty timetabling, examination timetabling, etc (Momodu and Omogbhemhe, 2014). Of all the various types, examination and lecture timetabling has more difficulties than others during the timetabling process because of the number of constraint and resources involved.

This research concentrates on the planning lecture timetable in the university.

This system presents an approach to interactive timetabling used by Department of Informatics, Manado State University timetabling system. There are 4 student groups, 47 combined lessons data to be scheduled into 40 timeslots and 8 rooms, with no constraint with the room. However, for some certain courses, there are also laboratory works to be scheduled differently. It is also making the time table harder. The details of the task mentioned that it has the requirements of each (constraint) that must be met, this causes the scheduling process to be complex in its implementation.

Carter et al. (1996) introduced various searching methods on constraint satisfaction problems and demonstrated that this technique could be applied to optimization problems. Landa (2004), explained some of the steps that must be taken in order to create an error-free timetabling in educational institutions. The steps includes: (1) every college needs to send a request for a certain number of days and time slots to the administrative unit; (2) the institution timetable officer from the administrative unit will allocate each college a certain number of days and available exam venues for each college is based on the combination of the request number, availability of resources and previous experience. Experience has shown that there are never enough resources to accommodate the demands of all the college s and departments. The timetable officer also reserves a number of “spare” slots for emergencies; (3) each college then come up with a feasible timetable using the resources they have been assigned. Some colleges, further break down these resources to departmental level and produce an assembled timetable for the entire departments in that college.

Although this research made it clear that this timetabling system experienced several setbacks like any form of error for a large number of courses.

Kembuan, O., Rorimpandey, G., Rompas, P. and Runtuwene, J.
Development of Web based Timetabling System.
DOI: 10.5220/0009010403170322
In *Proceedings of the 7th Engineering International Conference on Education, Concept and Application on Green Technology (EIC 2018)*, pages 317-322
ISBN: 978-989-758-411-4

Also the process is time consuming and takes several weeks to resolve which can slow down the academic activities if not well handled.

The currently timetabling system is very time consuming and resources optimization problems occur due to insufficient room resources and lab facilities. This method is not only inefficient in terms of time but also requires precision in the process because there are no error messages that indicate the occurrence of class collisions or mismanagement of lecture time. In addition, this process is very susceptible to errors in its implementation which, if an error occurs, can cause problems in the lecture process in the future.

We identified the necessity of an automated timetabling system. The problem with the timetabling system itself, that it has a lot of variation according with the policy of the institution. The preparation of lecture schedules in the Informatics Engineering Study Program includes determining the number of classes opened, allocating lecture halls and practice rooms, determining lecturers, determining the length of the lecture, determining the start and end hours of lectures, and determining the day of lecture.

This study aims to find a more accurate solution in the form of web-based lecture scheduling software by applied the Genetic Algorithms. This algorithm used computation using the principle of biological evolution modelling that can provide positive feedback to provide optimum results in finding solutions. This application is expected to help in scheduling lectures more efficiently, as well as minimizing the occurrence of errors that usually occur in the process of designing class schedules that are done manually.

This paper will be divided into four main parts. The first part discusses about some related works and about genetic algorithm in solving scheduling problem. The second part will be proposed the methodology that used. The third part will be architecture design of the system and discussion after implementing the system. The last part will be closed by the conclusion and also some suggestions to improve the system. PHP Programming language and MySQL were used in this timetabling application. The result showed that the proposed timetabling system was successfully minimize processing time and provide the optimal solution for the problem.

2 A GENETIC APPROACH TO THE TIMETABLING PROBLEM

A Genetic Algorithm is based on populations of solutions. Most genetic algorithms operate on a population of solutions rather than a single solution. The genetic algorithm generates other solutions, which tend to be better, by combining chromosomes, i.e. solutions, using three genetic operators that are fundamental for selection, crossover and mutation.

The genetic search begins by initializing a population of individuals. Initially a population is created by some mechanism. Then Individual solutions are selected from the population, then mate to form new solutions. The mating process, typically implemented by combining, or crossing over, genetic material from two parents to form the genetic material for one or two new solutions, confers the data from one generation of solutions to the next. Random mutation is applied periodically to promote diversity. If the new solutions are better than those in the population, the individuals in the population are replaced by the new solutions. Use of a genetic algorithm requires the definition of initialization, crossover, and mutation operators specific to the data type in the genome.

