Order number :
Serial number :

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES
Computer Science Department

Thesis
Submitted in partial fulfilment of the requirements for the degree of

ACADEMIC MASTER

Field: **Mathematics and Computer Science**
Option: **Computer Science**
Specialty: **Distributed Systems and Artificial Intelligence**

Presented by:

- **Bennadji Ziad**
- **Naami Aissa**

Title

Timetabling Application for Algerian High Schools.

Defended on June 14Th 2022

Examination Committee:

M.	Benali Abdelkamel	MCA	Chair
M.	Othmani Samir	MAA	Examiner
M.	Messaoud Abbas	MCA	Supervisor

Academic year: 2021-2022

Contents

List of Figures

Dedication

Firstly, I want to Thanks Allah to make it possible, and for his gifts to us, patience, healthy, Comforts and all thing that helped us.

Secondly, I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, Ahmed and Nacuer Akila whose words of encouragement and push for tenacity ring in my ears. My sisters Rayane, Nariman, Maghfira and Shifa have never left my side and are very special. Also, my brother Akram, Abdallah have never let me down.

Without forgetting, my best friends Abdelfettah, Sami and Mr. Mohamed thank you to all the fine moment we got it together. Also all the friends that I gathered with them pretty memories.

I am gratitude to all my teachers that contributed at building me.

Thanks a lot to plant in me a love of computer sciences.

Also, dedicate my thesis to my promotion of 2021/2020.

It was thing, I will not forget.

To everyone who is love us

I dedicate this work.

Allah bless you.

Bennadji Ziad

Dedication

Thank Allah that inspired us to accomplish this work and without it we do not know what should to do.

I dedicate my thesis to my family and friends and to everybody that loving us without any exception

A special feeling of gratitude to my parent Messouad and Zakhrofa Naami whose are push me to keep study, and from this position give them all my thanks.

My greeting to my sisters, brothers and my friends.

I dedicate my work to my teachers.

Without forgetting my promotion 2021/2022.

I dedicate my dissertation to my fiancée , who is encourage me and understend me in my worst case.

Allah bless you

Thanks so much.

Naami Aissa

Thanks

First of all we thank our God who gave us
the strength and the will to carry out this work.
We extend our sincere thanks to our supervisor
Dr. Messaoud Abass
Who was responsible for providing advice and guidance
for each small and each large in this graduation note.
We extend our sincere thanks to all the teachers of the computer
science department
at Echahid Hamma Lakhdar University
El-Oued.
Also to our colleagues from the 2021-2022 promotion. We
also thanks all those who participated closely or
far to develop this work.

Abstract

The achievement of timetables at Algerian secondary schools is a hard task that makes principals suffering at the start of each school year. This thesis aims to propose a solution that will facilitate significantly this hard task. Thus, we have split the whole problem into two sub-problems and used different algorithms (Backtracking and Genetic) to reach this goal. Data sets from concrete secondary schools has been used for the experimentation of the achieved solution.

Keywords: High school timetabling, Meta-heuristic algorithms, Genetic algorithms, Backtracking algorithms, Constraints programming, Mixed integer programming.

Résumé

La réalisation des emploi du temps aux lycées algériens est une tâche délicate qui fait souffrir les directeurs à chaque rentrée scolaire. Cette thèse vise à proposer une solution qui permet de faciliter considérablement cette tâche. Ainsi, pour atteindre cet objectif, nous avons divisé l'ensemble du problème en deux sous-problèmes et utilisé différents algorithmes (Backtracking et Genetic). Afin d'expérimenter la solution proposée, nous avons utilisé des données concretes provenant des lycés Algeriens.

Mots clés: Emploi du temps lyciens, Algorithmes meta-Heuristic, Algorithmes Genetiques, Backtracking, Programmation par contraites, Programmations des enties mixés.

ملخص

إن إنشاء الجداول الزمنية في المدارس الثانوية الجزائرية هي مهمة صعبة تجعل مديري المدارس يعانون في بداية كل سنة دراسية. تهدف هذه الأطروحة إلى اقتراح حل من شأنه أن يسهل بشكل كبير هذه المهمة الصعبة. وهكذا ، اقترحنا تقسيم المشكلة برمتها إلى مشكلتين فرعيتين واستخدمت خوارزميات مختلفة (الخوارزمية التراجعية و الخوارزمية الجنية) لإنشاء الجدول الزمني. جمعت بيانات من مدارس ثانوية حققية تم استخدامها لتجربة النظام الذي تم تطويره.

الكلمات المفتاحية: الجدول الزمني للمدارس الثانوية ، خوارزميات ميتاهيوريتيك ، خوارزميات وراثية ، خوارزميات التراجع ، برمجة القيود ، برمجة عدد صحيح مختلط.

Introduction

At high schools, the timetable is a tedious task that the heads (principals) should achieve each beginning of the studying years. They suffer for about one week or more to prepare the new timetable. In this master thesis we propose a solution that will help high school principals to easily achieve their timetables. The proposed solution is a software that should consider most timetables constraints. Among the essential constraints we consider are:

- The fair distribution of the hourly volume of the same subject (modules) between teachers of the subject.
- Teachers should not teach more than two levels.
- No overlapping for teachers, classes and rooms.

In order to perform this application. We divide the global problem into two sub-problems:

1. Fair distribution of the hourly volume of subjects between teachers of the same subject.
2. The assignment of seances (events) into timetables slots.

We have achieved the first task using both backtracking algorithm and genetic algorithm. However, for the second task, we plan to achieve it through a more complicated genetic algorithm. In addition to a general introduction and conclusion, the reminder of this document contains 4 chapters, organised as follows:

- Chapter 1 presents the basics vocabularies about high schools and timetables, such as classes, levels, specialities, teachers, etc.

- Chapter 2 shows the state of the art about timetabling at high schools and essential research works open them.
- Chapter 3 describes our own design and proposed solution to solve the timetabling problems.
- Chapter 4 presents implementation techniques, tools, material that have used, experimentation and results.

Chapter 1

Algerian High Schools

1.1 Introduction

In this section, we present the description of Algerian high schools structures and organization. First, we describe the general structure of Algerian high schools: levels and specialities. Then, we present the organization of studying in Algerian high schools. In particular, we detail the studied subjects (modules) for each speciality, their coefficients and the number of hours assigned for each subject.

1.2 Algerian high schools structures

The high school general and technological education constitutes the academic path downstream in Algerian basic education. Its missions is summarized in the following points:

- Consolidate and deepen the knowledge acquired in the different disciplinary fields.
- Develop methods and capacities for personal work and teamwork and to cultivate the skills of analysis, synthesis, reasoning, judgment, communication and decision-making responsibilities.

- Offer diversified courses in order to enable students to get different choices and skills.
- Prepare students for further university studies.

Fig. 1.1 shows the general structure of Algerian high schools.

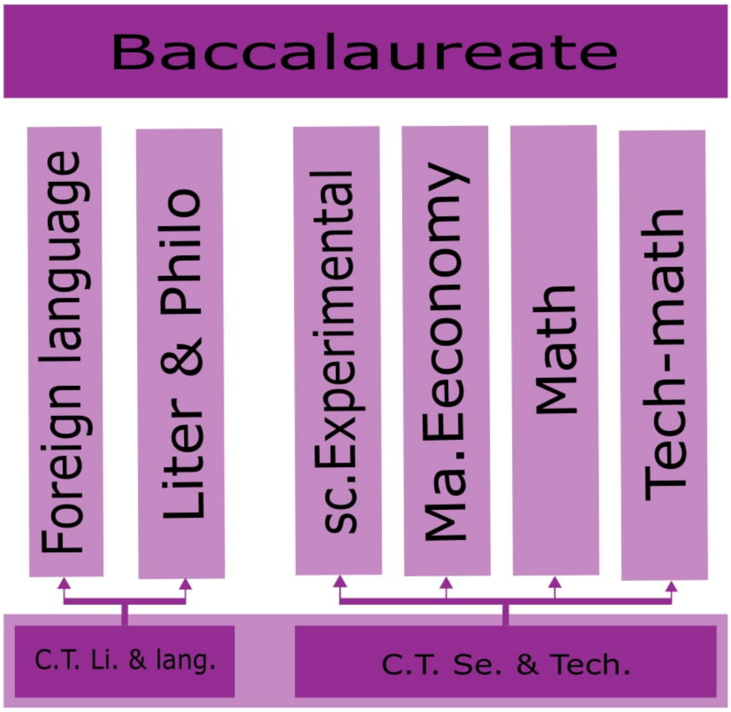

Figure 1.1: Structure of high schools in Algeria.

General and technological high education in Algeria lasting 3 years. Can be provided in accredited private educational and teaching establishments. It is always organized in specialities. It can be organized:

- Common trunks in the first year.
- Concrete specialties for the second and the third year.

The specialities are determined by the national education ministry.

Finally, The end of schooling in general high education and technology is evaluated by the baccalaureate of high education. Procedures for issuing the high education baccalaureate are set by the ministry for national education.

1.3 Organization of Algerian high schools

In Algerian high schools, the ministry of education responsible for tuning the organization of their establishments. Wherein every beginning of study year, the ministry of education send to high schools instructions ("plan of breeding") about studying classes and teachers.

The sending details the following points:

- The number of educational positions in the school.
- The number of classes at the school for each specialty.
- The program that should be followed by the school for each specialty.
- The coefficient and hour volume of each subject for each specialty.

