Utilization of Timetable Management System to a Medium Scaled University

ChayaAndradi, SamindaPremaratne
(Faculty of IT, University of Moratuwa, Sri Lanka)

ABSTRACT : *University timetable construction is hardworking and complicated task when there are large number of course arrays and limited resources. As a result, universities and some institutes tend to solve this issue manually even; the results may not always fully optimal. In this paper, we discuss about a framework of utilizing timetable management system to a medium scale university for resource optimization. Our endeavor through the overall research was to develop an automated timetable management system to the faculty of IT at university of Moratuwa to overcome the mentioned scheduling issues. We conducted a preliminary study and hypothesized it can be achieved by using Genetic Algorithm. In the solution, each individual called chromosome and it was evaluated using a fitness function in the implementation process. Five great Chromosomes with higher fitness value considered as optimal solution or timetable schedules. The timetable administrator can further refine the most suitable timetable. Tools such as PHP, Yii with MVC architecture and MYSQL were used in this system. Finally, this system was tested and evaluated in the university background and we suggest this framework is more desirable for medium scale universities.*

KEYWORDS -*Fitness, Genetic Algorithm, medium scale, optimal, Timetable Management System, utilize*

I. INTRODUCTION

Constructing error free timetable is a strenuous and complex task for academic institutes such as universities [1],[2],[3]. Every academic year, faculty of IT at the University of Moratuwa faces this rigorous task of preparing timetables. Although the existing manually operated, timetable system is efficient enough to carry out the courses without clashes, it is very time consuming and resource optimization problems occur due to insufficient lab resources and hall facilities. As a result, University identified the necessity of an automated timetable management system.

Current timetable management system with graph coloring heuristic technique[1] is efficient enough to carry out the courses without clashes manually. Nevertheless, problems occur due to insufficient lab resources and hall facilities. The problem is more complex when some batches have more than three hundred students while the largest hall can be allocated only two hundred and twenty two students. As a result, this research directed to resolve that real problems of an application distributing the courses and labs without collisions.

The main goal was to develop a web based Timetable Management System to optimize the resources of the IT faculty and introduce it as a scheduling framework for middle scaled universities. Moreover, investigate the available lab capacity and required resources, studying number of scheduling algorithms, conduct a comparative study of Genetic Algorithm used in other timetabling problems, study the development technologies for the automated timetabling, develop an automated web based timetable management system were main objectives of this research. Finally, the system shall be automatically generate timetables and used as a framework to solve the problem of resources optimization of IT faculty at the university. In particular, TMSFIT (Timetable Management System at Faculty of IT) was developed. This new system shall be providing the facilities of viewing the hall reservation information on the availability of the halls and laboratories by admin. Lectures and students must register through the TMSFIT before they start using the system. Hence, the security is very high, only admin of TMSFIT can update the timetable. There is an authenticating using the users passwords. The students of the other faculties cannot allow accessing the system.

Next section will review the related work, which provides context regarding issues of timetabling, certain algorithms and some more. Then, we describe methodology which contains the theory, design and implementation of the system. After that, we discuss about the system evaluation and its uses as a framework under discussion. Finally, its limitation and further work are reviewed in the conclusion.

II. RELATED WORK

According to DilipDatta and coworkers, preparation of timetable for specific university is a very complex task[2]. Therefore, they could introduce multi objective Evolutionary Algorithm based class timetable optimizer to reduce time.On the other hand, Jonathan Lee and coworkers could address some of the key challenges of timetabling by an automatic software engineering process as task–based conceptual graph (TBCG) [3]. There hard and soft constraints can be easily inserted or removed while the specifications are maintainable. However, there were some drawbacks as necessity of generalized methodology, specialists' skills while the problems are varying by concerning type of institute. Further, AnujaChowdhary and his colleagues also introduced an automated timetabling system of handling soft and hard constraints wisely with the limitations of mentioning the logic of the system[4]. In a different research, NelishiaPillay says even though there are number of researches found in timetabling few of them only developed as software[5].Their paper provides an overview of methodologies such as Bee algorithm, Constraint programming, Cyclic transfers, Evolutionary algorithms, Integer programming, Neural networks, Simulated annealing and so on. Yet another research, Edmund Bruke and coworkers, could compare and contrast some recent approaches of scheduling problems handled by the University of Nottingham [6]. As a result, they identified many present effective university-timetabling systems customized by the desired university and recent research directions in automated timetabling. Another aspect of automatic timetabling is defining constraints. Ben Peachter and his colleagues could introduce two major concepts behind them in their research[7].