In developing a genetic algorithm, we must have in mind that its performance depends largely on the careful design and set-up of the algorithm components, mechanisms and parameters. This includes genetic encoding of solutions, initial population of solutions, evaluation of the fitness of solutions, genetic operators for the generation of new solutions and parameters such as population size, probabilities of crossover and mutation, replacement scheme and number of generations.

Genetic Algorithm itself takes long time to be executed and requires a certain machine configuration. This can be a problem for execution time. The second limit of the algorithm is the importance of the random part. Due to a huge set of solutions, the algorithm cannot guaranty to get the best result or the achievement of a certain level of fitness.

2.1 Initialization

The initialization process is done by giving the initial values of the genes with random values according to predetermined limits.

In our research approach, inside the chromosome, there is a gene for each activity in the

timetable system. This gene represents the course schedule time of the activity. So, a chromosome is actually an array of genes, each gene representing the course schedule, room and time.

Set parameters of genetic algorithm including population size, maximum iteration, mutation probability and crossover probability. Then encode an initial solution into a chromosome. Repeat this step until the number of individual equals to the population size.

To speed up the computation time, we restrict the value of maximum generation with maximum value 2.

2.2 Selection

Selection is another important factor to consider in implementing genetic algorithm. It is a procedure to select offspring from parents to the next generation. According to the general definition, the selection probability of a chromosome should show the performance measure of the chromosome in the population. Hence a parent with a higher performance has higher probabilities of being selected to next generation (Chen et al, 2008).

After generated Chromosome that has the highest fitness, this chromosome has highest probability to be selected for next generation chromosomes. We are used random number to selection process.

2.3 Crossover

Crossover is an operation to generate a new chromosome (i.e. child or offspring) from two parents. It is the main operator of Genetic Algorithm. The parent chromosome is chosen randomly and the number of chromosomes that are crossed is affected by the *crossover_rate* (ρc) parameter. The pseudo code for crossover process is given in Figure 1.

K = kromoson
cr = crossover_rate

```
begin:
    k← 0;
    while (k<populasi) do
        R[k] ← random(0-1);
        if (R[k] < cr ) then
        select Chromosome[k] as
    parent;
            end;
        k = k + 1;
    end;
end;
```

Figure 1: Crossover Pseudo-code.

2.4 Mutation

Mutation is used to produce perturbations on chromosomes in order to maintain the diversity of population. The number of chromosomes that mutated in one population is determined by the *mutation_rate* parameter. The mutation process is done by replacing gene that is randomly selected with a new value that is randomly obtained. (Chen et al, 2008).

First we must calculate the total length of gen in The population. In this case the total length generation is calculated by multiplying the number of generation in Chromosome with number of population.

3 METHOD

The research method used is a design or experimental method. Rapid Application Development (RAD) in system development life cycle (SDLC) model was used as the system development methodology of this Timetabling system.

RAD allows users to participate in an iterative design and development process. Conceptually, the project “loops” through the Design, Construction, and Acceptance phases, followed by re-Design, revised Construction, Acceptance, and so on. The RAD approach has advantages, since it usually achieves results quickly, the design is less abstract, and users have assurance that up-to-date requirements are considered. Its disadvantages include difficulty in controlling the process and ensuring the creation of an acceptable product. Because, it was having precise requirements and well understood milestones. Detail requirement

analysis was conducted at each different user category getting help of admin of the current timetable management system. RAD Stages explained in Figure 2.

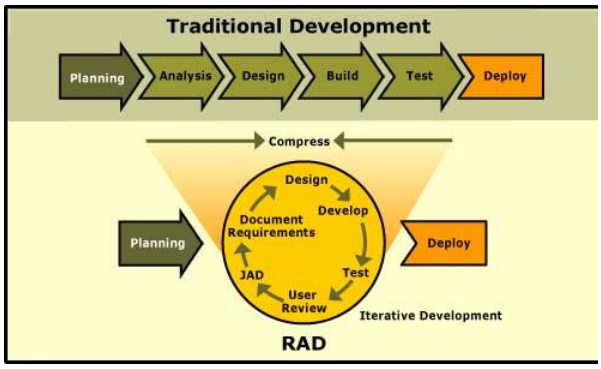

Figure 2: RAD Stages (Source from www.credata.com).

RAD uses two types of methodology [Alan D], Phased Development and Prototyping.

3.1 Stages of RAD

The Rapid Application Development path may be adapted to different CASE tools and development environments. This section briefly describes the four stages of RAD (Alan, 2009).

