An-Najah National University

Faculty of Graduate Studies

PANDEMIC-BASED SCHOOL CLASSES SCHEDULING AUTOMATION ALGORITHM: PALESTINIAN SCHOOLS AS A TESTING CASE

By

Eman Mukhaimer

Supervisors

Dr. Amjad Hawash

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Advanced Computing, Faculty of Graduate Studies, An-Najah National University, Nablus - Palestine.

2022

PANDEMIC-BASED SCHOOL CLASSES SCHEDULING AUTOMATION ALGORITHM: PALESTINIAN SCHOOLS AS A TESTING CASE

By

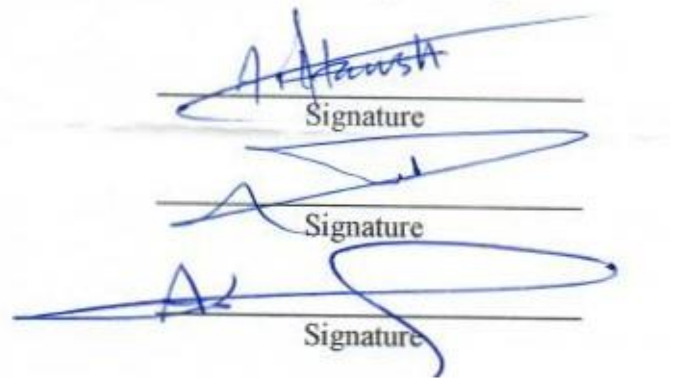
Eman Mukhaimer

This Thesis was Defended Successfully on 29/3/2022 and approved by

Dr. Amjad Hawash
Supervisor

Dr. Yousef Daraghmi
External Examiner

Dr. Ahmed Awad
Internal Examiner



Dedication

Dedicate to my parents. Also, to my students, who were the reason I came up with the idea of this thesis.

Acknowledgements

I would like to thank my supervisor Dr. Amjad Hawash, who made this work possible. His guidance and advice carried me through all the stages of writing my project, and without him, this thesis would be one page.

I would like also to say thank you to all the lecturers who taught me in the master's semesters in Advanced Computing. I have so much fun, and I have gained very good knowledge. That was very helpful in writing the thesis.

In addition, I want to thank the proposal defense committee members, especially Dr. Mohammad Elsaid who in addition for his remarks he helped us in searching for more references.

Also, I would like to thank Dr. Mohamed Matar, the Director-General of the Center for Educational Research and Development at the Ministry of Education, and all those working there to help us obtain permission to access the data on the school's web page. Moreover, the school headmasters\headmistresses, and the secretaries in Burqas schools for helping us to get the data.

Finally, I would like to thank the thesis defense committee members, Dr. Yousef Daraghmi and Dr. Ahmed Awad for their constructive remarks.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

PANDEMIC-BASED SCHOOL CLASSES SCHEDULING AUTOMATION ALGORITHM: PALESTINIAN SCHOOLS AS A TESTING CASE

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: ______________________________

Signature: ______________________________

Date: ______________________________

List of Contents

List of Tables

List of Figures

List of Appendices

PANDEMIC-BASED SCHOOL CLASSES SCHEDULING AUTOMATION ALGORITHM: PALESTINIAN SCHOOLS AS A TESTING CASE

By

Eman Mukhaimer

Supervisor

Dr. Amjad Hawash

ABSTRACT

During the COVID-19 pandemic, the distance learning was proposed as a vital solution to go on with the teaching/learning process and to keep both students and teachers (elementary and/or higher education) in contact with the avoidance of possible infection between them. However, despite the distance education offers and despite its role in eliminating the amounts of infections, it suffers from several drawbacks (in some communities): the lack of distance learning experience for both students and teachers, the need of high students' motivation and the need for sufficient number of devices especially if the family has more than one school/college student. The purpose of this study is to propose a solution for the last problem (considering elementary education) by the proper scheduling of school classes' sessions considering all of the affecting parameters like the number of lessons per teacher, the number of brothers students and the number of devices per family. The study is applied for 4 different school subjects: Arabic, English, Science and Mathematics and the study considers 4 elementary Palestinian schools to be involved in the study. The problem is modelled as an Integer Programming problem, and it is implemented using Gurobi. Comprehensive experimental tests are executed to compare between our work and the manual preparation of lessons scheduling in which a promising result are achieved. The IP algorithm decreased the number of conflicts by 40%.

Keywords: School Timetabling, Scheduling, Integer Programming.

Chapter One
Introduction and Theoretical Background

This chapter introduces a set of issues related to distance learning, especially the lesson scheduling problems during the COVID-19 period. Also, it will discuss the previous work related to the same issue and the motivation behind the work.

Moreover, it will describe the timetabling term as well as taking its related works and its usage in the education process by discussing its current problems and how our work introduced a solution to timetabling problems.

1.1 Background and Motivation

This section introduces the problems that emerged after directing the school's education in Palestine to the distance learning. Especially the lack of electronic devices per family. This situation stimulated the researcher to search for a solution. The solution is represented by the invention of an algorithm to automatically construct timetable. This section is divided into a set of subsections to summarize the problem and introduce the reader to the proposed solution.

1.1.1 Teaching During COVID-19 Pandemic

The COVID-19 pandemic changed people's daily habits in executing and conducting their activities. Education is one of these affected activities in which the way the teaching process is executed is changed from the regular pattern represented in doing it in a closed teaching environment and/or classes. Moreover, the pandemic situation opened the eyes for a possible future situation that could affect the quality of education provided. Several life activities changed, and people found themselves in front of new social protocols according to the still going on pandemic: the need for social distances, the interruption of worldwide travels and the deterioration of the general economic situation. However, the education is one of the main activities that also affected by this situation. Several (if not all) universities, schools, educational centers, conferences, and others are directed to distance learning and online sessions that makes online teaching more than an option [1]. The schools all over the world are forced to close which led to an increased necessity for online learning[1] and the need to find solutions and techniques to make it fruitful as a try to minimize the bad effect of their closure. Despite the distance learning decreases the infection possibility, there is a set of challenges to overcome. According to weforum[2], some students without reliable Internet access and/or technology features struggle to participate in digital learning that emerges a new challenge especially for the poor countries that originally suffer from a set of already existing problems.