However, accomplishing the algorithm construction phase of our system was most crucial factor. Because, none of the above mentioned logics matched with our requirements. Eventually, we directed our research path through the Genetic Algorithm.

2.1 Use of Genetic Algorithm

Alberto and coworkers used genetic algorithm in their research[8]. They have presented a model, a class of algorithms and a computing program for the timetable problem, with special reference to a real world application (the timetable of an Italian high school). Further, they have compared that GA-based approach with various versions of simulated annealing and tabu search by Hooshmand[9]. Finally, they conclude their experiments as GAs produced better timetables than simulated annealing, but slightly worse timetables than tabu search. An advantage of GAs over both SA and TS is that GAs gives the user the flexibility of choosing within a set of different timetables. Finally, they were identified their approach is a useful generalization of the GA and can be applied to other highly constrained combinatorial optimization problems. In a different case, Moreira could introduce a solution for problems of constructing timetables for exams using GA[10]. According to Branimir and colleagues, used GA in a different manner as algorithm performance was significantly enhanced with modification of basic genetic operators, which restrain the creation of new conflicts in the individual.

In view of Professor AshokaKarunanada, applications of GA are miracles in new technology[11]. In book of Artificial Intelligence, he has mentioned, when there is a necessity of some optimal solution such as timetabling, GA is applicable. Further, some data mining issues without having any solution and lottery games with probabilistic theory also use this algorithm. The major disadvantage of the GA is when the population is large the algorithm execution time also increasing.Chiu-Hung Chen and team workers supplies evidence with the useGA for solving multimodal manufacturing optimization problems [12]in the field of Manufacturing Robots. Creating and maintaining timetables is often a complex task for both people and software. When consider a Mimosa like commercial application, the technical side of Mimosa is kept as simple and as self-contained as possible. The technology is based on a collection of efficient optimization algorithms[13]. Moreover, some other semi-automatic timetabling software such as Open Course Timetabler[14]also free stand-alone application.

2.2 Limitations of GA

GA itself takes long time to be executed and requires a certain machine configuration. This can be a problem for time execution.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.

2.3 Problem Domain

Although there has been fair amount of researches about timetabling, few of them only considered the issues of university timetabling. According to NelishiaPillay, there is no comparative study on the success of different methodologies on timetabling problems[5]. The complexity of the timetabling is another issue. Manual scheduling generally takes number of weeks to generate timetables. Even today, there are many semi-automatic applications developed such as Mimosa, Time Tabler, still do not solve the whole problem[15]. The increasing number of students and the courses of universities also should take as an issue of timetabling[16]. Another problem occurs due to variation of constraints from one institution to another.

2.4 Technology obtained

Before the literature survey we had two options of directing this research path as whether use a rulebased system or use any algorithm such as GA. Finally, concerning the Literature review, Genetic Algorithm and some other free softwarewere selected to implement the timetabling problem of Faculty at IT the University of Moratuwa.Apache web server, MySql Database Management System, PHP and Yii with MVC architecture were compatible with each other.

III. METHODOLOGY

This section describes the approach, design and implementation of the TMSFIT as a framework.

3.1 Existing Timetabling System

Usually, the courses which are going to be offered from the faculty are approved by a senate. The lecturers in each department wish to specify preferred time on their courses. All the courses and course details must be given to admin of the timetable of university who is having the responsibility of creating near optimal timetables, which would serve as a guide for academic activities in the university. Timetable admin calls a meeting and prepares a general timetable to fetch preferred time slots from lecturers. The traditional manual timetabling system as Fig. 1.is very time-consuming and resource-intensive. Existingtimetabling process contains many steps and requires re-processing and data redundancy.

