Project Report For Solving a Faculty's Timetable Scheduling Problem Using Genetic Algorithms

**سلمى مجدى على ربيع**

**ID:202000392**

**رحمه هاشم داهش عبدالرحمن**

**ID:202000316**

**ريم اسماعيل حسن احمد**

**ID:** **202000346**

**شروق كرم سيد محمد**

**ID:** **202000431**

**شروق وليد السيد فرج**

**ID:** **202000432**

**شروق حسين محمد محمد**

**ID:** **202000428**

1. **Project idea:**

### Solving a Faculty's Timetable Scheduling Problem using Genetic Algorithms.

### Timetable scheduling is the process of creating timetables that fit the constraint of the scenario*.*

## ***Constraints***

### No teacher can hold two classes at the same time

### room capacity accommodates max number of students

### No classroom can receive two classes at the same time

## -**Input data:** for each class are departments, courses, room, meeting time and instructors.

## -**Output data:** schedule.

### In the algorithm presented here, each individual in the population represents one timetable. The algorithm starts from an infeasible timetable and tries to get the feasible one.

### That is why the feasibility function will mainly try to minimize the number of conflicts in an individual.

### Individuals in a population are sorted by ascending value, so the best individual has the smallest fitness value.

## **Main functionalities**

### In the algorithm presented here, each individual in the population represents one timetable. The algorithm starts from an infeasible timetable and tries to get the feasible one.

### Diagram Description automatically generated

### 

**(3) Similar applications in the market**

**\* GHC WEB AND MOBILE APP \***

**-the functionalities/features:**

* Timetable framework
* Make different timetables each week
* set up classrooms
* add class units
* Properties of the class units
* set up the teachers’ options
* Debugging impossible conditions
* Assessment board meetings

**-How does GHC work?**

* **Planner:**

Set up all the necessary configurations of your Faculty timetable: class units, teacher availability, relationship between lessons.

* **Engine:** the GHC engine automatically finds and optimizes the solution for your schedule**.**
* **Editor:**  View your schedules in different ways. Make changes easily thanks to collision detection. Print and export the school timetable in multiple formats and academic managers.
* **GHC web app& mobile:** Publication of schedules between dates. Management of teacher absences and substitutions. School, teachers and students always communicated and with their schedule on hand.

1. **An initial literature review**

[*Solving timetable scheduling problem using genetic algorithms*](http://www.zemris.fer.hr/~golub/clanci/iti2003.pdf)*.*

### [*B. Sigl*](https://www.semanticscholar.org/author/B.-Sigl/70325594) *,* [*M. Golub*](https://www.semanticscholar.org/author/M.-Golub/49091921)*,* [*V. Mornar*](https://www.semanticscholar.org/author/V.-Mornar/1721359)*:*

- Published 16 June 2003

- Computer Science

- Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.

A genetic algorithm for solving a timetable scheduling problem is described. the algorithm was tested on small and large instances of the problem. Algorithm performance was significantly enhanced with modification of basic genetic operators. Intelligent operators restrain the creation of new conflicts in the individual and improve the overall algorithm 's behavior.

### Significant improvements have been achieved by using intelligent operators. The intelligent algorithm converges much faster than the basic algorithm and represents agood starting point for complete solving of the full scale problem.

[solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable](https://www.researchgate.net/publication/221927228_Solving_Timetable_Problem_by_Genetic_Algorithm_and_Heuristic_Search_Case_Study_Universitas_Pelita_Harapan_Timetable)

### [SamuelLukas](https://www.researchgate.net/profile/Samuel-Lukas)*,* [Arnold Aribowo](https://www.researchgate.net/profile/Arnold-Aribowo)and[MilyandreanaMuchri](https://www.researchgate.net/profile/Milyandreana-Muchri)*:*

### *-* published March 2012

### - Faculty of Computer Science, Universitas Pelita Harapan, Indonesia.

### 

### Scheduling problem is a model of complicated problem. Too many things have to be considered in order to arrange a schedule, such as lecturer availabilities, a great number of classes and courses. To overcome this problem, genetic algorithm combined with heuristic search is proposed in this paper. This proposed method was tested several times, and the results show that despite small population, the best schedule still can be obtained.

[Solving Timetabling Problem Using Genetic and Heuristic Algorithms](https://ieeexplore.ieee.org/abstract/document/4287899/citations?tabFilter=papers#citations)

### [N. D. Thanh](https://ieeexplore.ieee.org/author/38185077700):

### -Published 2007

### -Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007).

### In this paper, we propose a hybrid algorithm that combines genetic and heuristic approach. By using this method, solving timetabling problem is converted to finding the optimal arrangement of elements on a 2D matrix. This algorithm was implemented and tested with the synthetic and real data of Nong lam University of HCM City, Vietnam. The experimental results reveal the usability and potential of the proposed algorithm in solving timetabling problems.

[Optimize Timetabling Problem Using Improved Genetic Algorithm](https://ieeexplore.ieee.org/abstract/document/4810475)

* [**Wang Yun**](https://www.semanticscholar.org/author/Wang-Yun/46852885)**,** [**W. Kun**](https://www.semanticscholar.org/author/W.-Kun/2074383443)**,** [**Wang Xiang Yun**](https://www.semanticscholar.org/author/Wang-Xiang-Yun/2071794730)**:**

### -Published 1 December 2008

### - Computer Science

### - 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop.

### The arrange timetable algorithms in common need to further improve. Aiming at the problem, classes divide into groups and elitist strategy with dissimilarity chromosome methods based on genetic algorithms (GA) was proposed. Compared with standard genetic algorithms by simulation, its efficiency was demonstrated. The experimental results show that it illustrates a good prospect of application and extension.