The work presented in [2] is related to the study of the obstacles faced by university professors and students during the distance learning. The study concluded that the lack of capabilities to communicate remotely (such as devices, Internet, and applications) is one of the biggest obstacles that prevent the achievement of good quality in distance learning. Moreover, according to [3], in order to get a good online learning, there is a set of factors have to be taken into account: (1) schools need to have the resources to implement remote learning, (2) students need to have access to computers and reliable Internet connections, and (3) parents need to have the ability, time, energy, and patience to turn into home- school instructors.

The case here in Palestine is the same if not worse due to the complex political situation exists. From the beginning of the pandemic, the ministry of education is directed to the distance learning for all the academic institutions (including schools) and used to take the initiative to spread awareness campaigns regarding distance education and held training courses in this regard for teachers to make it easy to conduct the distance learning. This introduced a solution for many students, but there is a problem that can't be neglected: There is a percentage of them couldn't participate because there are not enough devices (computers and handheld) per family, and if there are enough, the Internet bandwidth is another problem.

Due to all previous facts and especially the scarce of availability of handheld devices in Palestinian homes, this study is related to building up an algorithm to automatically

[1] Online learning, distance learning and online or distance teaching or education will be used the same during this research.

[2] https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/

construct lessons schedule per class/school taking into account a set of constraints represented in: (1) the number of available handheld devices per family, (2) the number of brothers students per family as well as (3) the availability of teachers, as a try to minimize the negative impacts emerged from the conversion to distance learning. The automated lessons scheduler is intended to help school administration staff to better conduct the lessons schedule to minimize the number of handheld devices conflict of usage.

1.1.2 Aim of the Study

Our contribution in this research is to implement school lessons scheduler considering the set of constrains mentioned in the previous section. The algorithm considers the avoidance in conflict with respect to devices usage per family as well as the conflict of lessons per teacher. The intention of the algorithm is to maximize the utilization of the computation devices per family by preparing a lessons schedule with the ability to minimize the negative impact of the set of existing mentioned constraints.

1.2 Timetabling

1.2.1 Timetabling Definition

Computer solution of timetabling, scheduling and rostering problems has been addressed in the literature since the 1950's according to [5] in which timetabling is defined as: "Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way to satisfy as nearly as possible a set of desirable objectives."

According to [6], a timetabling problem has four parameters: T, a finite set of times; R, a finite set of resources; M, a finite set of meetings; and C, a finite set of constraints. The problem is to assign times and resources to the meetings to satisfy the constraints as far as possible.

[7] also defined the problem of timetabling as the assignment of available resources to objects placed in time in such a manner to satisfy as many desirable objective targets as possible under the relevant constraints.

Timetabling covers various forms of real-world problems, including Employee Timetabling, Transport, Sports Timetabling and Educational Timetabling.

1.2.2 Educational Timetabling

The work presented in [8] classified educational timetabling into three main categories:

- School timetabling: The weekly scheduling for all the classes of high school, avoiding the possible conflicts of having two classes or more for the same teacher at the same time, and vice versa.

- Course timetabling: The weekly scheduling for all the lectures of a set of university courses, minimizing the overlaps of lectures of courses having common students.

- Examination timetabling: The scheduling for the exams of a set of courses, avoiding overlapping exams of courses having common students, and spreading the exam for the students as much as possible.

1.2.3 School Timetabling Definition

The work published in [9] summarized the definition of the high school timetabling problems as the availability of teachers and rooms, a number of lessons to be taught by teachers to specific classes or students and a set of constraints. To build a timetable, we need to assign resources, such as times, teachers, students, and rooms to a collection of lessons and minimize the constraint violations. For example, teachers must teach specific lessons, in particular rooms, to specific classes or students at specific times. Constraints are a set of conditions that should be satisfied in a solution, and they are categorized into two types: hard and soft. A valid solution must satisfy all the hard constraints; this solution will be a feasible solution. However soft constraints can be violated, and the quality of the solution is measured on the number of soft constraints satisfied, where each soft constraint violation can be weighted differently with a given penalty. More details can be found in subsection 1.2.7.

1.2.4 Literature Review

School timetabling is one type of educational timetabling, which is considered an ongoing challenging optimization problem and has been proved as an NP-hard problem

[10]. That's why it is had a respectful consideration in literature and makes it of interest for topics related to Operational Research and Artificial Intelligence.

Several works have been conducted throw the decades to study the timetable construction problems as well as the efforts pays to solve them or at least to minimize their drawbacks [8, 7, 10, 9].

From all the techniques mentioned in the above surveys, we are going to take Integer Programming (IP) as a solution method. According to [18] the main advantage of Integer Programming over heuristic methods is the ability to certificate optimality. Although approaching optimality requires too much time in IP, we were apple to get the nearest optimal solution in a reasonable time for our model.

[11] defines linear programming and integer programming as:

"In a linear program, there are variables, constraints, and an objective function. The variables, or decisions, take on numerical values. Constraints are used to limit the values to a feasible region. These constraints must be linear in the decision variables. The objective function then defines which assignment of feasible values to the variables is optimal: it is the one that maximizes (or minimizes, depending on the type of the objective) the objective function. The objective function must also be linear in the variables. Integer programming adds additional constraints to linear programming. An integer program begins with a linear program and adds the requirement that some or all the variables take on integer values."