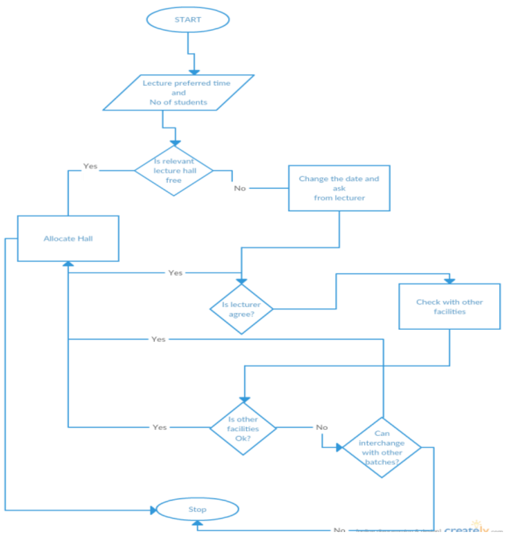

START
Lecture preferred time and No of students
Is relevant lecture hall free
Yes
No
Change the date and ask from lecturer
Allocate Hall
Yes
Is lecturer agree?
Check with other facilities
Yes
Yes
Is other facilities Ok?
No
Can interchange with other batches?
Stop
No
creately

Fig.1. Existing System

3.2 Proposed solution

Proposed solution is a website and it works as an alternative to the current timetable management system. We used Milestone approach as our research methodology. As the initial step, proper investigation could be launched about the current timetabling system with the interviews of timetable administrator, lecturers and students. As a result, pros and cons of that system could be clearly analyzed. Then, we conducted an appropriate literature survey of others work with referred to this subject to make an improved problem definition, find out technology to be used and solution. Afterwards, overall research design was constructed. Then, our TMSFIT was developed using several tools such as PHP, MYSQL, Yii, Wamp Server and some more. Consequently, we implemented, deployed and evaluated that new system using campus promises.

3.2.1 TMSFIT

TMSFIT is an abbreviation for Timetable Management System in Faculty of IT. This new system will provide the facilities for the hall reservation information on the availability of the halls laboratories in the admins module. Lectures and students must register through the TMSFIT before they start using the system. Hence, the security is very high, only admin TMSFIT can update the timetable. There will be an authenticating using the users passwords. The students of the other faculties cannot allow accessing the system.Key inputs of the system are as follows.

- The system is able to take number of inputs from the user (Admin TMSFIT) such as Student

list, Lecture list, Course list, Semester list, Hall list, Laboratory list and Timeslots.
- Various and constraints such as lecturer preferred time using web based forms.

Key outputs of the system are as follows.
- Display the generated timetable for a specific semester.
- Printable timetables
- Web based system will show the availability of the resources such as labs and courses.

Some features of the existing system are improved and some are very significant to the proposed system as follows.
- This proposed system provides an attractive graphical front-end and it is the main interaction point with user.
- The system also improves the flexibility of timetable construction.
- It will be able to generate printouts on timetabling.
- Upgraded versions of the timetable management system must be introduced
- To increase the optimization, generated timetables can be fine-tuned
- The system should save the time.

3.2.2 Application of Genetic Algorithms in This Research

The basic technology of the timetable problem is the use of the genetic algorithm to optimize a function over a discrete structure with many independent variables. Even though the timetabling problem is treated as an optimization problem, there is actually no fixed objective function. Therefore, GA can be used construction of semester based course timetables developed for the University of Moratuwa. The genetic algorithm employed combines two heuristic algorithms, the first finding a non-conflicting set of courses and the second assigning the selected course to halls and labs. The process is repeated until 500 loops and all courses have been scheduled with minimum conflicts. GA can quickly produce large populations of random feasible course timetables. Uniquely, the process takes each subject of the batch population and assigns it to the hall or a lab. The mutation and crossover procedures will then be applied to the population. The Fig.2 will illustrate the process of Genetic Algorithm.