- Requirements Planning.
  In this stage, there are some techniques that are used to investigate requirement. There are background reading, interviewing, and observation.
  As soon this project started, an interview was set up with the Head of Informatics Department and academic staff, which provided me some internal and useful information of the scheduling system in Department of Informatics.
- User Design.
  In this stage the key users, meeting in workshops, decompose business functions and define entity types associated with the system. They complete the analysis by creating action diagrams defining the interactions between processes and data.
  In this stage, prototypes of timetabling system are developed and shown the users for review and revision. A plan for implementing the system is prepared.
- Construction.
  In the Construction stage, a small team of developers, working directly with users, finalizes the design and builds the system.
  At this stage, we started developing programs using *php* and implementing genetic algorithms in it. We also prepared documentation necessary to operate the proposed timetabling system.
- Implementation.
  The implementation stage involves implementing the new system.
  Necessary adjustments to the hardware and system software configurations are completed, and instructions given to the academic staff who will be operating the system.
  This may include implementing bridges between existing and new systems, converting data, and training users. User acceptance is the end point of the implementation stage.

4 EXPERIMENTAL RESULTS

Currently used data and method are in general based on observation and interviewed result. There are 40 time slots ranges day from Monday to Friday at time 07:30 to 13.55. In the lecture data there is a combination of course data, name of lecturer, and 4 class groups. There are 8 lecture rooms that can be used with a capacity of up to 50 students.

Due to available resources and the necessity of a web based automated system by campus; we used PHP server scripting language for coding process.

The design phase is carried out to determine how the system will be operated. This relates to determining the hardware, software, network, program display, forms and reports that will be used. In addition, it is also necessary to specify the programs, databases and files needed. This system is database using MySQL. Accessing databases using php MyAdmin, Apache servers and web browsers.

4.1 Database Design

Database is the platform used to store data in most information systems which stores the data. Database design follows a sequential order. These include the inflow schema, Relational data model (Figure 3). The database was designed using MYSQL due to the level of functionality database can offer.

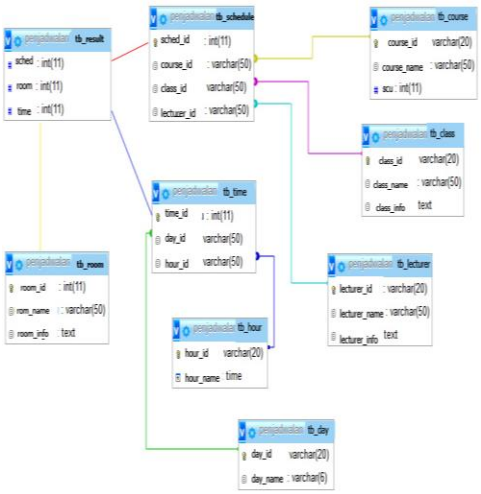

Figure 3: Relational Data Model.

File Design is a collection of interconnected records, where the file can be manipulated. In making the physical design (detail table) of a Relational Database scheme (RD), take precedence on the table containing on the side: 1) One (1), One or Many (1 / N), and 3) Many (N). So in designing the database for this Scheduling System there are or use 9 (nine) files, namely 1) tb_day, 2) tb_hour, 3) tb_time, 4) tb_course, 5) tb_lecturer, 6) tb_class, 7) tb_schedule, 8) tb_room , 9) tb_result.

Timetabling application that is designed to have one user access rights, administrator. The first output of this system is the login form. On the login page, the administrator/user can log in to the system by supplying the right username and password. On that same page (the login page) there exists a link to the password retrieval page. The screen output is shown in Figure 4. Below.

There are several management menus that can be done by the user, such as: insert lecture time data, lecturer data, course data, detailed lecture data, generating schedules and viewing scheduling results. (Figure 5 and Figure 6).

There are three variables in this timetabling problem, namely kuliah (course schedule), ruang (room), and waktu (time).