The work related to [12] was one of the earliest studies employing IP to solve the school timetabling problem. The use of IP to solve the school timetabling problem has grown since this early study according to [10].

The work conducted in [13] used IP to solve the school timetabling problem for Greek high schools. They developed the basic structure, modeled the problem as an IP problem, and tested the model on a typical Greek high school, this model provided an optimal solution for the specific constraints of the problem. This work extended further in [14].

[15] in his work presented an IP formulation for a variant of the Class-Teacher Timetabling problem, which considers the satisfaction of teacher preferences and the proper distribution of lessons throughout the week. They adapted a cut and column generation algorithm to solve the linear relaxation, this algorithm provided a strong lower bound in a reasonable time.

The work presented in [16] proposed one model for high school timetable generation, which uses two-phase linear IP to solve the problem. This reduces the required computation time, by decomposing the problem to determine the day and then, in the second phase, to generate a daily schedule. The experimental results demonstrated the numerical efficiency of the two-phase approach, in comparison to solving the original problem.

A column generation approach is presented in [17] for the solution of timetabling problems. This approach can solve difficult problems in acceptable time while producing high-quality results. They have applied this approach on the Greek high schools and achieved solutions with no idle hour for any of the teachers.

Dealing with a uniform way of modeling optimization problems was the work of [18] that described a method capable of handling an arbitrary instance of the XHSTT format. This XHSTT format [19] for high school timetabling provides a uniform way of modeling problem instances and corresponding solutions. The format supports a wide variety of constraints, and a few real-life instances from different countries. Kristiansen method is based on a Mixed Integer Programming (MIP) model, which is solved in two steps with a commercial general-purpose MIP solver. Computational results show that his approach can find previously unknown optimal solutions for 2 instances of XHSTT and proves optimality of 4 known solutions.

The work related to [20] proposed an IP based on the multi-commodity flow model for Brazilian high school timetabling problems. Dantzig–Wolfe decomposition principles are applied to obtain an alternative formulation then a column generation approach is used to solve linear relaxation of the alternative formulation. The lower bounds obtained through this approach are tight and can be generated faster than previous approaches reported in the literature.

Finally, the work presented in [21] dealt with the school timetabling problem for the case of Greek high schools. The problem is modeled as MIP and it is implemented using Gurobi and CPLEX. Two methodologies were proposed: the first deals with the problem utilizing a model that includes all hard and soft constraints, called “monolithic” model, and the second is based on a decomposition of the problem to six sub-problems.

Our work differs from the previous ones by dealing with the school timetabling problem during the COVID-19 pandemic. An online learning timetable requires the same hard constraints correspond to teachers, classes, and material. Soft constraints related to this new situation that requires minimum number of devices to be used during the online session. So, we formulate an IP model and implemented it using Gurobi.

1.2.5 Computational Complexity

According to [10], the school timetabling problem is an NP-complete or NP-hard problem depending on the constraints associated with the problem.

P, NP, NP- Complete and NP-Hard Problem Definition:

The P set of problems are polynomial algorithms that can be solved in polynomial time.

The NP set of problems cannot be solved in polynomial time. But they can be verified in polynomial time.

A problem is classified as NP-Hard when an algorithm for solving it can be translated to solve any NP problem. Then we can say, this problem is at least as hard as any NP problem, but it could be much harder or more complex.

NP-Complete problems are problems that are both NP and NP-Hard problem. This means that NP-Complete problems can be verified in polynomial time and that any NP problem can be reduced to this problem in polynomial time.

1.2.6 Terminology

This subsection defines the terms to be used in the research.

- Class: refers to a group of students that will be taught a particular subject.

- Grade: refers to the level of schooling in which each grade contains one or more classes.

- Subject: refers to the actual content being taught, e.g., English or Mathematics. The subjects to be taught to each grade are determined by a curriculum which is usually set by a governing educational body.

- Lesson: refers to a particular subject being taught to a class by a teacher. Several lessons are taught for each subject. These lessons are also referred to as meetings between a teacher and a class.

- Period: is a timetable slot which a given lesson can be scheduled in. The duration of periods differs from school to another.

- Idle: or free period for a teacher is a period in which the teacher does not teach. Similarly, an idle or free period for a class is a period in which the class is not assigned a lesson.

- Resource refers to any entity involved in a lesson. The standard resources are a class, a teacher, and the room in which the lesson is held.

1.2.7 Problem Constraints

In the work published in [23], 13 constraints have been listed and grouped into three groups: (1) constraints describing the basic scheduling problem, (2) others for events and courses, and (3) finally for resources. Here are the constrains as mentioned in [23].

- Basic scheduling constraints:

- Assign Time Constraint (Cost per event). Assign a time to each of the selected events.

- Assign Resource Constraint (Cost per event). Assign a resource to the role in each of the selected events.

Both constraints have a variant expressing the preference for the time, respectively the resource: Prefer times Constraint and Prefer Resources Constraint.

- Event constraints:

- Link Events Constraint (Cost per event group). Schedule the selected event groups at the same (starting) time.

- Spread Events Constraint. (Cost per event group). Schedule the events of the selected event groups to the selected time groups between a minimum and a maximum number of times.

- Avoid Split Assignments Constraint. (Cost per event group). For each selected event group, schedule the selected role of each event of this group to the same resource.

- Resource constraints:

- AvoidClashesConstraint. Schedule the selected resources without clashes. This is one of the basic (hard) constraints.

- AvoidUnavailableTimesConstraint. Avoid that the selected resources are busy in the selected times.

- LimitWorkloadConstraint. Schedule workload to the selected resources between a minimum and a maximum.

- LimitIdleTimesConstraint. The number of idle times in the selected time groups should lie between a minimum and a maximum for each of the selected resources. Typically, the time groups are a day or all days.