It also mentions the conditions and constraints of education. Among these constraints, we may cite:

- Each teacher should teach 2 levels at most.
- Enable a free half day for teachers of specific subject: this half day for meeting with breeding inspector.
- Respect the average of activities between teachers of specific subject: Fairly distributions the teaching task between teachers.
- Enable teachers to get free day for break.

These constraints are hard constraints that must be respected. Additional hard constraints related to teachers and classes are detailed as follows:

- There is no teacher who teaches a different classes in two different rooms simultaneously.
- No studying at break hours.
- No class studied by two different teachers simultaneously at the same room.

- It is mandatory for classes to study all the program, without exception.

Note that, there are some particular conditions assigned to teachers and classes in some high schools. In order to be more concrete, we had visited the some real high schools (the high school of 8 May, ElOued) and met the responsible (the head of the high school, the principals, etc.). Some high schools consider additional soft constraints when creating timetables. Among these constraints are:

- Assigning specific teachers for final classes, because of their competencies.
- Making sports seances before leaving a school. (for example, form 10:00 O'clock to 12:00 O'clock).
- The essentials subjects (of each speciality) should be programmed in the morning.
- It is preferable that no more than two core subjects be studied in one day.

1.3.1 Techniques and procedures for creating timetables

To get more details about techniques and procedures for timetable creating, The high school principals showed us the main steps (see the next sub-sections) that should be followed in order to create a timetable.

1.3.2 Distribution of classes by teachers

The first step, consists to fairly distribute classes upon teachers. Firstly, the principal select a specific subject (e.g. Math). Then, he calculates the total hours for this subject and the average of hours that will be assigned to each teacher. So when he starts distribution, he has to consider it with level constraints.

1.3.3 Distribution the events into chronological table

The second step, distributes the events into the timetable. Firstly, the principal puts the seances that has a long time period for each speciality (e.g. sport), then he puts the core subjects in each speciality. After all this, he puts the rest seances with randomly manner. This procedure shrinks the magnitude of the facing problem and ensures constraints.

1.3.4 Assigning rooms to classes

After the distribution of events, he should assign a room for each class. However this is not a serious problem to solve, since the majority of high schools having a sufficient rooms. In such a way, they assign a particular room for each class.

Note that, for some particular events (such as sport, art education, technology, etc.) there are some limited rooms and laboratories that must be shared between classes.

Finally, it is worth saying that, like this arrangement to solve timetabling problems has better to adopted. Because, it decomposes the main complicated timetabling problem into 3 small easier sub-problems:

1. Distribution of classes by teachers.
2. Distribution the events into chronological table.
3. Assigning rooms to classes.

1.4 Timings and coefficients (studied program)

Table 1.1 summarises the studied program in Algerian high school system. This studied program has been legislated in 2011. It contains information about the studied subjects for each speciality, their hourly volume and their coefficients.

Note that, there is a newest program especially established for COVID-19 epidemic. This is a temporary program that will be ended with COVID-19 epidemic. This is why we did not detailed this program.

Table 1.1: Timing and coefficient table

	Common trunks				Liter & Philo				Foreign language				sc.Experimental				Math				Ma.Eeconomy				Tech-math			
	C.T. Li. & lang.		C.T. Se. & Tech.		Second year		Third year		Second year		Third year		Second year		Third year		Second year		Third year		Second year		Third year		Second year		Third year	
	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient	Hours	Coefficient
Math	3	2	6	5	2	2	2	2	2	2	2	2	5	5	5	5	7	7	7	7	3	3	4	5	6	6	6	6
Physic	(1+1)2	2	(2+2)4	4	(1+1)2	2							(2+2)4	5	(2+2)4	5	(2+3)5	6	(2+3)5	6					(2+3)5	5	(2+3)5	6
Exp.Science	(1+1)2	2	(2+2)4	4	(1+1)2	2							(2+3)5	6	(2+3)5	6	(1+1)2	2	(1+1)2	2								
Islamic	2	**2**	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Arabic	(1+5)6	5	4	3	4	5	7	6	5	4	5	5	3	2	3	3	3	2	3	3	3	2	3	3	3	2	3	3
Philosophy					4	5	7	6			3	2			3	2			3	2			2	2			2	2
Social studies	4	3	2	2	4	4	4	4	4	4	3	2	3	2	2	2	3	2	2	2	4	3	4	4	3	2	2	2
French	5	3	2	2	4	3	4	3	5	4	4	5	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
English	4	3	3	2	4	3	4	3	5	4	4	5	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
Foreign language									5	4	5	4																
Art	2	1	1	1	2	1			2	1			2	1			2	1			2	1						
Sport	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
Computer science	(1+1)2	2	(1+1)2	2																								
Technology			(1+1)2	2																					(2+4)6	6	(2+4)6	7
M Economy																					(1+4)5	5	(1+4)5	6				
Management																					5	3	4	3				
Law																					2	2	2	2				
sum	34	26	34	30	32	30	32	27	32	26	30	28	32	28	32	30	32	27	32	29	32	27	33	34	33	28	34	33

1.5 Conclusion

In this chapter, we conveyed the picture of Algerian high school, their features and their constraints. We also present the tips that we got from the principal for creating timetables.These tips will be adopted in our system.

Chapter 2

Timetabling: State of the art

2.1 Introduction

Schools timetabling is still a hard administrative task to launch each year's beginning. This is because of the large number of constraints and requirements that have to be satisfied. Educational timetabling problems have been classified among the hard problems. There are at least three types of timetabling: exam timetabling, course timetabling and high school timetabling. In this thesis we only focus on high schools timetabling. In the last few years, many techniques and methodologies have been developed to address high school timetabling problems.

Due to different cultural settings and educational systems, high school timetabling problems differ for one country to another. Thus, requirements and timetabling structure are different in countries.

Many research works such as [KS13] and [Pil14] have discussed approaches used in solving high school timetabling problem. This chapter aims to provide an overview upon technical and methods used for solving high school timetabling problem.

2.2 High school timetabling problem description

High school timetabling problems can be seen as the assignment of teachers, rooms, a number of lessons to be taught by teachers to specific classes or students and a set of constraints. This assignment works to minimise the constraint violations as much as possible. Constraints are a set of conditions that should be satisfied in a solution. There are two types of constraints: hard constraints and soft constraints. Hard constraints must be satisfied for a solution to be valid, whereas soft constraints can be violated but assigned a given penalty.

The following points show examples of constraints to be satisfied when creating timetables:

- Each event must be assigned to a time slot.
- No overlapping of resources (teachers, rooms, classes, ...).

2.3 Approaches for solving timetabling problem

Let's describe now the available approaches and algorithms applied to the high school timetabling problem. Existing algorithms can be classified into one or more of the following types:

- Mathematical optimization algorithms: Mathematical algorithms such as Integer Programming and Constraint Programming.
- Meta-heuristic algorithms: Meta-heuristic algorithms are used to find an acceptable solution in a reasonable amount of computational time. Among these algorithms are: Population-based algorithms and Single solution-based algorithms.
- Graph colouring algorithms: They use graph theory to represent the problem variables.
- Matheuristics approaches: They combine heuristic method with the mathematical optimisation algorithms.

- Hybrid approaches: combine several (two or more) meta-heuristic algorithms in a unified framework.

2.3.1 Integer Programming

Integer/mixed-integer programming (IP)/(MIP) is an effective way to represent the problem using mathematical approach, where it restricts at least one or more variables to be integer. It is frequently used to solve scheduling problem.

In Brazilian high school, Dornles [DM17] proposed an Integer programming based on the multi-commodity flow model. Each commodity represents a teacher and the arcs represent transition of time periods in an automate graph. They apply Dantzig–Wolfe decomposition principles to obtain an alternative formulation. Then a column generation approach is used to solve linear relaxation of the alternative formulation. The results show that the lower bounds obtained through their approach are tight and can be generated faster than previous approaches [PDdB14] reported in the literature.

Another related work [TIB19] proposed a MIP model to solve timetabling problems in Greece high schools. They implemented their model through Gurobi[1] and CPLEX[2] tools. They used two techniques to solve this problem. The first technique is called "Monolithic" model. It included all hard and soft constraints. The second split the main problem into six sub-problems. The experimentation results showed that the second method was effective in producing optimal solutions while the first method did not produce satisfactory results.

Constraint programming

Constraint programming (CP) is is a paradigm for solving combinatorial problems. In constraints programming the problem should be model as couple of constraints, this constraints most be satisfied in obtained solutions. Constraints programming is a famous technique in artificial intelligence and operation research.

[1]https://www.gurobi.com/

[2]https://www.ibm.com/analytics/cplex-optimizer

This paper [DS18] proposed a constraint programming method to the data set of International Timetabling Competition (ITC2011)[3] to solve a school timetabling problems. They formulated the CP model by using a scheduling-based point of view. They also included a solution-based phase saving technique to direct the first search in close proximity to the best solution found. The initial solution was generated by Kingston High School Timetabling Engine. They implemented event swap and event block swap techniques to improve the solution. The method was better than the standard CP approach.The experiments demonstrate that their approach outperforms the IP and maxSAT complete methods and provides competitive results when compared to dedicated heuristic solvers, like Variable Neighbourhood Search (VNS) [FS14].