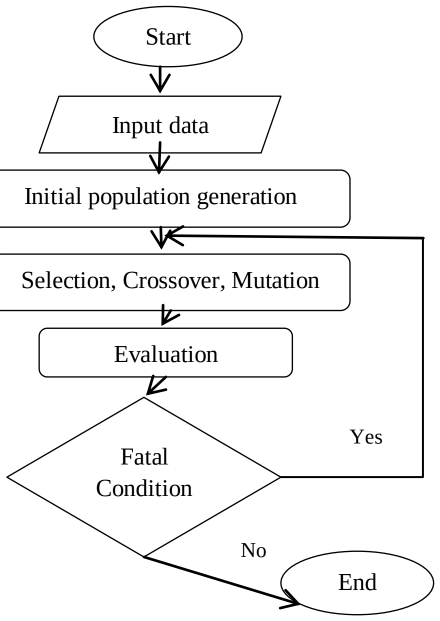


Fig.2. Process of Genetic Algorithm

3.2.3 Technology Implementation

Due to available resources and the necessity of a web based automated system by campus; we used PHP server scripting language for coding process. Moreover, Yii PHP framework with MVC architecture was used to develop the system as its ability of high performance and maintainability features. Since, MYSQL database management system is supportive for PHP; we used it as our RDBMS. Further, since WAMP server contains PHP, MYSQL and Apache web server, we used it as our local host.Eclipse for PHP plugging was successfully used in this research to modify the code with regards toits professional Integrated Development Environment (IDE).Basically, this TMSFIT was developed and installed in a personal computer with 2GB RAM, 2GHz or more processing power, 500GB Hard-disk and more.

3.2.4 Hard Constraints

Hard constraints (which can't be violated) were used to calculate the fitness value.If breaks one of the hard constraints the schedule is infeasible. They are as below.
- Room Overlap – Check if there are two lectures in one room
- Room not enough – No of students of a class is > seats of room
- Required resource not available - Does the lab have required no of Computers?

- Lecturer Overlap – One lecturer can't be in two rooms at the same time
- Student Overlap - One student can't be in two rooms at the same time

3.3 Design of the TMSFIT

We will be discussing here, what the TMSFIT does and what are the relationships among each module or level.
The system design was categorized as First level, Second level and Third level as shown in Top level design diagram in Fig. 3as below.

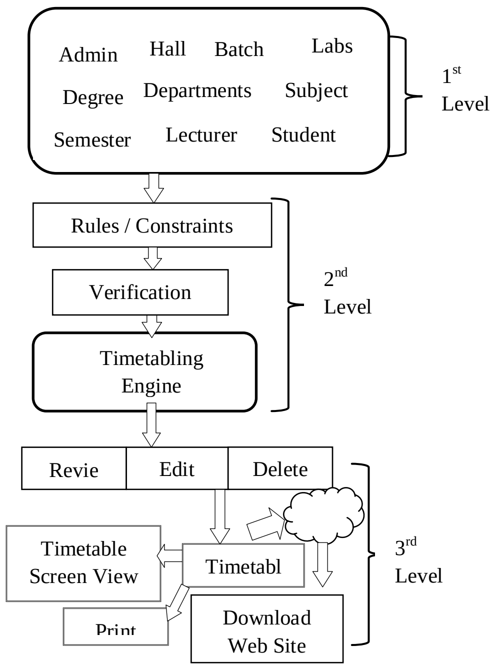


Fig.3. Top Level Design Diagram

3.3.1 First Level Module

This contains sub modules of Admin, Lecturer, Degree (course), Student, Subject, Resources and the batch. Those are in ttms database. It interacts with Timetabling Enginewhich generates timetables.

3.3.2 Timetabling Engine

The timetabling engine is primarily a web server which connects the database. It should maintain admin profile, student profile, lecturer profile, process queries; prepare outputs in various formats and so on. This is also responsible for accuracy and up-to-date information in the database. It is basically designed for maintain the system integrity, security and the privacy.This cooperates with the second level of the system.

3.3.3 Database Design of Timetable Management System

The Timetable Management System Database abbreviated as ttms. It stores data of students, lecturers, users, degree programs, subjects, timetables and some more. Student data, resources data, lecturer data, batch data, subject data and timetable data which can retrieve from the database. Admin has the authority of modifying and deleting data. Student, lecturer and course details were taken from the faculty of Information Technology at University of Moratuwa.