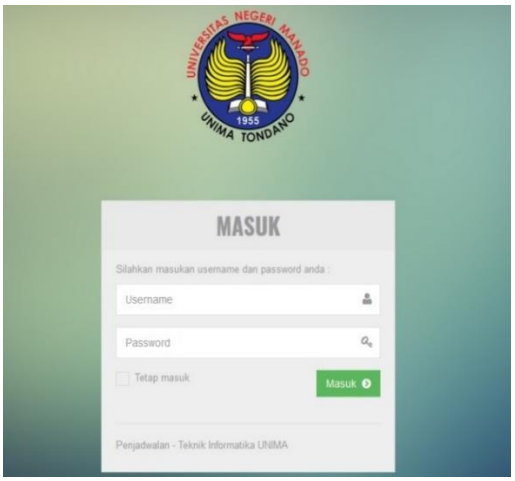

Figure 4: Login Page.

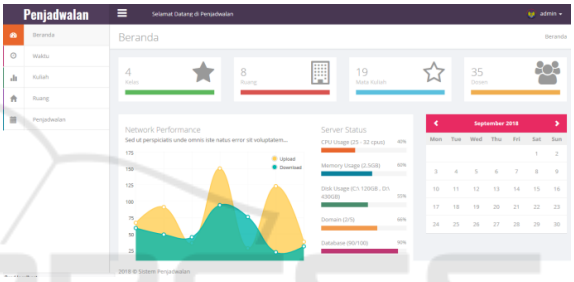

Figure 5: Management Page.

Below is the output page of the system which is university examination timetable showing the courses, the time and the respective examination venues obeying all given constraints. In the output page of timetabling system, there is a sorting feature based on day, time, subject, number of credits, class, lecture room, and lecturer name.

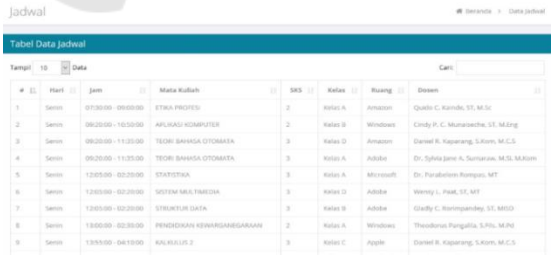

Figure 6: View Schedule Page.

In our experiment we generated data with execution time averaged 68, 55 seconds and an average memory usage of 10487.73 kb. The effectiveness of the resulting time cannot be compared to the usual manual scheduling. The manual scheduling process usually takes more than one week to process.

5 CONCLUSIONS

This paper describes part of a research project with the objective of developing a timetabling system for Department Informatics, Manado State University. Based on the research that has been done, it can be concluded that the applications that have been developed are expected to optimize and improve the efficiency of the lecture scheduling process in the Department of Informatics that was previously done manually. The result showed that the proposed timetabling system was successfully can produce schedule without class collision, minimize processing time which are provide the optimal solution for the problem.

In the future, we plan to continue with further application of the timetabling system at Manado State University. We have already started to work on the course timetabling problem for the Faculty of Engineering, where there is an intent to have the system used by all departmental schedule deputies right from the start.

ACKNOWLEDGMENTS

Our thanks go to the team of the Department of Informatics at Manado State University who helped us to get necessary data. I express my sincere thanks to Mrs. Gladly Rorimpandey, Mr. Soenandar Milian Tompunu Tengker, and Ms. Merry Liow for their encouragement that made this paper possible.

REFERENCES

Branke, J., 1999. *Evolutionary Approaches to Dynamic Optimization Problems - A Survey-, GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pp 134-137.

Burke, E. K., McCollum, B., Meisels, A., Petrovic, S.and Qu, R., 2007. *A graph-based hyper-heuristic for educational timetabling problems*. European Journal of Operational Research Vol 176, pp 177-192.

Carter, M. W. Laporte, G. and Lee, S. Y., 1996. *Examination timetabling: algorithmic strategies and applications*. Journal of Operational Research Society, Vol 47 (3), pp. 373–383.

Chen J.S., Pan J. C. H., Lin C. M., 2008. *A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem* [in] Expert Systems with Applications, Vol. 34, Chiang 2008, pp. 570-577.

Landa, J. D., 2004. *The Design of Electronic Timetabling System*, Adhor Publishers Inc, Canada.

Momodu, I. B., Omogbhemhe, M. I. 2014. Applied Science Research Journal. Vol 2 (1) 27 – 37.

Alan D. and Barbara H. W., 2009. *Systems Analysis and Design*, John Wiley & Sons, Inc. Ch.1, pp.8-14.

Zhang C., Rao Y., Li P., 2008. *An effective hybrid genetic algorithm for the job shop scheduling problem* [in] The International Journal of Advanced Manufacturing Technology, Vol. 39, 2008, pp. 965-974.