[*Interactive timetabling system using genetic algorithms*](https://www.researchgate.net/publication/228793874_Interactive_Timetabling_System_Using_Knowledge-Based_Genetic_Algorit)

### [*H. Kanoh*](https://www.semanticscholar.org/author/H.-Kanoh/27015156)*,* [*Y. Sakamoto*](https://www.semanticscholar.org/author/Y.-Sakamoto/2639988):

### -  Published 10 October 2004

### - Computer Science 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583).

### This paper discusses a new solution to university course timetabling problems. Problems that belong to the NP-hard class are very difficult to solve using conventional optimization techniques. Our solution methodology is based on genetic algorithms which use an installed knowledge base. The knowledge here is a set of candidate partial solutions of the final solution. The proposed method is to use both a knowledge base and constraints to solve the problems efficiently. The timetables obtained can satisfy teachers' personal requests and preserve the advantages of past timetables. Experiments using timetables of University of Tsukuba showed that this approach is an effective solution method. The proposed method includes general techniques concerning the use of domain specific knowledge that can be applied to a variety of large-scale real-life combinatorial optimization problems.

### An Application of Genetic Algorithm for University Course Timetabling Problem

* [**R. Sanjay**](https://www.semanticscholar.org/author/R.-Sanjay/143789218)**,** [**S. Rajan**](https://www.semanticscholar.org/author/S.-Rajan/2061132802)**:**

### - Published 6 August 2016

### - Computer Science International Journal of Applied Information Systems

### Timetabling problem is a process of assigning given set of events and resources to the limited space and time under hard constraints which are rigidly enforced and soft constraints which are satisfied as nearly as possible. As a kind of timetabling problems, University course timetabling is a very important administrative activity for a wide variety of institutes. Genetic algorithm is an advanced heuristics method which is very effective in many areas. It is frequently deployed meta-heuristics algorithm to solve difficult combinatorial optimization problems. In this paper, genetic algorithm is used to solve university course timetabling problem. At first, a model of problem to be solved is defined. Then, the genetic representation is determined and a fitness function is established according to the constraints. Finally, a case of university course timetabling from real world is discussed and solved. It is demonstrated that the method proposed in this paper is feasible and efficient.

## **(6) Details of Genetic Algorithms**

### Genetic algorithm: Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

### Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

### *Diagram Description automatically generated*

### Block diagram of a simple genetic algorithm.

### The working principle of a canonical GA:

### *Text Description automatically generated with medium confidence*

### individuals is generated by pseudo random generators whose individuals represent a feasible solution. This is a representation of solution vector in a solution space and is called initial solution. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space

### In the next step, individual members, chromosomes of the population represented by a string are evaluated to find the objective function value. This is exclusively problem specification. The objective function is mapped into a fitness function that computes a fitness value for each chromosome. This is followed by the application of GA operators.

### Reproduction or selection is usually the first operator applied on a population. It is an operator that makes more copies of better chromosomes in a new population. Thus, in reproduction operation, the process of natural selection causes those chromosomes that encode successful structures to produce copies more frequently. To sustain the generation of a new population, the reproduction of the chromosomes in the current population is necessary. For better chromosomes, these should be generated from the fittest chromosomes of the previous population.

### A crossover operator is used to recombine two chromosomes to get a better one the two chromosomes participating in the crossover operation are known as parent chromosomes and the resulting ones are known as children chromosomes. It is intuitive from this construction that good sub-strings from parent chromosomes can be combined to form a better child chromosome, if an appropriate site is chosen With a random site, the children chromosomes produced may or may not have a combination of good sub-strings from parent chromosomes, depending on whether or not the crossing site falls in the appropriate place. But this is not a matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover.

### Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes to be different from those of their parent. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level. When the bits are being copied from the current string to the new chromosomes, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability pm.

Selection type: tournament selection.

Crossover type: uniform crossover (0.1).

Mutation type: uniform mutation (0.01).

### Tournament Selection (GA): Tournament Selection is a Selection Strategy used for selecting the fittest candidates from the current generation in a Genetic Algorithm. These selected candidates are then passed on to the next generation. In a K-way tournament selection, we select k-individuals and run a tournament among them. Only the fittest candidate amongst those selected candidates is chosen and is passed on to the next generation. In this way many such tournaments take place and we have our final selection of candidates who move on to the next generation.

### If the tournament size is larger, weak candidates have a smaller chance of getting selected as it has to compete with a stronger candidate:

### 1.Select k individuals from the population and perform a tournament amongst them.

### 2.Select the best individual from the k individuals.

### Block diagram of the typical genetic algorithm

### *Diagram Description automatically generated*

### flow chart

### *Diagram Description automatically generatedDiagram Description automatically generated*

### *Diagram Description automatically generated*

### 

### Result

### 

### After pressing generate table we finally achieved the output we desired which in our project was a zero conflict timetable

### Graphical user interface, text, application Description automatically generated

### Table Description automatically generated