- LimitBusyTimesConstraint. The number of occupied times for the selected resources should lie between a minimum and a maximum for each of the selected time groups. Typically, the time groups are the days.

- ClusterBusyTimesConstraint. The number of time groups with an assigned time should lie between a minimum and a maximum for the selected resources. Typically, the time groups are days; for example, a teacher requiring at most 3 days with lessons.

However, authors of the work [7] listed the primary hard and soft constraints of High School Timetabling, here are some of them:

- Primary hard constraints:
- No lessons conflicts: A class can't have two lessons at the same period.
- A subject must be scheduled for the required number of lessons for each class.
- Primary soft constraints:
- Limit idle periods for students and/or teachers.
- Lessons spreading.

The number of different constraints to be added to the timetabling problem depends on the educational system, so it varies from country to country [9]. The constraints we used for our timetabling problem are: No lessons conflict for the teachers, no lesson conflict for the class, and the subject must be scheduled for the required number of lessons for each class; those are the hard constraints. In addition, we add a constraint for lesson spreading as a soft constraint.

1.2.8 Techniques Applied to the High School Timetabling Problem

Authors of the work published in [9] have reviewed the state-of-the-art of algorithms applied to the high school timetabling problem. Accordingly, these algorithms can be classified into one or more of the following types:

- **Mathematical optimisation algorithms:** Mathematical algorithms such as Integer Programming and Constraint Programming assume the objective function and the constraints are linear system and restrict some or all the problem variables into integer values.

- **Meta-heuristic algorithms** are very general-purpose problem-solvers designed to find an acceptable solution in a reasonable amount of computational time. Example of these algorithms are:

- Population-based algorithms: maintain and work on set of candidate solutions in which each solution corresponding to a point in the problem search space.

- single solution-based algorithms: maintain a single candidate solution and use move operator to explore the area around the current solution.

- **Graph coloring algorithms:** use graph theory to represent the problem variables. Classes and teachers can be represented as vertices. Each class is linked to each teacher by edges. Given p colors (each period corresponding to a color), the problem consists of finding an assignment of a color to each edge such that no two adjacent edges have the same color [8].

- **Matheuristics approaches:** combine the strengths of heuristic method with the mathematical optimization algorithms. One of the types of research [9] referred as an example is [24]. A mixed-integer linear programming and fix-and-optimize heuristic combined with variable neighborhood descent is used to solve high school timetabling problems in Brazil.

- **Hyper-heuristic approaches:** use a set of heuristic and a selection method to automate the selection of which heuristic should be applied. Examples can be found in [9].

- **Hybrid approaches:** combine the strengths of several (two or more) meta-heuristic algorithms in a unified framework. Such as using Simulated Annealing with Iterative Local Search as in [25].

We used IP as an optimization algorithm for our timetabling problem. Since many recent studies adopted this method and there are good commercial optimizers for solving IP.

1.3 Palestine's Education System

In Palestine, there are three types of schools classified according to their student's gender: Boys schools, girls' schools, and co-educational ones.

The educational school system in Palestine consists of 12 grades divided into two categories: the mandatory education that includes grades from 1 to 10 and is divided into preparation level from grade 1 to 4, and empowerment level from grade 5 to 10. The second category is the secondary education, which is not obligatory and includes both grades 11 and 12. The subjects and the number of weekly lessons for each subject per each grade is determined by the Ministry of education Table 1 in Appendix A.

The number of classes contained in some school depends on the grades it (the school) has and the number of classes for each grade in the school. Each student is enrolled in a class at the beginning of the academic year, and each class has its own room in the school. As for the teachers, each teacher is pre-assigned to the subject and class at the beginning of the academic year as well Table 2 in Appendix A. Usually, the headmaster of the school constructs the timetable manually at the beginning of each academic year.

During the Covid-19 pandemic and the academic year 2019-2020, the administration staff in the ministry of education adopted blending learning. They directed the school headmaster/headmistress to make timetables that can be used for online learning. They also recommended the splitting of the time between the categories mentioned above, so to minimize the lesson conflicts within the family. We used all that information to build our school timetable, there is more details in chapter 3.

1.4 Overview of the Thesis

This thesis consists of four chapters. This chapter presents the background, motivation the aims of the study and literature review. It also describes the main terms considering the timetabling problem, the technique applied to solve the problem, and a brief description of Palestine’s education system.

Chapter two introduces the methodology: the definition of the Integer Programming, the methods for solving it, an example for each method, and the software used for implementing it. The second section describes the Integer Programming formulation: the constraints and the objective function. A case study is provided in section three to explain the Integer Programing problem in more detail.

In chapter three, we conducted an experimental evaluation of the proposed model to test its accuracy. Finally, the conclusion is provided in chapter four.

Chapter Two
Methodology

This chapter gives a short introduction in Integer Programming, the IP model for our school timetabling problem, and a case study in the third section.

2.1 Integer Programming

IP problems are Linear Programming (LP) problems in which the variables are constrained to have integer values.

Most integer programming problems are classified as hard optimization problems, and many integer programming problems belong to the class of NP-hard problems. So, while a general linear programming problem may be solvable in polynomial time, finding an optimal integer solution to the same formulation usually requires an exponential amount of computation time [26]. However, the same work classified integer programming as follows: Mathematical programming problems in which all decision variables must have positive integer values are called general integer programming problems. If all the decision variables are restricted to have only the values zero or one, the problem is then called a zero–one programming (or binary integer programming) problem. In this case, the constraints on the variables are sometimes called binary or Boolean constraints, and the model is often referred to in abbreviated form as a 0–1 problem. Variations on problems arise if some of the variables must be integer, others must be zero or one, while still others may have real values. Any problem involving such combinations is described as a MIP problem.

Many real-life problems restricted their involved variables to be integer values. For example, in a manufacturing scenario, it would be difficult to implement a solution that specifies producing 10.4 cars or 7.2 tables. Fractional values are infeasible.