2.3.2 Particle Swarm optimisation (PSO)

In computational science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

This paper [TB12] proposed a PSO algorithm to solve the problem of the timetabling in Greece. The algorithm starts with 150 active particles with different fitness value. In each generation, each particle was evolved with three procedures (swap with probability, random change with personal best, and random change with global best) to produce a new solution. If the fitness value of a particle exceeded a tolerance value, the particle was inactivated in the evolution procedure until the number of active particles was equal to 30. The purpose was to explore a wider search space and decrease the execution time. After 10,000 iterations, the best solution was further processed by a local search procedure to minimise the teachers idle time periods. This method outperformed constraint programming and other evolutionary algorithms.

[3] https://www.patatconference.org

2.3.3 Evolutionary algorithms

Evolutionary algorithms are population-based methods. They operate on a set of solutions and evolve new ones using selection, crossover and mutation procedures. The paper [DH12] proposed an evolutionary algorithm for ITC2011, and they were among the finalist. Two techniques are implemented in the solver: an indirect representation of timetable (each encoded solution consisted of a permutation of sub-events) and an evolutionary search controlled by an evolution strategy (a randomised method). In pre-processing, a permutation of eligible sub-events was invoked by taking several constraints into account. In the mutation process, two indices from the permutation were randomly chosen and then swapped. After that, times and required resources were assigned to the eligible sub-events during the decoding procedure.

2.3.4 Graph colouring

The paper [BG14] proposed a two-phase graph colouring approach to solve five randomly generated school timetabling problem. They used a bipartite multigraph to construct daily requirement matrices in the first phase, while a bipartite graph was used to assign lectures over unconstrained time slots in the second phase. the computational results demonstrated that it was effective.

2.3.5 Matheuristics

Matheuristics is a combination between two techniques metaheuristics and mathematical programming (MP). An essential feature is the central role played by the mathematical programming model, around which the overall heuristic is built.

This paper [Sav+20] proposed two pattern-based formulations and a cooperative parallel meta-heuristic in solving high school timetabling problems in Brazil. The two pattern-based formulations relied on teachers/classes meeting patterns. The first formulation used the original variables associated with layouts and blackbox solver to tackle instances of a practical size. The second formulation used a column generation framework. New agents were introduced to the parallel framework to exploit the developed column generation method in the second formulation. Their implementation was found to be competitive with the existing formulations [PDdB17].

2.3.6 Hyper-heuristics

Hyper-heuristic research comprises a set of approaches that share the common goal of automating the design and adaptation of heuristic methods to solve hard computational search problems [Bur+10].

Ahmed in his paper [A15] presented a three move acceptance method and a set of selection hyper-heuristics combining five different selection methods. to solve the ITC2011 instances. The three move acceptance methods were: Improve or Equal moves (IE), Simulated Annealing (SA) and Great Deluge (GD). The five different selection methods were: Random Descent (RD), Random Permutation (RP), Choice Function (CF), Simple Random (SR) and Random Permutation Descent (RPD). The selection hyper-heuristics were used with a set of nine low level heuristics (seven mutational and two hill climbing heuristics) during the search process. The experimental results revealed their selection method RP embedded with GD move acceptance method outperformed other selection hyper-heuristics. They concluded that the choice of selection hyper-heuristic and the adaptability of move acceptance components influenced its overall performance.

2.4 Conclusion

As conclusion we will first exploit the idea of splitting the problem into subproblems [TIB19]. Second, we will use the best evolution algorithm to solve the problem. Combining integer programming with genetic and backtracking algorithms will be our preferred choices for solving the Algerian high school timetabling.

Chapter 3

Design and proposal of an algorithm

3.1 Introduction

In this chapter, we are going to discuss the data structures (conceptual diagram, use case diagram, activity diagram) for modeling the timetabling of high school systems. Then, we study the possible computing models (algorithms). That is the proposed design model for computing timetabling (using UML [Fak] diagrams).

3.2 System architecture

In the above chapter, we have described the structure and organisation for high school. So here, we present the designed architecture for this system.

3.2.1 Use cases of the system

The Diagram 3.1 shows functionalities of the proposed system.

In the studied system, we have one actor "the principal". The principal is the person who is responsible for creating timetable (admin).

Fig. 3.1 shows the main use cases of the sole actor.

- Manage jobs: add, delete and edit the list of teachers and assign jobs (subjects) to them.

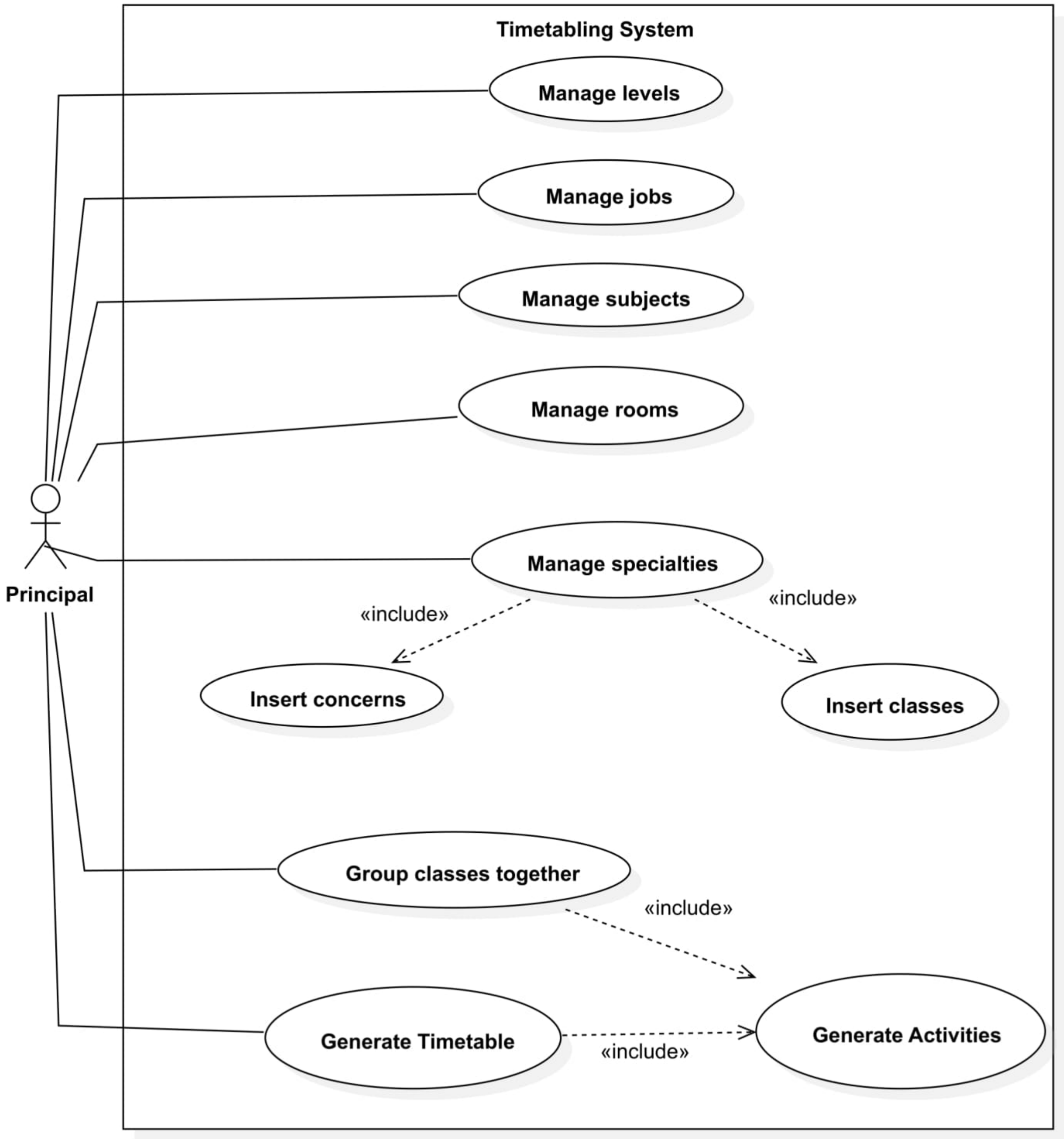

Figure 3.1: Use case diagram.

- Manage subjects: add, delete and edit subjects.
- Manage rooms: add, delete and edit rooms (stadium, hall, laboratory, etc).

- Manage Specialities: add, delete and edit specialities and specify the studied program and classes for each speciality.
- Group class together: this feature enable to link tow or more classes together in order to study a specific subject within the same seance (timetable slot).
- Generate Timetable: this is the main behavior of our system. It enables us to generate timetables.
- Generate events (activities): this use is used by tow other use cases (group class together, Generate timetable).

3.2.2 Class diagram

The class diagram that models the system structures is presented in Fig. 3.2.