3.3.4 Second Level Module

This interact with the first level of the system and includes the logic of the timetable, constraints or rules, verification, timetable generation, view, delete and edit. Algorithms usually kept in this level.

3.3.5 Third Level Module

This level producescreen views of generated timetable, view the timetable on the web and print the timetable.

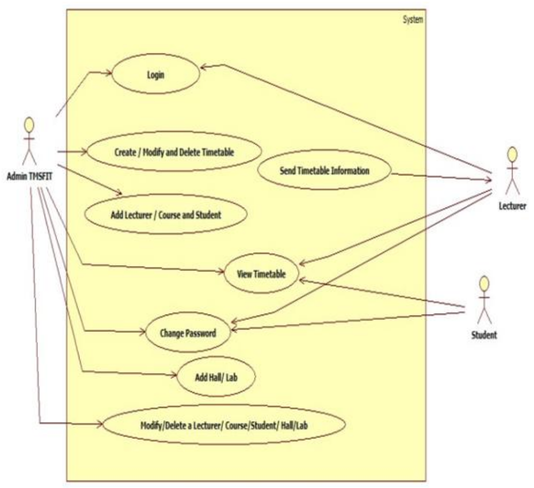

System
Login
Create / Modify and Delete Timetable
Send Timetable Information
Add Lecturer / Course and Student
View Timetable
Change Password
Add Hall/ Lab
Modify/Delete a Lecturer/ Course/Student/ Hall/Lab
Admin TMSFIT
Lecturer
Student

Fig.4. Main Use Case Design

3.3.6 Modeling the system

Use case diagramas Fig. 4describe what the system does from the standpoint of an external observer. It shows the interactions between users of the system and the system.

3.4 Implementation of TMSFIT

Waterfall model was used as the system development methodology of this TMSFIT. 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.
After the system study, the Software Requirement Specification (SRS) for the proposed system was prepared.

3.4.1 Interface Implementation

There were fifty interfaces included in this TMSFIT system. Those classes are residing in model and controller of the MVC (Model, View and Controller) architecture.Some of the classeswere associated with particular interfaces as follows.

- LoginForm, SiteController and User classes used for login interface.
- DashboardController class used for dashboard interface as Fig. 5.
- UserController class used all the interfaces with CRUD operators.
- StudentController, Student and Studentenrolsubject classes used for Create Student, Update Student, Delete Student, Manage Student, View Student Timetable

3.4.2 Database Implementation

MYSQL was used as the back end of this system. Because it includes number of engines and delivers SQL commands to operate database.As the first step of developing this system, a proper database was constructed in phpMyAdmin. In this system, it was called ttms.InnoDB engine use for foreign keys, support transactions and row level locking.According to the ER diagram in design, with main entities such as student, subject, timeslot, lecturer, degree and department main tables were generated. Moreover, batch, degree, department, employee, employeesubject, preferredtime, resource, student, studentenrolsubject, subject, timeslot, timetabletimeslot and user tables also implemented.Primary keys were assigned properly to avoid duplicate fields.In addition, user also can back up database, import database and export database any time.

Then,Process of Mapping Database tables with Model class were done. In that case,we logged on toYii code generator and used the model generator, which generates a model class for the specified database table Eg batch. Then, it generated all the appropriate user interfaces mapping with tables of the database. Later, using CURD generator, we could generate a view script file which displays a form to collect input for the specified model class. That CURD generator was important to generate controller and views.

3.4.3 Logic Implementation

Implementation of the logic of the timetable (Algorithm), constraints or rules, verification, timetable generation, view, delete and edit operations were in this second level module.Yii used AlgoritmController class and Algorithm class for the algorithm development. Hard constraints which can't be violated were reside in the calFiness() and always use to evaluate the fitness value of the timetable schedule.