2.1.1 Integer Programming Algorithms

Let's take the pure integer programming problem:

Maximize

$$z = c^T \boldsymbol{x}$$

subject to

$Ax = b$

$x \geq 0$

x_j =integer $\forall j \in I$ **(2.1)**

Here A is an $m \times n$ matrix, c is an $n \times 1$ column vector, x is an $n \times 1$ column vector,

$I = \{1, 2, ..., n\}$, and b is an $m \times 1$ column vector.

Note that $x \geq 0$ means that $x_j \geq 0 \; \forall j \in I$.

[27] summarizes the strategy of all the algorithms for solving IP 2.1 in three steps.

- **Step 1** Relax the solution space of the IP by deleting the integer restriction on all integer variables.
- **Step 2** Solve the LP and identify its continuous optimum.
- **Step 3** Starting from the continuous optimum point, add special constrains that iteratively modify the LP solution space in a manner that will eventually render an optimum extreme point satisfying the integer requirements.

Two methods have been developed for generating the special constraints in step3:

- Branch and Bound (B&B).
- Cutting Planes Method.

However, according to [27] neither method is effective computationally, but experience shows that the B&B method is far successful than the cutting planes method.

2.1.2 Cutting Planes Method

The cutting planes algorithms according to [28] were developed by Ralph Gomory in 1958. The algorithm works as follows: we begin with the problem given by 2.1, and we solve it with the simplex method without the restriction that the solution should be an integer. If the solution is an integer, we are done. If not, a new constraint that "cut off" (eliminates) some non-integral solutions is added, including the ones just obtained by the simplex method. The new problem is then solved, and the simplex algorithm is repeated. We finally get an optimal integer solution when we add enough constraints.

The cutting planes algorithm consist of three steps:

step1 Solve the LP according to 2.1.

step2 If the solution is integer, then it must be optimal.

step3 Otherwise, generate a cutting plane that excludes the current LP solution, but does not exclude any integer points, and then return to Step 1.

There are many ways for obtaining a cutting plane, one of the best known is called a Gomory fractional cut according to [26] while [28] explains how to generate Gomory cutting plane as follows:

If we suppose that the problem a solution which is feasible and has a finite optimal solution. Also, **A**, **B** and **c** in 2.1 have integer entries. If we take the final simplex table in the LP problem, then i_{th} constraint will be:

$$\sum_{j=1}^{n} t_{ij} x_j = x_{B_i} \tag{2.2}$$

here x_{B_i} represents the value of the i_{th} basic variable in the optimal solution of the LP problem. Let $\lfloor a \rfloor$ be the greatest integer less than a. For example, $\lfloor 3/2 \rfloor = 1$ and $\lfloor -2/3 \rfloor = -1$. Since $\lfloor t_{ij} \rfloor \leq t_{ij}$ and $x_j \geq 0$, then 2.2 can be written as

$$\sum_{j=1}^{n} \lfloor t_{ij} \rfloor x_j \leq x_{B_i} \tag{2.3}$$

If an integer vector x satisfies 2.2, then it will satisfy 2.3. Moreover, the left-hand side of 2.3 is an integer, so it must be less than or equal $\lfloor x_{B_i} \rfloor$. This can be written as:

$$\sum_{j=1}^{n} \lfloor t_{ij} \rfloor x_j \leq \lfloor x_{B_i} \rfloor \tag{2.4}$$

Constraint 2.4 can be transformed into an equation by introducing the slack variable u_i:

$$\sum_{j=1}^{n} \lfloor t_{ij} \rfloor x_j + u_i = \lfloor x_{B_i} \rfloor \tag{2.5}$$

If we suppose x_{B_i} is not an integer. We can write $\lfloor x_{B_i} \rfloor + f_i = x_{B_i}$ and $\lfloor t_{ij} \rfloor + g_{ij} = t_{ij}$ where $0 < f_i < 1$ and $0 < g_{ij} < 1$. Here, f_i is denoted by the fractional part of x_{B_i}.

For example, if $t_{ij} = \frac{10}{3} = 3\ \frac{1}{3}$, then $g_{ij} = \frac{1}{3}$.

Another example, if $t_{ij} = -\frac{5}{4} = -(1\ ¼)$, then $\lfloor t_{ij} \rfloor = -2$ and $g_{ij} = \frac{3}{4}$.

If we subtract 2.2 from 2.5, we have

$$\sum_{j=1}^{n}(-g_{ij})x_j + u_i = -f_i \qquad (2.6)$$

Equation 2.6 is the cutting plane constraint that will be added to the constraints in 2.1.

Since the coefficient of all the basic variables except the i_{th} one in the list will be zero in 2.2, it may be written as

$$x_{ri} + \sum_{j \in N} t_{ij}x_j = x_{B_i} \qquad (2.7)$$

where N is the set of indices of the non-basic variables and where the variable labeling this i_{th} constraint is x_{ri}. Let x_{ri} be a non-integer variable then we can obtain the Gomory cutting plane from 2.7 and write it as:

$$\sum_{j \in N}(-g_{ij})x_j + u_i = -f_i \qquad (2.8)$$

where $\lfloor x_{B_i} \rfloor + f_i = x_{B_i}$ and $\lfloor t_{ij} \rfloor + g_{ij} = t_{ij}$.

We will explain the method by an example. More details and more examples can be found in [28], [27] and [26].

Example 2.1.1.

Maximize

$$\boldsymbol{z = x + y}$$

subject to

$$2x + 3y \le 12$$

$$2x + y \le 6$$

where $x, y \in Z$.

