Using Genetic Algorithm For Solving Time Tabling Multidimensional Issues and its Performance Testing

Majlinda Fetaji[1], Bekim Fetaji[1], Mirlinda Ebibi[1]
[1]*South East European University, Contemporary Sciences and Technologies, Ilindenska bb, 1200 Tetovo, Macedonia*

Abstract. **The focus of this research is analysing the possibility of applying genetic algorithm concept into time tabling or otherwise known as scheduling. In order to test the concept a desktop software solution is created and tested its performance. The contribution of the research study is the proposed approach with the use of genetic algorithms when developing time tabling software solution. Findings regarding the allocation of resources over time to perform a collection of tasks, while taking all constraints into account have been stated and recomendations are provided.**

Keywords. **Genetic Algorithm, Time Tabling, Scheduling, Testing.**

1. Introduction

The focus of this research study is the investigation into multidimensional specifics of time tabling and scheduling and finding the most apropriate algorithm for a software solution.

In order to test our concept a software solution has been developed. Insights and recomendations have been proposed.

The creation of a teaching schedule is a complex problem that is part of NP-hard problems. As it is known the time required for solving this type of problems increases exponentially with the problem size. In order to generate this type of schedules we must choose the right optimization techniques that produce optimal solutions in an acceptable time depending on size. The most efficient methods for resolving this type of problems are genetic algorithms.

Teaching schedule is a highly complex problem which is part of the field of scheduling. Scheduling is defined as "allocation of resources over time to perform a collection of tasks, while taking all constraints into account, so that input demands are met in a timely and cost-effective manner" [1].

There are a lot of categories of scheduling inside school institutions, such as teaching, exam and student scheduling.

Timetabling is a multi-dimensional assignment problem, in which students and teachers (or faculty members) are assigned to courses, course sections or classes and events (individual meetings between students and teachers) are assigned to classrooms and time slots. [2]

Traditionally, timetables have been constructed by hand and then modified as appropriate each year (a process known as local repair). This is a laborious process and it would be desirable to automate it in some way.

The usage of many dimensions increases the time needed for creating the schedule, so in order to minimize scheduling conflicts it's better to use genetic algorithms. During this study are listed other reasons why to choose this algorithms for solving this type of problems. The constraints can be divided into two groups: hard constraints and soft constraints. The schedule has to satisfy all hard constraints in order to be feasible and it should satisfy as much as possible soft constraints in order to be good quality.

2. Review of Timetabling and Scheduling

The term scheduling is used a lot and in different computing areas. Firstly it was used in operating systems. According to Pinedo "Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives." [3] For different real-life domains may be different resources (runways at an airport, crews at a construction site, processing units in a computing environment, etc), different tasks (operations in a production process, take-offs and landings at an airport, stages in a construction project, executions of computer programs, etc) and different objectives (the minimization of the completion time of the last task , the minimization of

the number of tasks completed after their respective due dates, etc). One of these domains that require scheduling is the ed¬ucation sector (i.e. courses timetabling and exam timetabling). These kinds of problems are a continuous challenge for Artificial Intelligence and Operational Research as this kind of problems are very difficult to solve for the reasons stated below:

- Scheduling and timetabling problems are NP-hard, and that they cannot be solved in polynomial time using a deterministic algorithm.
- Complexity of scheduling task is totally dependent on the number of instances and constraints. But in case of limited resources and strict time and cost restrictions, it is very hard to formulate schedules.

Scheduling, as a decision-making process, plays an important role in most manufacturing and production systems as well as in most information processing environments. It is also important in transportation and distribution settings and in other types of service industries. [3]

3. Review and Analyses of Genetic Algorithms

"Genetic algorithms are a type of iterative mathematical modeling technique used to find the optimal combinatorial state given a set of parameters of interest. Usage of the term "genetic" refers to the mechanism of the algorithm where, through the process of iteration, individual models "evolve" over time and compete with each other in a Darwinian fashion where the fittest model emerges as the solution, similar to how chromosomes evolve to create speciation." [4]

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. [5]

Genetic Algorithms (GAs) was invented by John Holland and developed this idea in his book "Adaptation in natural and artificial systems" in the year 1975. Holland proposed GA as a heuristic method based on "Survival of the fittest". GA was discovered as a useful tool for search and optimization problems. [6].