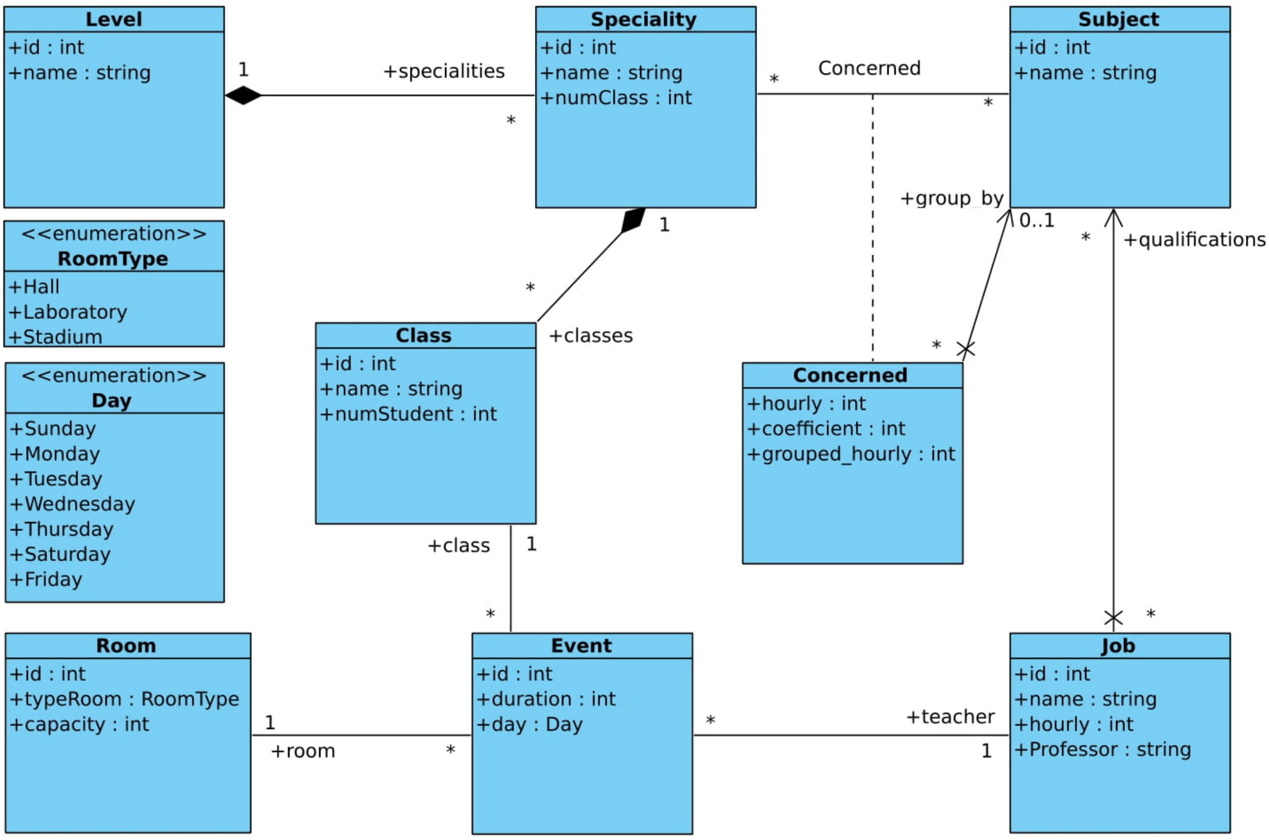

Figure 3.2: Data structure.

In our diagram, we have the following classifier.

- Level: this class represent the studied levels in a high school. The attribute name represents the name of the level.
- Speciality: in addition to the id, this class has following attributes:
  - name: name of speciality.
  - num_class: number of classes in the speciality.
- Subject: it represents the studied modules. The attribute name is name of module.
- Class: it represents a group of study in a specific speciality. It has the following attributes (in addition to id):
  - name: name of class.
  - numb_student: number the students in the class.
- Concerned : it represents the studied program for each speciality. It has the following attributes:
  - hourly: the total studied hours in one week.
  - coefficient : coefficient of the module.
  - grouped_hourly: The alternative module with which the current module will be grouped.
  -
- Job: it represents teaching positions. It has the following attributes:
  - name: name of the position.
  - hourly: max volume hours for each position.
  - professor_name: name of the person occupying the position.
- Room: it has the following attributes:
  - name: name of room.

  - typeroom: the room type can be: stadium, hall, laboratory or workshop.
  - capacity: the number of available places in a room.

- Event: it represents the studied seances. it has the following:
  - duration : the required time for the seance (event).
  - day: Days of the week.

In this structure, we have used the following enumerations:

- RoomType: it may be one of the following values: stadium, hall, laboratory or workshop.
- Day: days of the week.

3.2.3 Activity diagram

Fig. 3.3 shows the activity diagram of the timetabling system. It describes the actions that we should follow to create a timetable.

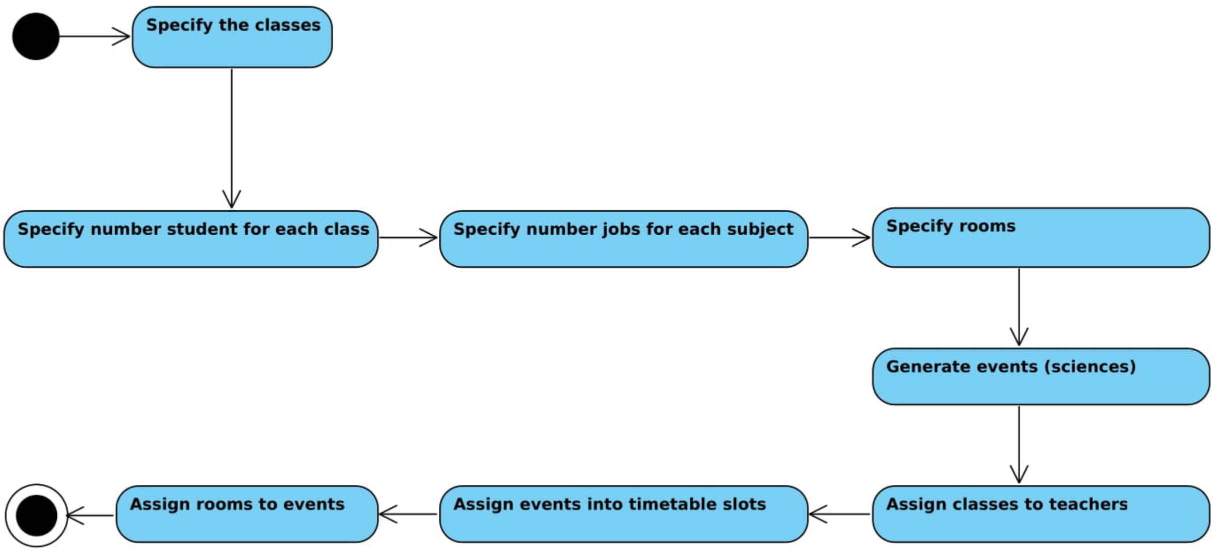

Figure 3.3: Activity diagram.

Note that, the standard data in high school such as levels, specialities, subjects, studied programs, etc, are already stated in the system. The user will only update them according to his own school.

3.3 Computing model(Algorithm model)

Under this title, we present the used computing model for solving timetable problems. As mentioned earlier, we will decompose the main problem into two sub problems. So, here we show the decomposition for timetable problem and present their respective computing models (algorithms).

3.3.1 Assign classes to teachers

In this problem, we have a set of teachers $\mathbf{T} = t_1, t_2, \ldots, t_n$ who teach specific subject (e.g. Math), as well as a set of classes $\mathbf{C} = c_1, c_2, \ldots, c_m$ that study the subject. Note that, each class has its own hourly volume (weekly studied hours for the subject). The objective is to get the best assignment of classes to teachers, considering the following two constraints:

1. No teacher has more then two level.
2. Fair division of the classes hourly volumes between teachers.

The search space had n^m cases, where `n` is the number of teachers and `m` is the number of classes.

So, in order to resolve this problem, we have tried the following three algorithms:

Solution1: Simple linear algorithm

The following pseudo-code (Algorithm 1) presents our first attempt to solve this problem.

This algorithm uses the following three parameters:

- List of teachers (jobs).
- List of classes (seances).
- The average of hours that will be assigned to each teacher (see formula 3.1).

Algorithm 1: Assign classes to teachers.

Input: $jobs : [], activities : []$.

Output: dictionary contains the teacher and their assigned seance

```
Function Main(jobs, activities):
    Order The list activities by level and speciality.
    result ← initial dictionary type of (key : job, value : list < activity >).
    for j ∈ jobs do
        levelChanged ← false
        activity ← get the first unassigned activity.
        add the (key : j, value : [activity]) to result dictionary.
        oldLevel ← activity.level.
        sumHours ← activity.hours
        for a ∈ (unassigned activities) do
            if sumHours < avg − 1 then
                result[j].add(a)
                sumHours ← sumHours + a.hours
                levelChanged ← not(oldLevel ≠ a.level)
            else
                if sumHours > avg + 1 AND levelChanged = true then
                    lastActivity ← get last assign activity to j
                    result[j].remove(lastActivity)
                    sumHours ← sumHours − lastActivity.hours
                    diff ← avg − sumHours
                    Seek the closest seance for this diff.
                else if levelChanged = false then
                    lastActivity ← get last assign activity to j
                    result[j].remove(lastActivity)
                    sumHours ← sumHours − lastActivity.hours
                    diff ← avg − sumHours
                    Seek the closest seance for this diff and with a different level.
                break
    return result
```

$$avg = (\sum activities.hour)/count(jobs) \tag{3.1}$$

In each iteration, the algorithm handles one teacher and try to assign some unassigned classes (seances) to him. If the algorithm does not meet the condition in line 11, it should check the last assigned class that breaks the condition. Then, it replaces the last assigned class with another one that satisfies the constraints.

Obviously, this algorithm just provides a candidate solution. It only gets back one step. To sum up, this algorithm does not give optimal results. Thus, we deal with another solution using a full back tracking algorithm.

Solution2: Backtracking algorithm

Backtracking is a general algorithm for finding solutions to some computational problems, notably constraint satisfaction problems, that incrementally builds candidates to the solutions, and abandons a candidate ("backtracks") as soon as it determines that the candidate cannot possibly be completed to a valid solution[Doi19].