First, we solve the problem as a linear programming using the simplex method dropping the integrality requirement. To do so, we first insert the slack variables and find the slack equations instead of the inequalities,

$$2x + 3y + u = 12$$

$$2x + y + v = 6$$

after that we rewrite the objective function to match the format of the slack equations.

$$\text{Maximize } z = x + y + 0u + 0v$$

then we write the initial simplex Table 2.1a. After solving the problem, the final table is given in Table 2.1b. As seen, $x = \frac{3}{2} = 1\ \frac{1}{2}$ is not an integer and y is integer, so we

will take the second row to obtain the cutting plane. Here $f = \frac{1}{2}$ and

$$\left\lfloor -\frac{1}{4} \right\rfloor + \frac{3}{4} = -\frac{1}{4}$$

$$\left\lfloor \frac{3}{4} \right\rfloor + \frac{3}{4} = \frac{3}{4}$$

The Gomory cutting plane is then

$$-\frac{3}{4}u + -\frac{3}{4}v + s = -\frac{1}{2}$$

Table 2. 1

First Step Tables for Example 2.1.1.

(a)

Basic	x	y	u	v	Solution
z	-1	-1	0	0	0
u	2	3	1	0	12
v	2	1	0	1	6

(b)

Basic	x	y	u	v	Solution
z	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{9}{2}$
y	0	1	$\frac{1}{4}$	$-\frac{1}{2}$	3
x	1	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{3}{2}$

Table 2.2

Second Step Tables for Example 2.1.1.

(a)

Basic	x	y	u	v	s	Solution
z	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{9}{2}$
y	0	1	$\frac{1}{4}$	$-\frac{1}{2}$	0	3
x	1	0	$\frac{1}{4}$	$\frac{3}{4}$	0	$\frac{3}{2}$
s	0	1	$-\frac{3}{4}$	$-\frac{3}{4}$	1	$-\frac{1}{2}$

(b)

Basic	x	y	u	v	s	Solution
z	0	0	0	0	$\frac{1}{3}$	$\frac{13}{3}$
y	0	1	0	-1	$\frac{2}{3}$	$\frac{8}{3}$
x	1	0	0	1	$\frac{1}{3}$	$\frac{5}{3}$
u	0	1	1	1	$-\frac{4}{3}$	$\frac{2}{3}$

Our new table is Table 2.2a. The dual simplex is used to solve the problem and to go back to feasibility, so Table 2.2b is obtained.

Because the solution is not integral, we make another Gomory cutting plane using the y row.

$$\lfloor -1 \rfloor + 0 = -1$$

$$\left\lfloor \frac{2}{3} \right\rfloor + \frac{2}{3} = \frac{2}{3}$$

$$\left\lfloor \frac{8}{3} \right\rfloor + \frac{2}{3} = \frac{8}{3}$$

The cutting plane equation is:

$$0v + -\frac{2}{3}s + r = -\frac{2}{3}$$

The new table is Table 2.3a.

Using dual simplex method on Table 2.3a , we obtain Table 2.3b.

We have found the optimal integer solution

$$x = 2, \quad y = 2, \quad z = 4$$

The set of feasible solutions for the pure IP problem is the set of points with integer coordinates, this set is called lattice points, lying within the convex region defined by constraints (Figure 2.1a).

The first cutting plane,

$$-\frac{3}{4}u + -\frac{3}{4}v + s = -\frac{1}{2} \qquad (2.9)$$

can be written in terms of *x* and *y* by substituting 2.10 and 2.11 in 2.9

$$u = 12 - 2x - 3y \qquad (2.10)$$

$$v = 6 - 2x - y \qquad (2.11)$$

to have equation 2.12.

$$3x + 3y + s = 13 \qquad (2.12)$$

we can write it as constraint as in 2.13.

$$3x + 3y \leq 13 \qquad (2.13)$$

see Figure 2.1b.

The second cutting plane,

$$0v + -\frac{2}{3}s + r = -\frac{2}{3} \qquad (2.14)$$

gives

$$3x + 3y \leq 12 \tag{2.15}$$

see Figure 2.1c.

The main disadvantages associated with the Gomory fractional cut method are as [28] and [27] mentioned:

- Integer solutions are not obtained until the very end, unlike the branch-and-bound method that discussed in the next subsection. Pure cutting plane methods are therefore not considered to be very practical for large problems.

- The cutting planes are built by using fractional parts of the coefficients and the righthand side of a constraint. This will cause a round-off errors which will make the method to converge slowly.

Table 2.3

Third Step Tables for Example 2.1.1.

(a)

Basic	***x***	***y***	***u***	***v***	**s**	**r**	**Solution**
z	0	0	0	0	$\frac{1}{3}$	0	$\frac{13}{3}$
x	0	1	0	-1	$\frac{2}{3}$	0	$\frac{8}{3}$
y	1	0	0	1	$-\frac{1}{3}$	0	$\frac{5}{3}$
u	0	0	1	1	$-\frac{4}{3}$	0	$\frac{2}{3}$
r	0	0	0	0	$-\frac{2}{3}$	1	$-\frac{2}{3}$

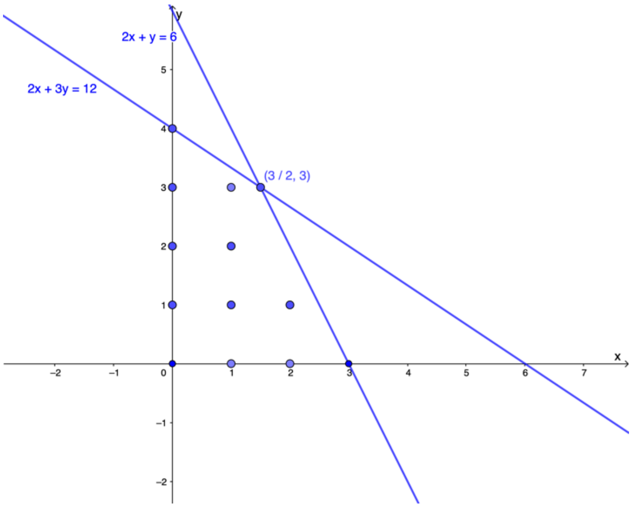
(b)