3.4.3.1 Genetic Algorithm Implementation

In the initialization process of the GA, Chromosome or a class schedule must be defined first. Eg public $_chromosomes. Then, initial population was created and it is usually randomly generated 100 chromosomes as the gene or pool.In the evaluation process of the GA used to, find better individuals in each generation using fitness function as the main goal. The fitness value was calculated by calFitness()how well it fits with our desired requirements. The main operations of GA such as selection, crossover and mutation were evaluated against by fitness function.
Chromosome as Fig.5.or Schedule Evaluation done with calFitness(). If no room overlapping then increase the score by 1
If it has enough room space then increase the score by 1
If required resources are there, then increase score by 1
Check lecturer overlapping then increase score by 1
Check student overlapping then increase score by 1
Finally score of criteria should be $ci += 5;

```
FunctioninitObject($numberOfCrossoverPoints,$mutatio
nSize,$crossoverProbability,$mutationProbability,$fitnes
s,$subjectClass)
            {
                        // reserve space for time-space
slots in chromosomes code
                        $this->_slots = new
SplFixedArray(Schedule::DAYS_NUM *
Schedule::DAY_HOURS * count(Resource::model()-
>findAll()));
$this->_criteria =  newSplFixedArray( 5 *
count($subjectClass));
$this->_mutationSize = $mutationSize;
$this->_numberOfCrossoverPoints =
$numberOfCrossoverPoints;
$this->_crossoverProbability = $crossoverProbability;
```

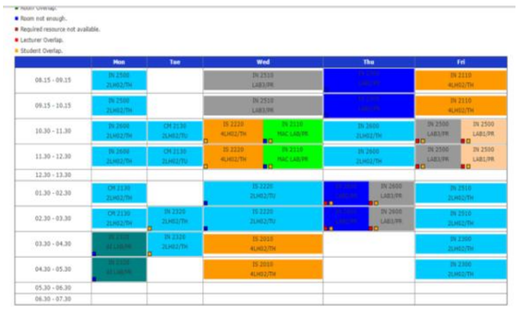
Fig.5.Code segment of Chromosome

3.4.3.2 Selection, Crossover and Mutation operations

Selection chooses superior individuals in every generation. This discards the bad designs and keeps only the best individuals in the population. Use 100 Chromosomes as this pool (gene). Then, calculate fitness value for chromosome by using calFitness() and that function uses the hard constraints and increase the score one by one 0 to 5. Then select the most suitable 5 chromosomes as timetable schedules.

Crossover operations of GA create new individuals by combining attributes of our selected individuals. As a result, crossover operator chooses two individuals from current population (parents) and creates a new individual (child) based on parents' genetic material.Here we considered no of crossover points as 2, crossover probability as 80%. Using crossover ($parent2) function, made new offspring by combining parent codes. Then, checked the fitness again using the calFitness(). If found a fitter chromosome than a previous selected one change it to new schedule.

Mutation typically works by making very small changes at random to an individual's genome.The mutation operator changes the value of some genes in an individual and helps to search other parts of problem space. With regards to this solution, our mutation probability is 3 and usedmutation() to generate new chromosomes. Then again check their fitness with previous 5 best chromosomes using calFitness() and If found a fitter chromosome than a previous selected one change it to new one. These steps, selection, crossover, and mutation, achieved in a 500 while loop.Then, best five chromosomes (Timetable Schedules)



- Room Overlap.
- Room not enough.
- Required resource not available.
- Lecturer Overlap.
- Student Overlap.

	Mon	Tue	Wed	Thu	Fri
08.15 - 09.15	IN 2500 2LH02/TH		IN 2510 LAB3/PR	[illegible]	IN 2110 4LH02/TH
09.15 - 10.15	IN 2500 2LH02/TH		IN 2510 LAB3/PR	[illegible]	IN 2110 4LH02/TH
10.30 - 11.30	IN 2600 2LH02/TH	CM 2130 2LH02/TU	IS 2220 4LH02/TH; IN 2110 MAC LAB/PR	IN 2600 2LH02/TH	IN 2500 LAB3/PR; IN 2500 LAB1/PR
11.30 - 12.30	IN 2600 2LH02/TH	CM 2130 2LH02/TU	IS 2220 4LH02/TH; IN 2110 MAC LAB/PR	IN 2600 2LH02/TH	IN 2500 LAB3/PR; IN 2500 LAB1/PR
12.30 - 13.30					
01.30 - 02.30	CM 2130 2LH02/TH		IS 2220 2LH02/TU	[illegible]; IN 2600 LAB3/PR	IN 2510 2LH02/TH
02.30 - 03.30	CM 2130 2LH02/TH	IN 2320 2LH02/TH	IS 2220 2LH02/TU	[illegible]; IN 2600 LAB3/PR	IN 2510 2LH02/TH
03.30 - 04.30	[illegible]	IN 2320 2LH02/TH	IS 2010 4LH02/TH		IN 2300 2LH02/TH
04.30 - 05.30	[illegible]		IS 2010 4LH02/TH		IN 2300 2LH02/TH
05.30 - 06.30					
06.30 - 07.30					