Fig. 3.9 shows the class diagram of the Backtracking algorithm.

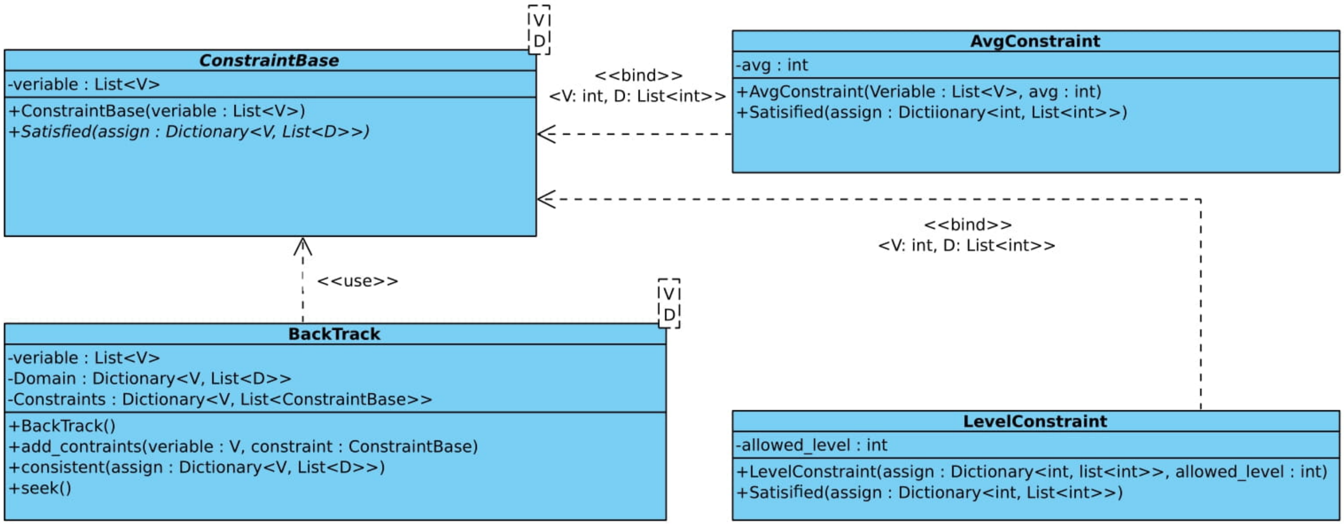

Figure 3.4: Class diagram of backtrack algorithm implementation.

In order to implement the backtracking algorithm, we model the problem as constraints satisfaction problem.
The constraints satisfaction are composed of variables with value that fall into ranges (domains). For solve the problem it most satisfied the constraints between a variables.

The pseudo-code presented in Algorithm 2 shows backtracking algorithm implementation model. The feedback of backtracking algorithms ensures that it give mostly good results. However, in some particular cases, it fails. Thus, we will consider backtracking algorithm outputs and use them as inputs for a genetic algorithm that will improve significantly the final results.

Solution3: Genetic algorithm

Genetic algorithm (GA) is a meta heuristic inspired by the theory evolution. GA is commonly used to get high-quality solution for optimization problems and search problems. Genetic algorithms are based on an analogy with natural genetic structure and behaviour of chromosomes of the population. The following points summarises main analogy between GAs and natural genetic process:

1. Individual in population compete for resources and mate.

Algorithm 2: BackTrack

```
Function BackTrack(Assign):
    unassigned_teacher ← Get all unassigned jobs (teacher).
    The base case. stop when all the variables assigned.
    if unassigned_teacher.count = 1 then
        add to Assign all unassigned seance
    Pick one out.
    teacher ← unassigned.First()
    for S ∈ (unassigned seance) do
        Make copy of assign.
        copy_assign ← Copy(Assign)
        Add(unassigned_teacher, S) to copy_assign
        if consistent(copy_assign) then
            result ← backTracking(copy_assign);
            if result ≠ null then
```

2. Those individuals who are successful (fittest) then mate to create more offspring than others.
3. Genes from "fittest" parent propagate throughout the generation, that is sometimes create a new better offspring than their parents.
4. Each successive generation is more suited for their environment.

Fig. 3.5 shows the general activity diagram of genetic algorithms.

So, for implementation genetic algorithm we should define the following:

Variables encoding system

In order to solve the timetabling problem using GA, we should firstly choose an adequate data structures for chromosomes (candidate solution).

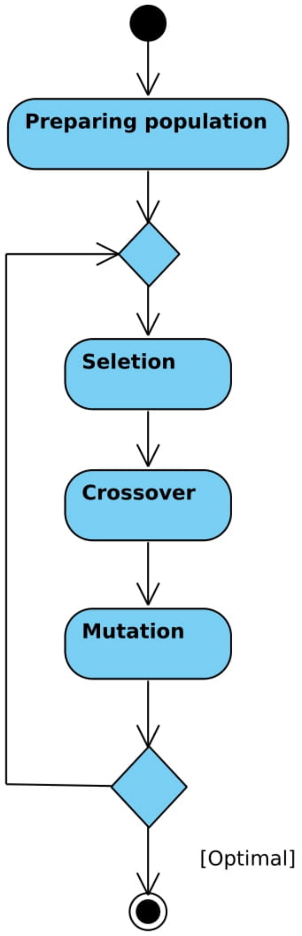

Figure 3.5: Activity diagram of genetic algorithm.

Fig. 3.6 shows the proposed encoding matrix, rows represent teachers and columns represent classes seances. Each intersection cell is an object including the following information:

- Level identifier (id_level).
- Speciality identifier (id_speciality).
- Class identifier (id_class).
- Subject identifier (id_subject).
- Hourly volume (count).

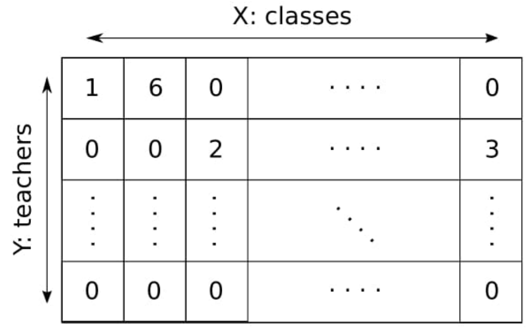

Figure 3.6: Encoding matrix.

To simplify the presentation we show only hourly volume in cells of the table not whole object. Note that, `0` value means that the corresponding teacher does not teach the intersected class.

Selection function

Selection is genetic operator used for select candidate solution for mating pool. It is analogous to biological natural selection. The survival to the best. So for evaluate the fitness of candidate solution (chromosome) we present the following fitness function 3.2.

$$fitness = \sum_{n=t_1}^{n=t_n}(|\sum_{c_1}^{c_m}(Hourly\_volume) - avg| + f(n)) \tag{3.2}$$

$$f(n) = \begin{cases} 10 & if\ n\ (teacher)\ has\ 1\ level. \\ 0 & if\ n\ (teacher)\ has\ 2\ levels. \\ 30 & \text{if n (teacher) has 3 levels.} \end{cases}$$

Crossover function

The crossover is a genetic operator used to vary solution contents (chromosomes) from one generation to the next. Crossover is sexual reproduction. Two chromosomes are picked from the mating pool (The selected solution for mating) at random to crossover in order to produce superior new solution (offspring).

Fig. 3.7 shows the proposed crossover to solve the timetables problems.

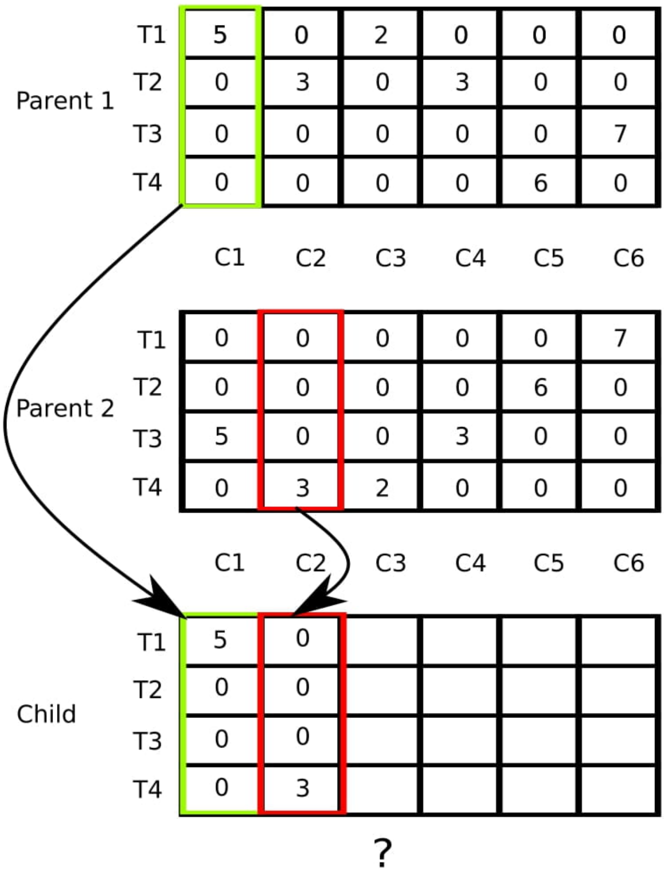

Figure 3.7: Crossover work flow

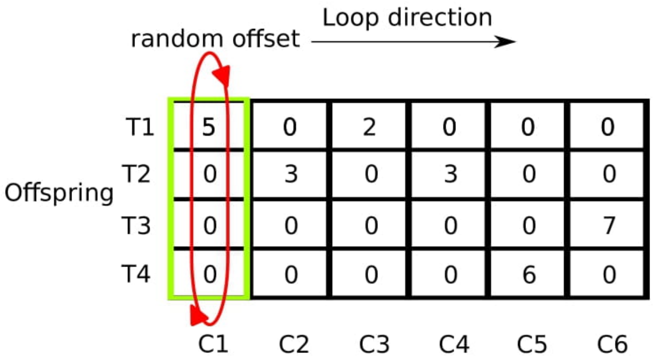

Figure 3.8: Mutation work flow

Mutation function

The mutation function is genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation.
Fig. 3.8 shows the mutation we proposed to solve the timetabling problems and Fig. 3.9 shows the class diagram of genetic algorithm implementation.