Basic	***x***	***y***	***u***	***v***	**s**	**r**	**Solution**
z	0	0	0	0	0	$\frac{1}{2}$	4
x	0	1	0	-1	0	1	2
y	1	0	0	1	0	$-\frac{1}{2}$	2
u	0	0	1	1	0	2	2
s	0	0	0	0	1	$-\frac{3}{2}$	1

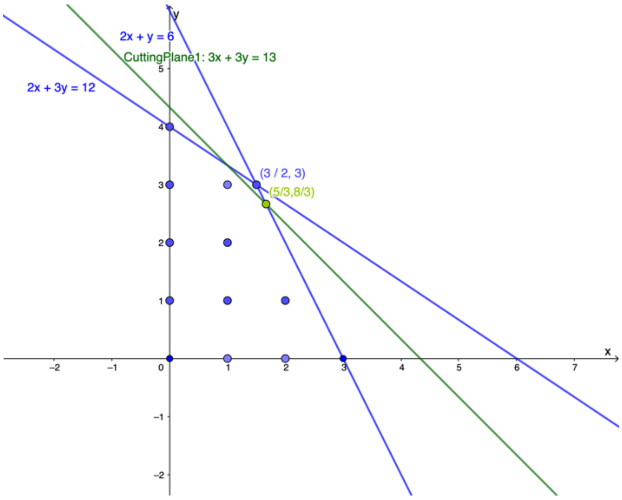
Figure 2.1

Geometric Representation for Example 2.1.1.

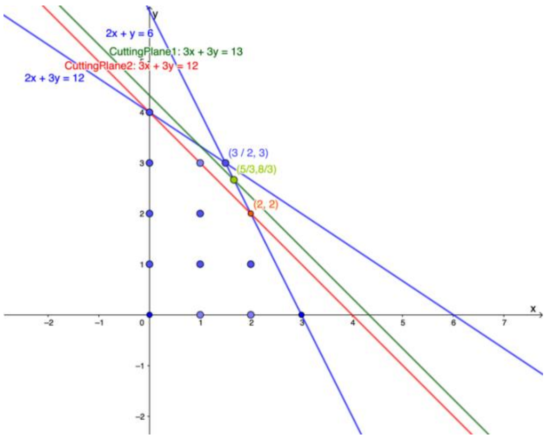
(a) *The lattice points for the LP in example 1.*



(b) *The first cutting plane.*



(c) *The second cutting plane.*



2.1.3 Branch and Bound Method (B&B)

The first B&B algorithm was developed in 1960 by A.Land and G. Doig for the general, mixed and pure IP problem. R. J. Dakin 1966 modified the Land-Doig method. This later method is widely used in the computer codes for IP.

Dakin's method procedure is explained in detail in [28]. Here an example on this method.

Example 2.1.2.

Consider the pure integer programming problem

Maximize

$$z = x + 4y$$

subject to

$$x + 6y \leq 36$$

$$3x + 8y \leq 60$$

$$x \geq 0, y \geq 0$$

where x, $y \in$ Z. We use the simplex method to find the solution for the related IP. The final table is Table 2.4c in which we see that an optimal solution is $x = 7\frac{1}{5}$, $y = 4\frac{4}{5}$, $z = \frac{132}{5}$.

If we pick x to be the branching variable. The constraints that will be added are:

$$x \leq 7 \text{ and } x \geq 8$$

To move on to the next step we add each of these constraints to the final table 2.4c. We get the tables shown in table 2.5a and 2.5b. This is how it is done:

Table 2.4

First Step Tables for Example 2.1.2.

(a)

Basic	**x**	**y**	**u_1**	**u_2**	**Solution**
z	-1	4	0	0	**0**
u_1	1	6	1	0	**36**
u_2	**3**	**8**	**0**	**1**	**60**

(b)

Basic	x	y	u_1	u_2	Solution
z	$-\frac{2}{6}$	0	$\frac{4}{6}$	0	24
y	$\frac{1}{6}$	1	$\frac{1}{6}$	0	6
u_2	$\mathbf{\frac{10}{6}}$	0	$-\mathbf{\frac{8}{6}}$	1	12

(c)

Basic	**x**	**y**	**u_1**	**u_2**	**Solution**
z	0	0	$\frac{2}{5}$	$\frac{1}{5}$	$\mathbf{\frac{132}{5}}$
y	0	1	$\frac{3}{10}$	$-\frac{1}{10}$	$\mathbf{\frac{24}{5}}$
x	**1**	**0**	$-\mathbf{\frac{4}{5}}$	$\mathbf{\frac{3}{5}}$	$\mathbf{\frac{36}{5}}$

We add a slack variable u_3 to the constraint $x \leq 7$, this will turn the constraint to:

$$x + u_3 = 7$$

using the row for x in Table 2.4c. We can turn the above constraint into:

$$-\left(-\frac{4}{5}u_1 + \frac{3}{5}u_2\right) + u_3 = 7 - \frac{36}{5} = -\frac{1}{5}$$

In the same way the new constraint for $x \geq 8$ or $-x \leq -8$ is:

$$-\frac{4}{5}u_1 + \frac{3}{5}u_2 + u_3 = -8 + \frac{36}{5} = -\frac{5}{4}$$

We now apply the dual simplex method to each of these tables to obtain Tables 2.5c and 2.5d, respectively.

Table 2.5

Second Step Tables for Example 2.1.2 Considering x as a Branching Variable.