Fig.6. Generated timetable from TMSFIT

constructedand put them in to the flag as _bestflags ($chromosomeIndex).

3.5 Timetable Generation

Timetabling System Administrator has the authority of constructing timetables. Generating 100% optimal solution from GA is not a reality,for an automated timetable management system. Therefore, admin has to manually changethe schedule to make it more accurate.Fig.6. will show the automate TMSFIT with constraints to be changed.

As discussed, timetabling system administrator has all the responsibilities of the Timetable Management System including manual changes to the generated timetable. Further, he or she has authority to register lecturers and students to the system.By using valid username and a password, Lecturer can log on to the system, view timetable, change password, view allocated resources and mention the preferred time. Student also view timetable and change he password asrequired.

IV. EVALUATIONANDDISCUSSION

4.1 Evaluation

The main goal of this evaluation was to discuss whether the system meets the objectives defined earlier.The significance on evaluating the system was described through this system by examines the expected output and the actual output.If it satisfied our expectations, we considered that the system was behaving well.Therefore, we evaluated the system with black box testing with test data and white box testing with sample test casesusing specific software such as understanding tool. Further, we assessed the performance and robustness of the TMFIT.

For this process, we used actual student data and resources details of the faculty for the testing process. In that sense, the answers for the following questions introduced through an evaluation strategy with evaluation techniques such as interviews, observation and questionnaires could be used.Several interviews were conducted with Admin of the timetable management system, some of the lecturers and some of the undergraduate students. System deployed in parallel way and Admin staff gave their direct feedback about the system functionalities. There were more than three face-to-face interviews conducted with admin staff and some of the lecturers and during the interview their feedback about the system were noted down. System observed through sample input data for each interface, their anticipated outputs and their definite output results in the evaluation phase of the TMSFIT system. Several browser capabilities such as Google Chrome, Firefox, and Opera also successfully tested. If the system takes too much loading time, users may not satisfy about it. Therefore, loading time for all the interfaces had to be considered. Until it was hosted on a server of the campus, timetable component had to be given to admin staff. It was installed their personal computers with WAMP server. Finally, TMSFIT was further evaluated through a questionnaire by supplying that to timetable admin staff, some of the lecturers and some undergraduate students. From that, we could discover their satisfaction level of the system.

4.2. Discussion

From the admin's point of view, overall system is success. Further, this TMSFIT is time effective and user friendly. There are four batches running through the year, and we have to input all the details of them. When the quantity system data increasing, timetable generation process also gradually increasing and fitness value of the generated timetable is decreasing. As a result, additional manual work has to be handled. However, resource optimization phase is satisfied.

On the other hand, even, it's set up under probability theory; sometimes it supplies not the optimal but the best timetable schedulewhich reached the fitness value as 1. Further, we could evolve the system with customer feedback such as adding advanced searching options, view available resources and printing option. Therefore, after deploying our system at the university, continues system evaluation had to be done for use it as a framework by other universities.

The key research question raised in this work was can an Automated Timetable Management System solve the problem of resources optimization of IT faculty? Prior literature survey suggested there isn't a general way of solving these types of scheduling issues. Further, since we used only requirements of IT facultyour research area was limited. Even this is workable as a framework;application of this system to other faculties may slightly different.