3.3.2 Assigning events into timetable slots

In order to assigning events into chronological tables, we have first the set of events $\mathbf{E} = e_1, e_2, \ldots, e_n$ and 3-way tensor (three-dimensional array: $Days \times periods \times classes$). The objective is to place the events into the global timetable: a 3-way tensor. When assigning events, we should consider the following two types of constraints:

Hard constraints

- No overlap between the teachers.
- No overlap between the classes.
- No overlap between the rooms.
- It is mandatory to assign all events into the table.

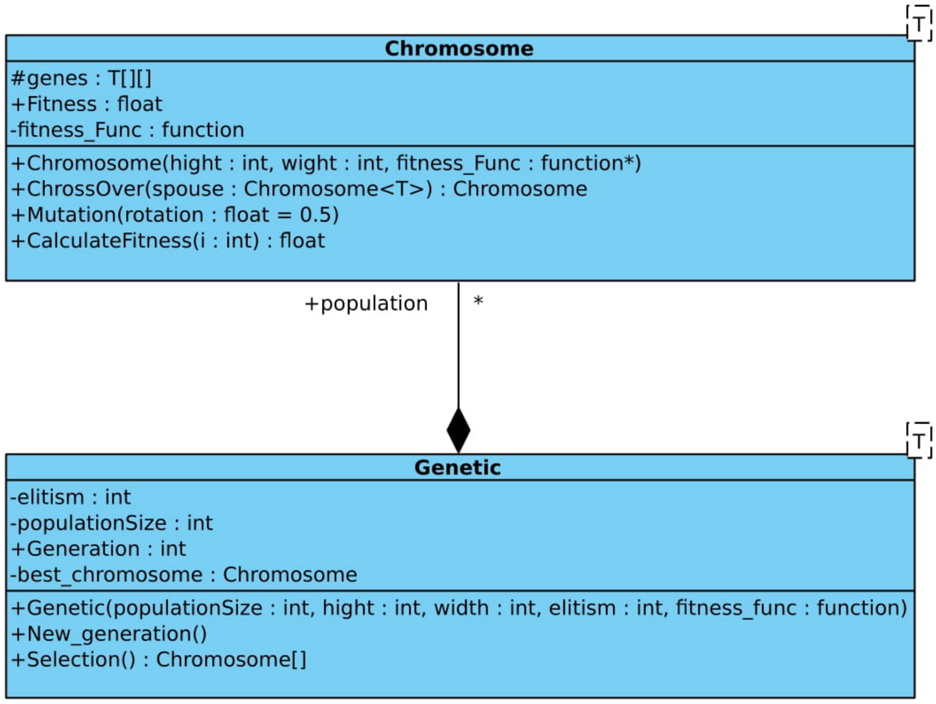

Figure 3.9: Class diagram of genetic algorithm implementation.

Soft constraints

- Enable a free half day for teachers of specific subject: this half day is dedicated for meeting with breeding inspectors.
- It is preferable that no more than two core subjects be studied in one day.
- The essentials subjects (of each speciality) should be programmed in the morning.
- Making sports seances before leaving school. (for example, form 10:00 O'clock to 12:00 O'clock).

In this problem, we implement a genetic algorithm on a mixed-integer programming model.

A mixed-integer programming (MIP) is an effective way to represent the problem using mathematical approach, where it restricts at least one or more variables to be integer. It is frequently used to solve scheduling problem.

In order to solve the problem, we decompose it into stages. In first stage, we force the algorithm to respect the hard constraints. Then, in the second stage, we run the algorithms to satisfy the remaining constraints (soft constraints).

For the implementation of our genetic algorithm, we update the original algorithm as follows: both in the first and second stages, we use only one encoding system. We have a 3-way tensor (three-dimensional array: $Days \times periods \times classes$). The first vector represents days, the second vector represents periods and the last vector represents classes. In order to create mixed-integer programming model we will encoding all variables (days, periods, classes) using integers.

3.4 Conclusion

In this chapter, we have presented our vision to solve the Algerian timetabling problem. First, we split the whole problem into two sub-problems. The first consists to distribute classes upon teaches. The second will assign each event to a time slot (time period). We proposed using Backtracking and Genetic algorithms to perform the proposed solution.

Chapter 4

Implementation

4.1 Introduction

In this chapter, we first describe the used tools. Then, we present comparison of the obtained results through the application of the studied algorithms.

4.2 Development environment

Here, we present the used programming language, database management system, and the integrated development environment.

4.2.1 The programming language

We had written our code using C# programming language. C# is a general-purpose, type-safe, object-oriented programming language. The goal of the language is programmer productivity. To this end, C# balances simplicity, expressiveness, and performance. The chief architect of the language since its first version is Anders Hejlsberg (creator of Turbo Pascal and architect of Delphi). The C# language is a neutral platform works with a range of platform-specific runtimes[Alb21].

We have chosen C# for the following reasons:

- We have a considered expertise on it.
- C# is adopted with windows. That is serves our propose to make the system works in desktop.
- C# has powerful and useful third-parts libraries.

4.2.2 The integrated development environment(IDE)

We had chosen Microsoft Visual studio. The latter is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, etc... [Ter]. The following reasons direct us to use it:

- Visual studio is adopted by Microsoft.
- C# is integrated on Visual studio.
- Visual studio has a powerful tools to design user-interface.
- etc.

4.2.3 The database management system (DBMS)

We use an XML databases to store the data of the studied system . XML database is semi-structured database to store non-critical data. This data can be queried, transformed, exported and returned to a calling system. XML databases are flavor of document-oriented databases.

XML helps us at the following points:

- XML is Light weight and does not need a lot of tools for working.
- XML is readable by man and machine.
- XML is transformable.
- etc.

The used materials (Hardware)

We implement our system in a laptop with the following specifications:

- CPU: Intel® Core™ i5-3230M CPU @ 2.60GHz × 4.
- RAM: 8 GiB.
- GPU: AMD® Hainan / Intel® HD Graphics 4000 (IVB GT2).
- OS: Windows 8.1 64-bit professional.

4.3 Data set

We collect our data set from the following high-schools:

1. Ibn Rachiq Al-Qayrawani Al-Taibet High School.
2. Mujahid Abdul Razzaq Eida 8 may High School.

We get three data sets from these highs schools. We get the first data set from Ibn Rachiq Al-Qayrawani high school. It is so big school because it is the only one in its region at Taibet (W. of Tougourt). From the second high school (Mujahid Abdul Razzaq Eida) we get two data sets. The first one is before COVID-19 pandemic and the second is after. This high school is medium high school. It is located at 8 May in El-Oued.

Table 4.1 presents more details about these data sets.

4.4 Realization

Due to the size of problem and size of task on the studied system, we couldn't accomplish all the programming works within this project. However, we have accomplished the first step (assigning classes to teachers).

Table 4.1: Content of data sets.

High Schools	Number of teachers	Number of Classes	Number of halls	number of computer science Lab.	Number of science Lab	Number of Subjects
Ibn Rachiq Al-Qayrawani Al-Taibet High School	64	42	50	1	2	18
Mujahid Abdul Razzaq Eida 8 may High School (before COVID-19)	33	18	40	1	2	18
Mujahid Abdul Razzaq Eida 8 may High School (after COVID-19)	33	18	40	1	2	17

4.5 Experimentation and results

Under this title, we test and show the first phase (assigning classes to teachers) of our proposed system. Note that, due to the size of data sets. We will not discuss all instances of the system.

4.5.1 Eight may high school data set

In this data set, we only focus on mathematics and natural science subjects:

Subject1: Mathematics

For mathematics subject, we have the following related information:

Teachers

In This data set, we have four mathematics teachers. See the table 4.2.

Table 4.2: Mathematics teachers list.

ID	Name of Teacher	Subject
1	Math1	Mathematics
2	Math2	Mathematics
3	Math3	Mathematics
4	Math4	Mathematics

Classes

Table 4.3shows the classes studying mathematics and their hourly volumes.

Table 4.3: List of classes concerned with math.

ID	Classes	Hourly volume in week	Concern
1	1 Lit1	2	Mathematics
2	1 SET1	4	Mathematics
3	1 SET2	4	Mathematics
4	1 SET3	4	Mathematics
5	1 SET4	4	Mathematics
6	1 SET5	4	Mathematics
7	2 Lit1	2	Mathematics
8	2 Lang1	2	Mathematics
9	2 SE1	3	Mathematics
10	2 SE2	3	Mathematics
11	2 SE3	3	Mathematics
12	2 GÉ1	3	Mathematics
13	3 SE1	4	Mathematics
14	3 SE2	4	Mathematics
15	3 SE3	4	Mathematics
16	3 GÉ1	3	Mathematics
17	3 Lang1	2	Mathematics
18	3 Lit1	2	Mathematics

Results

Table 4.4 shows the result we get using Backtracking algorithm only.