The real challenge in implementing genetic algorithm is usually to implement the fitness function, which contains all the knowledge of the problem to be solved. "The fitness of an individual in a genetic algorithm is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one." [6]

In case of multi-objective or Pareto optimization there is not any one number to describe the quality but several different qualities. The user is supposed to choose the proper quality combination that best fit to his/her purposes. Synonyms of fitness function include: cost function, score, and ranking.

4. Problem Definition

Creating a schedule (timetable) for a university means finding the right combination between a group of events (courses to be taught in a time period) and a group of resources (teachers, rooms, classes) by fulfilling the group of constraints given.

In this section, we give a description of the scheduling problem addressed in this study. This scheduling problem arises in a university in Tetovo. Faculty of CST - Contemporary Sciences and Technologies is a large faculty in terms of the number of teachers and the number of students and that's why we will focus on this faculty.

In order to graduate from this faculty, each student has to complete a number of courses, where some courses are elective and some others

obligatory. Here, a course refers to a subject taught one or more times within a week, and each course has a certain number of credits.

There are two types of teachers: full time and part time teachers. For this faculty, some courses could be taught jointly by part time and full time teachers. The part time teachers' requirements, such as the course can only be taught during certain time periods have to be satisfied. For the full time teachers, they are further divided into groups such as administrative positions (Vice Dean, Dean, etc) and other full time teachers who do not hold any key administrative position.

In this study, we focus on courses which can be taught by either full time teachers only, or taught jointly by full time and part time teachers. The teacher assignment problem involves assigning and scheduling the teachers to the courses by taking some factors, such as the teacher's abilities and number of courses offered, into consideration.

The objective of the software solution is the creation of a weekly teaching schedule for different categories of teachers in the university mentioned before. Scheduling problems are considered as multi objective optimization problems. If the number of the events and resources used in a university is not very large the schedules can be created manually, but if the number of students and courses increases and the university has lack of rooms the problem becomes very difficult to be solved manually. A schedule created manually will be difficult to modify. In order to simplify the schedule creation process for large universities we have chosen to solve it with genetic algorithms as one of the best techniques for this kind of optimization problems.

There are various types of constraints that mostly vary from the type of problem that will be scheduled. Timetable problems are subject to many constraints that are usually divided into two categories: "hard" and "soft". Hard constraints are rigidly enforced and have to be satisfied in order the timetable (schedule) to be feasible. Soft constraints are those that are desirable but not absolutely essential, because usually it is impossible to satisfy all of them. [10]. A schedule is not useful if it does not satisfy hard constraints, and it is not optimal if it does not satisfy soft constraints. Below are given some of examples of hard and soft constraints taken in consideration during this work.

HARD CONSTRAINTS
The room is big enough for all the attending students and satisfies all the features required by the event
No resource (teacher, student or classroom) is assigned to different events at the same time
There cannot be more than 2 classes for a subject on one day
For each time period there should be sufficient resources (e.g. rooms and lecturers) available for all the events that have been scheduled for that time period
Room capacities not exceeded
Maximum number of time periods per day should not exceed particular value
Each lecture is held in a classroom belonging to specific set of valid rooms for lecture
Each classroom has its own availability schedule
Each lecture is assigned to a teacher that belongs to specific set of teachers that can deliver lecture
SOFT CONSTRAINTS
A particular class may need to be scheduled in a particular time period
A student has a class in the last time slot of a day.
A student has more than two classes in a row
Each teacher has its own availability schedule ensuring he submits plan with desirable time periods that suits him best
Classrooms should not be booked which are much larger than the size of the class
Every staff should get at least one first hour
Classes should be scheduled within preferred hours
Every teacher has minimum and maximum limit of weekly work-hours.
Most students do not wish to have empty periods in their timetable

Table 1. Hard and soft constraints for the teaching schedule

The teaching schedule is formulated as follows: Problem = {T, C, E, R, CC, W}, where T = {T1, T2,…} is a set of teachers, C = {C1,C2,…} is a set of courses, E = {E1,E2,..} is a set of timeslots with a day of week and a time period, where the day of week is Monday to Friday and the time period is 1 to 5. R = {R1, R2, ...} is the set of classrooms and CC= {CC1, CC2,..} is a set of constraints, W= {W1,W2,…} is a set of weights for each constraint.