V. CONCLUSIONAND FURTHER WORK

In this paper we have introduced a model or a prototype for timetabling issues of middle scaled university in Sri Lanka. Even though the timetabling problem treated as an optimization issue, there is actually no fixed objective function to solve it. Therefore, after a proper literature survey, GA was selected to construction of course timetables developed for the University of Moratuwa. This timetabling project seeks to generate near optimal timetables using the principles of genetic algorithm (selection, mutation and crossover) and it is easily understandable, less paper work,efficient and automated system, which helpful for authorities of the IT faculty.

Major limitation of this TMSFIT are, the proposed system can only generate timetables based on a few hard constraints, it gives only optimal solutions not the best solution and it only generates timetables for courses and the execution time of GA itself is high.In future, this concept can be adapted to fit the construction of examination timetables also. We suggest this timetablingsystem can be used as a framework and it will be more appropriate for medium scale universities.

VI. Acknowledgements

This effort was a research in utilization of timetable management system to faculty of it at University of Moratuwa, Sri Lanka. We acknowledge to all the senior lecturers and the staff members of the IT department at the university.Thanks also to great comments of the reviewers.

REFERENCES

[1] E. K. Burke, D. G. Elliman, and R. Weare, "A university timetabling system based on graph colouring and constraint manipulation," *J. Res. Comput. Educ.*, vol. 27, no. 1, pp. 1–18, 1994.

[2] D. Datta, K. Deb, and C. M. Fonseca, "Solving class timetabling problem of IIT Kanpur using multi-objective evolutionary algorithm," *KanGAL Rep.*, vol. 2006006, pp. 1–10, 2006.

[3] J. Lee, S.-P. Ma, L. F. Lai, N. L. Hsueh, and Y.-Y. Fanjiang, "University timetabling through conceptual modeling," *Int. J. Intell. Syst.*, vol. 20, no. 11, pp. 1137–1160, Nov. 2005.

[4] A. Chowdhary, P. Kakde, S. Dhoke, S. Ingle, R. Rushiya, and D. Gawande, "TIMETABLE GENERATION SYSTEM," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 2, 2014.

[5] N. Pillay, "A survey of school timetabling research," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 261–293, Jul. 2014.

[6] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, 2002.

[7] B. Paechter, R. C. Rankin, and A. Cumming, "Improving a lecture timetabling system for university-wide use," in *International Conference on the Practice and Theory of Automated Timetabling*, 1997, pp. 156–165.

[8] A. Colorni, M. Dorigo, and V. Maniezzo, "A genetic algorithm to solve the timetable problem," *Politec. Milano Milan Italy TR*, pp. 90–60, 1992.

[9] S. Hooshmand, M. Behshameh, and O. Hamidi, "A Tabu Search Algorithm With Efficient Diversification Strategy for High School Timetabling Problem," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 21–34, Aug. 2013.

[10] J. J. Moreira, "A system for automatic construction of Exam Timetable using Genetic Algorithms," *Rev. Estud. Politécnicos Polytech. Stud. Rev.*, vol. 6, no. 9, 2008.

[11] Mp. Professor Ashoka Karunananda Bsc. PhD, *Artificial Intelligence*, 2004.05. Tharanji Prints, Highlevel Road, Nawinna, Maharagama, 2004.

[12] C.-H. Chen, T.-K. Liu, and J.-H. Chou, "A Novel Crowding Genetic Algorithm and Its Applications to Manufacturing Robots," *IEEE Trans. Ind. Inform.*, vol. 10, no. 3, pp. 1705–1716, Aug. 2014.

[13] "Mimosa - Scheduling Software for School and University Timetables." [Online]. Available:

http://www.mimosasoftware.com/. [Accessed: 08-Mar-2016].

[14] "Open Course Timetabler 0.8.1 - Free download." [Online]. Available: http://open-course-timetabler.soft112.com/. [Accessed: 19-Apr-2016].

[15] L. Carpente, A. Cerdeira-Pena, G. de Bernardo, and D. Seco, "An Integrated System for School Timetabling.," in *ICAART (1)*, 2011, pp. 599–603.

[16] J. J. Moreira, "A system for automatic construction of Exam Timetable using Genetic Algorithms," *Rev. Estud. Politécnicos Polytech. Stud. Rev.*, vol. 6, no. 9, 2008.