Table 4.4: Result of assigning classes to teachers using Backtracking (Mathematics).

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
Math1	3Lang1 (2), 1SET1 (4), 3SE1 (4), 3SE3 (4)	14	14	True	True
Math2	1Lit1 (2), 2Lit1 (2), 1SET2 (4), 2SE1 (3), 2SE2 (3)	14	14	True	True
Math3	1SET3 (4), 2GÉ1 (3), 1SET4 (4), 1SET5 (4)	15	14	True	True
Math4	2Lang1 (2), 3Lit1 (2), 2SE3 (3), 3GÉ1 (3), 3SE2 (4)	15	14	True	True

Subject2: Natural sciences

For natural sciences, we have the following related information:

Teachers

In this data set, we have three natural Sciences teachers. See table 4.5.

Table 4.5: Natural sciences teachers list.

ID	Name of Teacher	Subject
1	NS1	Natural sciences
2	NS2	Natural sciences
3	NS3	Natural sciences

Classes

Table 4.5 shows the classes studying natural sciences and their hourly volumes.

Results

Table 4.7 shows result we get using Backtracking algorithm only.

Table 4.6: List of classes concerned with natural sciences.

ID	Classes	Hourly volume in week	Concern
1	1 Lit1	1	Natural sciences
2	1 SET1	3	Natural sciences
3	1 SET2	3	Natural sciences
4	1 SET3	3	Natural sciences
5	1 SET4	3	Natural sciences
6	1 SET5	3	Natural sciences
7	2 Lit1	1	Natural sciences
8	2 SE1	4	Natural sciences
9	2 SE2	4	Natural sciences
10	2 SE3	4	Natural sciences
11	3 SE1	4	Natural sciences
12	3 SE2	4	Natural sciences
13	3 SE3	4	Natural sciences

Note that,in these instances the system just implements backtracking algorithm only. because its result is sufficient.

4.5.2 Al-Taibet high school data set

In this data set, we only focus on physics and Arabic subjects:

Subject1: Physics

For physics, we have the following related information:

Teachers

In this data set, we have eight physics teachers. See table 4.8

Classes

Table 4.9 shows the classes studying physics with their hourly volumes.

Results

Table 4.10 shows result we get using Backtracking algorithm only, and table 4.11 shows the obtained result through the combination of both backtracking and

Table 4.7: Result of assigning classes to teachers using Backtracking (Natural Sciences).

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
NS1	1Lit1 (1), 2Lit1 (1), 2SE1 (4), 2SE2 (4), 2SE3 (4)	14	14	True	True
NS2	1SET1 (3), 1SET2 (3), 3SE1 (4), 3SE2 (4)	14	14	True	True
NS3	1SET3 (3), 1SET4 (3), 1SET5 (3), 3SE3 (4)	13	14	True	True

Table 4.8: Physics teachers list.

ID	Name of Teacher	Subject
1	PHY1	Physics
2	PHY2	Physics
3	PHY3	Physics
4	PHY4	Physics
5	PHY5	Physics
6	PHY6	Physics
7	PHY7	Physics
8	PHY8	Physics

genetic algorithm.

Subject2: Arabic

For Arabic, we have the following related information:

Teachers

In this data set, we have seven Arabic teachers. See table 4.12

Classes

Table 4.13 shows the classes studying Arabic with their hourly volumes.

Results

Table 4.9: List of classes concerned with physics.

ID	Classes	Hourly volume in week	Concern
1	1 Lit1	3	Physics
2	1 Lit2	3	Physics
3	1 Lit3	3	Physics
4	1 Lit4	3	Physics
5	1 SET1	6	Physics
6	1 SET2	6	Physics
7	1 SET3	6	Physics
8	1 SET4	6	Physics
9	1 SET5	6	Physics
10	1 SET6	6	Physics
11	1 SET7	6	Physics
12	1 SET8	6	Physics
13	1 SET9	6	Physics
14	1 SET10	6	Physics
15	1 SET11	6	Physics
16	2 SE1	6	Physics
17	2 SE2	6	Physics
18	2 SE3	6	Physics
19	2 SE4	6	Physics
20	2 SE5	6	Physics
21	2 Lit1	3	Physics
22	2 Lit2	3	Physics
23	2 Mat1	7	Physics
24	2 Tec1	7	Physics
25	3 SE1	6	Physics
26	3 SE2	6	Physics
27	3 SE3	6	Physics
28	3 SE4	6	Physics
29	3 SE5	6	Physics
30	3 Tec1	7	Physics
31	3 Mat1	7	Physics

Table 4.10: Result of assigning classes to teachers (Physics).

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
Phy1	1Lit1 (3), 1SET1 (6), 1SET3(6), 1SET4(6)	21	22	True	True
Phy2	1Lit2 (3), 1SET2 (6), 1SET5 (6), 1SET6 (6)	21	22	True	True
Phy3	1Lit3(3), 1SET7 (6), 1SET8 (6), 1SET9 (6)	21	22	True	True
Phy4	1Lit4 (3), 1SET10 (6), 1SET11 (6), 2SE1 (6)	21	22	True	True
Phy5	2Lit1 (3), 2SE2 (6), 2SE3 (6), 2SE4(6)	21	22	True	True
Phy6	2Lit1 (3), 2SE5 (6), 3SE1 (6), 3SE2 (6)	21	22	True	True
Phy7	2Mat1 (7), 2Tec1 (7), 3Tec1 (7)	21	22	True	True
Phy8	3SE3 (6), 3SE4 (6), 3SE5(6), 3Mat1 (7)	25	22	True	False

Table 4.14 shows result we get using Backtracking algorithm only, and table 4.15 shows the obtained result through the combination of both backtracking and genetic algorithm.

Note that, you can see the whole results got by our system here : `https://drive.google.com/drive/folders/1T1kSc8dz7z-bpvHjq95Rn1lB-Kh-0EQZ?usp=sharing`

4.6 Conclusion

As a conclusion, this chapter described the tools and materials we had used to create and implement our system. Obviously, the results got by backtracking algorithm are not optimal but they are nearly to be optimal. However, The Genetic Algorithm (which uses the output of backtracking as inputs) provides better results

Table 4.11: Result of Assigning classes to teacher (Physics) by combined backtracking with genetic algorithm.

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
Phy1	1SET3(6), 1SET4(6), 1Lit1(3), 3SE1(6)	21	22	True	True
Phy2	1SET5(6), 1SET6(6), 3SE2(6), 3SE3(6)	24	22	True	False
Phy3	1SET7(6), 1Lit2(3), 2SE1(6), 2SE2(6)	21	22	True	True
Phy4	1SET1(6), 2SE4(6), 2Tec1(7)	19	22	True	False
Phy5	1Lit3(3), 1SET8(6), 2SE3(6), 2Mat1(7)	22	22	True	True
Phy6	1SET9(6), 1SET10(6), 2Lit1(3), 2SE5(6)	21	22	True	True
Phy7	2Lit2(3), 3SE4(6), 3Tec1(7), 3SE5(6)	22	22	True	True
Phy8	1Lit4(3), 1SET11(6), 1SET2(6), 3Mat1(7)	22	22	True	True

Table 4.12: Arabic teachers list.

ID	Name of Teacher	Subject
1	ARA1	Arabic
2	ARA2	Arabic
3	ARA3	Arabic
4	ARA4	Arabic
5	ARA5	Arabic
6	ARA6	Arabic
7	ARA7	Arabic

(almost optimal) compared with the backtracking algorithm.

Table 4.13: List of classes concerned with arabic.

ID	Classes	Hourly volume in week	Concern
1	1 Lit1	7	Arabic
2	1 Lit2	7	Arabic
3	1 Lit3	7	Arabic
4	1 Lit4	7	Arabic
5	1 SET1	4	Arabic
6	1 SET2	4	Arabic
7	1 SET3	4	Arabic
8	1 SET4	4	Arabic
9	1 SET5	4	Arabic
10	1 SET6	4	Arabic
11	1 SET7	4	Arabic
12	1 SET8	4	Arabic
13	1 SET9	4	Arabic
14	1 SET10	4	Arabic
15	1 SET11	4	Arabic
16	2 SE1	3	Arabic
17	2 SE2	3	Arabic
18	2 SE3	3	Arabic
19	2 SE4	3	Arabic
20	2 SE5	3	Arabic
21	2 Lit1	4	Arabic
22	2 Lit2	4	Arabic
23	2 Mat1	3	Arabic
24	2 Tec1	3	Arabic
25	2 GÉ1	3	Arabic
26	2 Lang1	5	Arabic
27	3 SE1	3	Arabic
28	3 SE2	3	Arabic
29	3 SE3	3	Arabic
30	3 SE4	3	Arabic
31	3 SE5	3	Arabic
32	3 Tec1	3	Arabic
33	3 Mat1	3	Arabic
34	3 GÉ1	3	Arabic
35	3 Lang1	5	Arabic
36	3 Lit1	7	Arabic
37	3 Lit2	7	Arabic

Table 4.14: Result of assigning classes to teachers (Arabic).