5. Development and Implementation

Partitioning the application into logical modules is the first step of the overall problem. The architecture chosen for developing this software is a multi-tier architecture, exactly three-tier architecture, which are:

- Data Access Tier (TSADataAcces)
- Business Logic Tier (TSABusiness)
- User Interface Tier (TSAUserInterface)

The working logic and relationship between these layers is explained below.

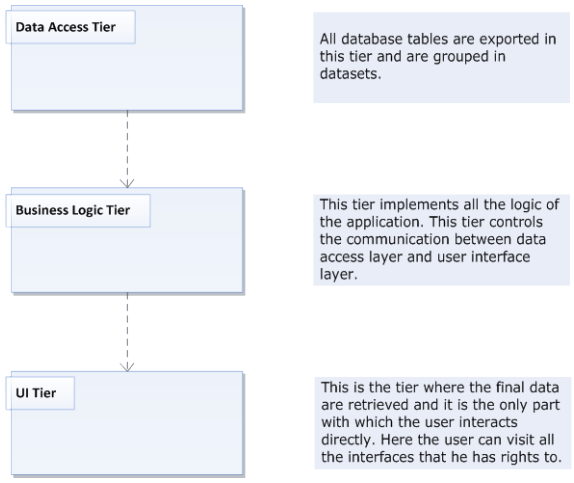

Figure 1. Software Architecture

Inside Data Access Layer are created three different datasets. This division is done by taking in consideration the group of functionalities that we have decided to implement in this application.

- Student Dataset
- Teacher DataSet
- TSA DataSet

The datasets have been grouped almost with the same logic as entity relationships. Each dataset contains data tables, table adapter (save the main queries used in the program) and the relations between data tables. The relations are the same as in ER diagrams explained above. All the queries listed on data table adapters are used in Business Layer Tier.

For each table in the database is created an object in the business layer. DomainEntity is one of the basic classes of this layer and this class is not written by me.

Does not exist any reference where to find this class because it is not in the internet or in any book. All the other classes inherit from this basic class.

For all the classes used the structure is the same:

- Is declared one data table, one row and one table adapter variable.
- Are declared two constructers that have different parameters.
- For each field in the corresponding table a specific class is created a property that is of the same type as the field in the database table.
- 3 override methods (Load, Save, Delete) are used .This method override the exact methods of DomainEntity class.
- For each query in the table adapter that corresponds the specific class is created a method.

The user interface layer has these main modules:

1. Teachers Module - the user can make the basic actions with teacher's data (Add/Modify/Delete data). A teacher can be deleted if does not exist any event connected with him/her.

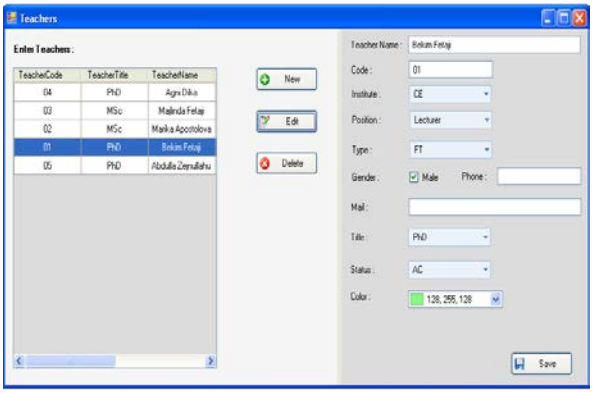

Figure 2. Teachers Module

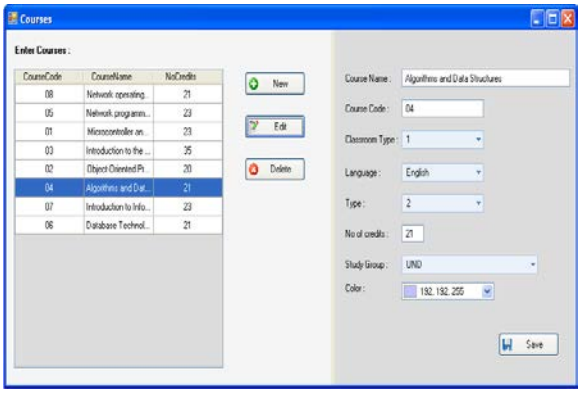

Figure 3. Course Module

2. Courses Module - the user can make the basic actions with courses data (Add/Modify/Delete data). A course can be deleted if does not exist any event connected with it.

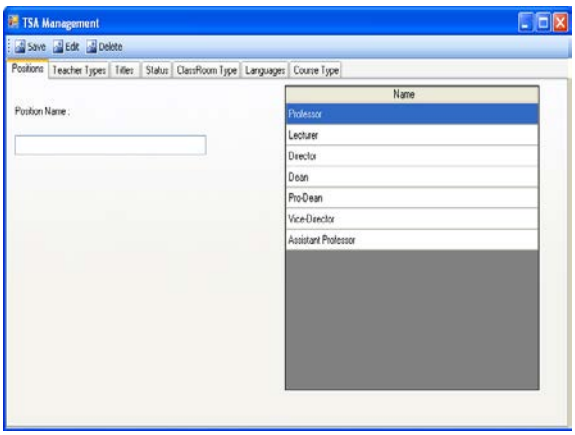

Figure 4. Management Module

After having finished all the necessary adjustments on teaching tasks Generate Schedule Module is the final module that the user must access to finish the whole process.

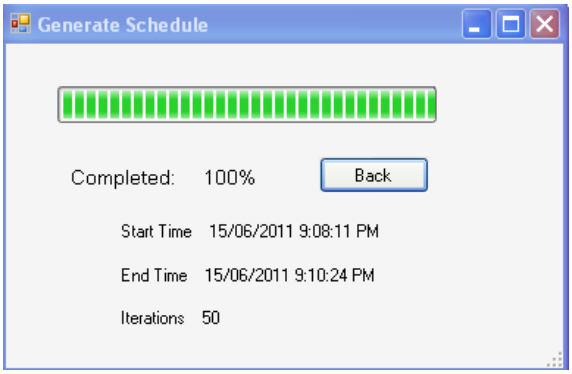

Figure 5. Generate Schedule after module entries

6. Testing Results

There are a lot of testing realized and most importantly the Incremental integration testing, using bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. Done by programmers or by testers as described in [4].

This testing is done during the software development. The test is done in the following order, Dataset testing, class testing and modules testing.

Inside Datasets is implemented the database logic. All the database connections, data result set retrieval operations, connection pooling, checking the data is performed in these classes. Dataset testing is illustrated below. Here is testing if the database accepts TeacherCode longer than 2 characters, as the MaxLength property of the Teacher Dataset requires. The test shows that the software outputs an error if TeacherCode is not correct.