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
ARA1	2Mat1 (3), 2SE1 (3), 2SE2 (3), 2SE3 (3), 2SE4(3), 2SE5 (3), 1SET1 (4)	22	22	True	True
ARA2	2Tec1 (3), 2GÉ1 (3), 3Tec1 (3), 3SE1 (3), 3SE2 (3), 3SE3 (3), 1SET2(4)	22	22	True	True
ARA3	3SE4 (3), 3SE5 (3), 3Mat1 (3), 3GÉ1 (3), 2Lang1 (5), 3Lang1 (5)	22	22	True	True
ARA4	1SET3 (4), 1SET4 (4), 1Lit1 (7), 1Lit2 (7)	22	22	True	True
ARA5	1SET5 (4), 1SET6 (4), 1Lit3 (7), 1Lit4 (7)	22	22	True	True
ARA6	1SET7 (4), 1SET8 (4), 3Lit1 (7), 3Lit2 (7)	22	22	True	True
ARA7	1SET9 (4), SET10 (4), SET11 (4), 2Lit1 (4), 2Lit1 (4)	20	22	True	False

Table 4.15: Result of Assigning classes to teacher (Arabic) by combined backtracking with genetic algorithm.

Name of teacher	Classes	Total hour	Average	Is level respect	Is average respect
ARA1	2SE1 (3), 2Tec (3), 1SET1 (4), 1Lit1 (7), 1SET2 (4)	21	22	True	True
ARA2	2Mat1 (3), 2SE2 (3), 3SE1 (3), 3SE2 (3), 3Mat1 (3), 3Lit1 (7)	22	22	True	True
ARA3	2SE3 (3), 3SE3 (3), 3GÉ1 (3), 2Lang1 (5), 3Lit2 (7)	21	22	True	True
ARA4	2SE4 (3), 1Lit2 (7), 1SET3 (4), 1SET4 (4), 2Lit1 (4)	22	22	True	True
ARA5	3SE4 (3), 1SET5 (4), 1Lit3 (7), 1Lit4 (7)	21	22	True	True
ARA6	1SET6 (4), 3Tec (3), 1SET7 (4), 3SE5 (3), 3Lang1 (5), 1SET8 (4)	23	22	True	True
ARA7	2SE5 (3), 2GÉ1 (3), 1SET9 (4), 1SET10 (4), 1SET11 (4), 2Lit2 (4)	22	22	True	True

Conclusion

In this thesis, we have studied the timetabling problem. More specifically, we focused on Algerian high school timetabling. Although this problem has been handled in other countries, the Algerian high school has its own cultures and laws. Thus, we have first studied the specifications of Algerian high school (in the first chapter). Then (in the second chapter) we studied how other people (in other countries) dealt with the timetabling problem in their particular high schools. After that (in the third chapter), we proposed our own solution for Algerian high school timetabling. We have mainly split the problem into sub-problems and used backtracking and genetic algorithms. Finally, (In chapter 4) we proposed an implementation using VS environment and C# Language. The optioned results from the first stage of the proposed solution (sharing the hourly volume of a subject upon teachers of the same subject), were largely sufficient. Several experimentation using real high schools has been achieved. They proof that constraints about the hourly average and levels are mostly satisfied.

Perspectives

As perspective, we plan to continue the implementation of the second stage of the proposed solution. It will be a genetic algorithm using three-dimensional table, where column represent days of the week, rows represent periods and layers represent classes ($Days \times periods \times classes$). The most challenge here is to elaborate a best fitness function that will help to rapidly and correctly improve the model. Note that hard constraints (No overlapping of teacher, classes and rooms) must be completely satisfied.

References

[Alb21] Joseph Albahari. *C 9.0 in a Nutshell : the definitive reference*. Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo O'reilly March, 2021. ISBN: 9781098100964.

[ASP19] Carlos E. Andrade, Thuener Silva, and Luciana S. Pessoa. "Minimizing flowtime in a flowshop scheduling problem with a biased random-key genetic algorithm". In: *Expert Systems with Applications* 128 (2019), pp. 67–80. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.03.007. URL: https://www.sciencedirect.com/science/article/pii/S0957417419301605.

[A15] Leena N. Ahmed, Ender Özcan, and Ahmed Kheiri. "Solving high school timetabling problems worldwide using selection hyper-heuristics". In: *Expert Systems with Applications* 42.13 (2015), pp. 5463–5471. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.02.059. URL: https://www.sciencedirect.com/science/article/pii/S0957417415001670.

[BG14] Rakesh P. Badoni and D.K. Gupta. "A graph edge colouring approach for school timetabling problems". In: *International Journal of Mathematics in Operational Research* 6.1 (2014), p. 123. DOI: 10.1504/ijmor.2014.057853.

[Bur+10] E Burke et al. "A classification of hyper-heuristic approaches". In: Jan. 2010, pp. 449–468. DOI: 10.1007/978-3-319-91086-4_14.

[DH12] Jonathan Domrös and Jörg Homberger. "An Evolutionary Algorithm for High School Timetabling". In: 2012.

[DM17] Emir Demirović and Nysret Musliu. "MaxSAT-based large neighborhood search for high school timetabling". In: *Computers Operations Research* 78 (2017), pp. 172–180. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2016.08.004. URL: https://www.sciencedirect.com/science/article/pii/S0305054816301927.

[Doi19] Paul Doiron. *Backtrack: A Mike Bowditch Short Mystery*. St. Martin's Publishing Group, 2019. ISBN: 9781250259578.

[DS18] Emir Demirović and Peter J. Stuckey. "Constraint Programming for High School Timetabling: A Scheduling-Based Model with Hot Starts". In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Ed. by Willem-Jan van Hoeve. Cham: Springer International Publishing, 2018, pp. 135–152. ISBN: 978-3-319-93031-2.

[DSD20] Dipankar Dutta, Jaya Sil, and Paramartha Dutta. "A bi-phased multi-objective genetic algorithm based classifier". In: *Expert Systems with Applications* 146 (2020), p. 113163. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.113163. URL: https://www.sciencedirect.com/science/article/pii/S0957417419308814.

[Fak] Kirill Fakhroutdinov. *The Unified Modeling Language*. URL: https://www.uml-diagrams.org/.

[Fak13] Kirill Fakhroutdinov. *UML 2.5 diagrams overview*. 2013. URL: https://www.uml-diagrams.org/uml-25-diagrams.html.

[FS14] George H.G. Fonseca and Haroldo G. Santos. "Variable Neighborhood Search based algorithms for high school timetabling". In: *Computers Operations Research* 52 (2014). Recent advances in Variable neighborhood search, pp. 203–208. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2013.11.012. URL: https://www.sciencedirect.com/science/article/pii/S0305054813003328.

[HH04] Randy L Haupt and S E Haupt. *Practical genetic algorithms*. second edition. John Wiley, 2004. ISBN: 9780471455653.

[KS13] Simon Kristiansen and Thomas Riis Stidsen. "A comprehensive study of educational timetabling, a survey". In: *Department of Management Engineering, Technical University of Denmark.(DTU Management Engineering Report* 8 (2013).

[K14] Ahmed Kheiri, Ender Özcan, and Andrew J. Parkes. "A stochastic local search algorithm with adaptive acceptance for high-school timetabling". In: *Annals of Operations Research* 239.1 (2014), 135–151. DOI: 10.1007/s10479-014-1660-0.

[PDdB14] Árton P. Dorneles, Olinto C.B. de Araújo, and Luciana S. Buriol. "A fix-and-optimize heuristic for the high school timetabling problem". In: *Computers Operations Research* 52 (2014), pp. 29–38. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2014.06.023. URL: https://www.sciencedirect.com/science/article/pii/S0305054814001816.

[PDdB17] Árton P. Dorneles, Olinto C.B. de Araújo, and Luciana S. Buriol. "A column generation approach to high school timetabling modeled as a multicommodity flow problem". In: *European Journal of Operational Research* 256.3 (2017), pp. 685–695. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2016.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0377221716305392.

[Pil14] Nelishia Pillay. "A survey of school timetabling research". In: *Annals of Operations Research* 218.1 (2014), pp. 261–293.

[Sav+20] Landir Saviniec et al. "Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems". In: *European Journal of Operational Research* 280.3 (2020), pp. 1064–1081. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2019.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0377221719306538.

[SSC18] Landir Saviniec, Maristela O. Santos, and Alysson M. Costa. "Parallel local search algorithms for high school timetabling problems". In: *Eu-*

ropean Journal of Operational Research 265.1 (2018), pp. 81–98. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.07.029. URL: https://www.sciencedirect.com/science/article/pii/S0377221717306562.

[TB12] Ioannis X. Tassopoulos and Grigorios N. Beligiannis. "Solving effectively the school timetabling problem using particle swarm optimization". In: *Expert Systems with Applications* 39.5 (2012), pp. 6029–6040. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2011.12.013. URL: https://www.sciencedirect.com/science/article/pii/S0957417411016848.

[Ter] TerryGLee. *Overview of Visual Studio*. URL: https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022.

[TIB19] Ioannis X. Tassopoulos, Christina A. Iliopoulou, and Grigorios N. Beligiannis. "Solving the Greek school timetabling problem by a mixed integer programming model". In: *Journal of the Operational Research Society* 71.1 (2019), 117–132. DOI: 10.1080/01605682.2018.1557022.

[YLW20] Shuaipeng Yuan, Tieke Li, and Bailin Wang. "A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times". In: *Expert Systems with Applications* 152 (2020), p. 113360. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113360. URL: https://www.sciencedirect.com/science/article/pii/S0957417420301858.