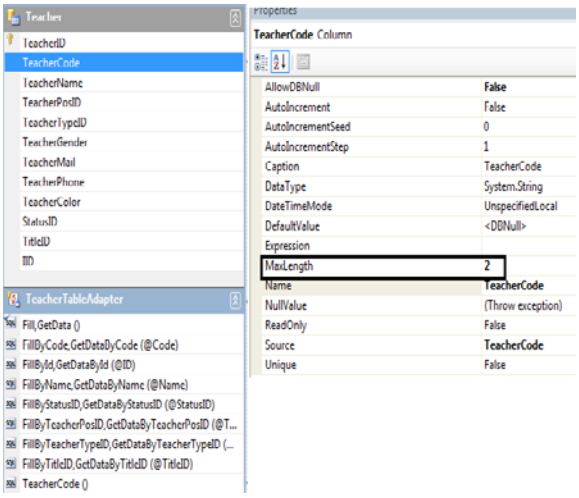

Figure 6. Dataset restrictions

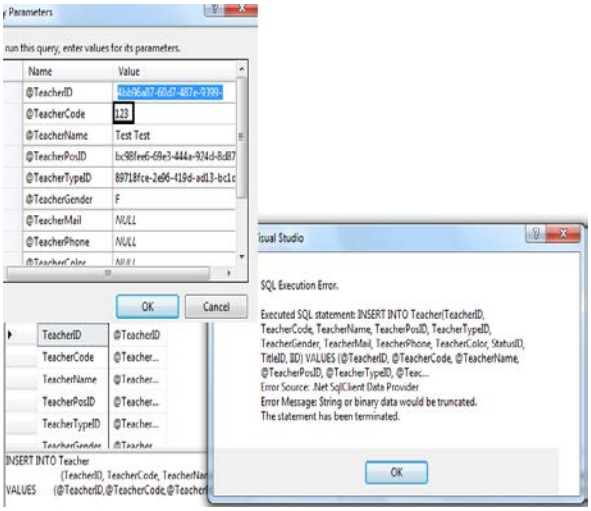

Figure 7. Execution Testing

The modules testing stage depends on the business logic tier. Because if the logic of the business is implemented right, also the information will retrieve without errors. Each action in User Interface Tier is related to a method of business objects, that is related with DataAcces Tier (Figure 23).

Stress testing – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load. [21]

Stress testing is done in order to test the software performance. These tests show that if the user

increases the data in the EduTask tables it requires much more time for the final result to be shown.

Usability testing – User-friendliness check. Application flow is tested, Can new user understand the application easily. [21]

This type of testing is done in parallel with modules testing. Is tested if it's easy for the user to understand how is data entry done in this application, and how can they see the software output. Each step has its own messages, in both cases of output, correct or incorrect one.

7. Conclusion

The research study contributes with genetic algorithms approach in developing time tabling automated solutions that according to the testing results and user feedback has been shown effective.

This study attempted to resolve some of the definitional and methodological difficulties encountered by previous researchers. It involved review and comparative analyses of algorithms most suited for solving the problem with time tabling and scheduling.

The difficulties for achieving a best performing algorithm have made it almost impossible to have an advanced graphical user interface.

- This work has been very useful, because inside it we can find a very good combination between software engineering and artificial intelligence.
- Inside this software is implemented a model that allows one course to be taught by different teachers, and to be in different languages.
- The final scheduling interface is created in C# especially for this project.

The model used inside this work has been tested for real data obtained from a university and some data created randomly. The software shows that for the real data the proposed model creates faster timetable results than those timetables that can be created manually. But the model needs still some other improvements in the fitness function in order to have better results.

The project did meet the main objective to analyze and develop a teaching scheduling application. The student scheduling part isn't implemented in this solution, but this application has the basics (database logic) for satisfying this request that is planed as future work.

References

[1] Fetaji M. Fetaji, B. (2008)."Usability testing and evaluation of a mobile software solution: A case study" – IEEE conference, ITI, Dubrovnik, Croatia, 23-26 June 2008, pp (501 - 506).

[2] Hossam Faris, Alaa Sheta, Ahmed Tobal. A parallel genetic algorithm for solving timetabling problem. ICGST-AIML Journal, ISSN: 1687-4846, Volume 8, Issue II,September 2008.

[3] Lovrekovic, Z, Lovrekovic, T., (2012). "Software Application for Managing Multidisciplinary and Interdisciplinary Projects as a Part of Educational Process", Journal TEM, Technology Education Management Informatics, ISSN: 2217-8309, Volume 1, Number 1, 2012, Association for Information and Communication Technology Education and Sciences, Serbia.

[4] Rita Gaidukeviciene, Eugenijus Kurilovas. Comparative Study of Profiled School Scheduling Programs in Lithuania. Informatics in Education, 2005, Vol. 4, No. 1,19–42

[5] Sadaf Naseem Jat, Shengxiang Yang. A Guided Search Genetic Algorithm for the University Course Time tabling Problem. Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2009).

Corresponding author: *Majlinda Fetaji*
Institution: *South East European University,Macedonia.*
E-mail: *m.fetaji@seeu.edu.mk*