(a)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	$\frac{2}{5}$	$\frac{1}{5}$	0	$\mathbf{\frac{132}{5}}$
y	0	1	$\frac{3}{10}$	$-\frac{1}{10}$	0	$\mathbf{\frac{24}{5}}$
x	1	0	$-4/5$	$3/5$	0	$\mathbf{36/5}$
u_3	0	0	$\mathbf{4/5}$	$\mathbf{-3/5}$	1	$\mathbf{-1/5}$

(b)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	$\frac{2}{5}$	$\frac{1}{5}$	0	$\mathbf{\frac{132}{5}}$
y	0	1	$\frac{3}{10}$	$-\frac{1}{10}$	0	$\mathbf{\frac{24}{5}}$
x	1	0	$-\frac{4}{5}$	$\frac{3}{5}$	0	$\mathbf{\frac{36}{5}}$
u_3	0	0	$-\mathbf{\frac{4}{5}}$	$\mathbf{\frac{3}{5}}$	1	$-\mathbf{\frac{4}{5}}$

(c)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	$\frac{2}{3}$	0	$\frac{1}{3}$	$\mathbf{\frac{79}{3}}$
y	0	1	$\frac{1}{6}$	0	$-\frac{1}{6}$	$\mathbf{\frac{29}{6}}$
x	1	0	0	0	1	7
u_2	0	0	$-\mathbf{\frac{4}{3}}$	1	$-\mathbf{\frac{5}{3}}$	$\mathbf{\frac{1}{3}}$

(d)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	26
y	0	1	0	$\frac{1}{8}$	$\frac{3}{8}$	$\mathbf{\frac{9}{2}}$
x	1	0	0	0	-1	8
u_1	0	0	1	$-\mathbf{\frac{3}{4}}$	$-\mathbf{\frac{5}{4}}$	1

Figure 2.2 shows the tree we have at this point. The objective function value in node 2 is larger than that in node 3. So, node 3 is added to our list of dangling nodes. After that, we continue from node 2. The constraints

$$y \leq 4 \text{ and } y \geq 5$$

will be added to the problem in node 2 to obtain two new problems. The new tables are shown in Table 2.6a and 2.6b. As before, we apply dual simplex to solve each of these tables to obtain Tables 2.6c and 2.6d.

As you can see in Figure 2.3, nodes 4 and 5 are terminal nodes since the values of the variables are integers. But node 5 is better, because it has a larger objective function value. After that, we examine the list of dangling nodes to see if there are other branches of the tree with better objective function. The dangling node we have has the objective function value z=26, which is equal to the value obtained in node 5. Making a branch at this point will not make the objective function value better. So, the optimal solution is:

$$x = 6 \text{ and } y = 5$$

In the above solution at step 2, we have chosen the non-integer variable arbitrary. But as we mentioned in step 2, we should take the variable whose value has the largest fractional part. (There are more rules for choosing this variable, but this is the easiest one as [28] mentioned). i.e., the constraints that should be added are:

$$y \leq 4 \text{ and } y \geq 5$$

adding this constraint to table 2.4c as before gives Tables 2.7a and 2.7b.

Table 2.6

Third Step Tables for Example 2.1.2 Considering x as a Branching Variable.

(a)

Basic	x	y	u_1	u_2	u_3	u_4	Solution
z	0	0	$\frac{2}{3}$	$\frac{1}{3}$	0	0	$\mathbf{\frac{79}{3}}$
y	0	1	$\frac{1}{6}$	0	$-\frac{1}{6}$	0	$\mathbf{\frac{29}{6}}$
x	1	0	0	0	1	0	7
u_2	0	0	$-\frac{4}{3}$	1	$-\frac{5}{3}$	0	$\mathbf{\frac{1}{3}}$
u_4	0	0	$-\mathbf{\frac{1}{6}}$	0	$\mathbf{\frac{1}{6}}$	1	$-\mathbf{\frac{5}{6}}$

(b)

Basic	x	y	u_1	u_2	u_3	u_4	Solution
z	0	0	$\frac{2}{3}$	0	$\frac{1}{3}$	0	$\mathbf{\frac{79}{3}}$
y	0	1	$\frac{1}{6}$	0	$-\frac{1}{6}$	0	$\mathbf{\frac{29}{6}}$
x	1	0	0	0	1	0	7
u_2	0	0	$-\frac{4}{3}$	1	$-\frac{5}{3}$	0	$\mathbf{\frac{1}{3}}$
u_4	0	0	$\mathbf{\frac{1}{6}}$	0	$-\mathbf{\frac{1}{6}}$	1	$-\mathbf{\frac{1}{6}}$

(c)

Basic	x	y	u_1	u_2	u_3	u_4	Solution
z	0	0	0	0	1	4	23
y	0	1	0	0	0	1	4
x	1	0	0	0	1	0	7
u_2	0	0	0	1	3	8	7
u_3	0	0	1	0	-1	-6	5

(d)

Basic	x	y	u_1	u_2	u_3	u_4	Solution
z	0	0	1	0	0	2	26
y	0	1	0	0	0	-1	5
x	1	0	1	0	0	6	6
u_2	0	0	-3	1	0	-10	2
u_3	0	0	-1	0	1	-6	1

Solving both tables using dual simplex gives Tables 2.7c and 2.7d.

At this point we have the tree at Figure 2.4. As you can see the objective function in node 3 is greater than node 2. So, node 2 is added to the list of dangling nodes. Node 3 is terminal since the variables are integers. Then we examine the list of dangling nodes to check that if the other branches of the tree should be explored. The dangling node that has the objective function value $z = 25\frac{1}{3}$ is the only one, which is less than the value obtained in node 3. As before making a branch at this point will not improve the objective function value, so the optimal solution to the problem is:

$$x = 6 \text{ and } y = 5.$$

This solution is the same solution we found at the first attempt, but here we reach the optimal solution much faster.

Figure 2.2

First Branch in the Branch and Bound Tree for Example 2.1.2 Considering x as a Branching Variable.

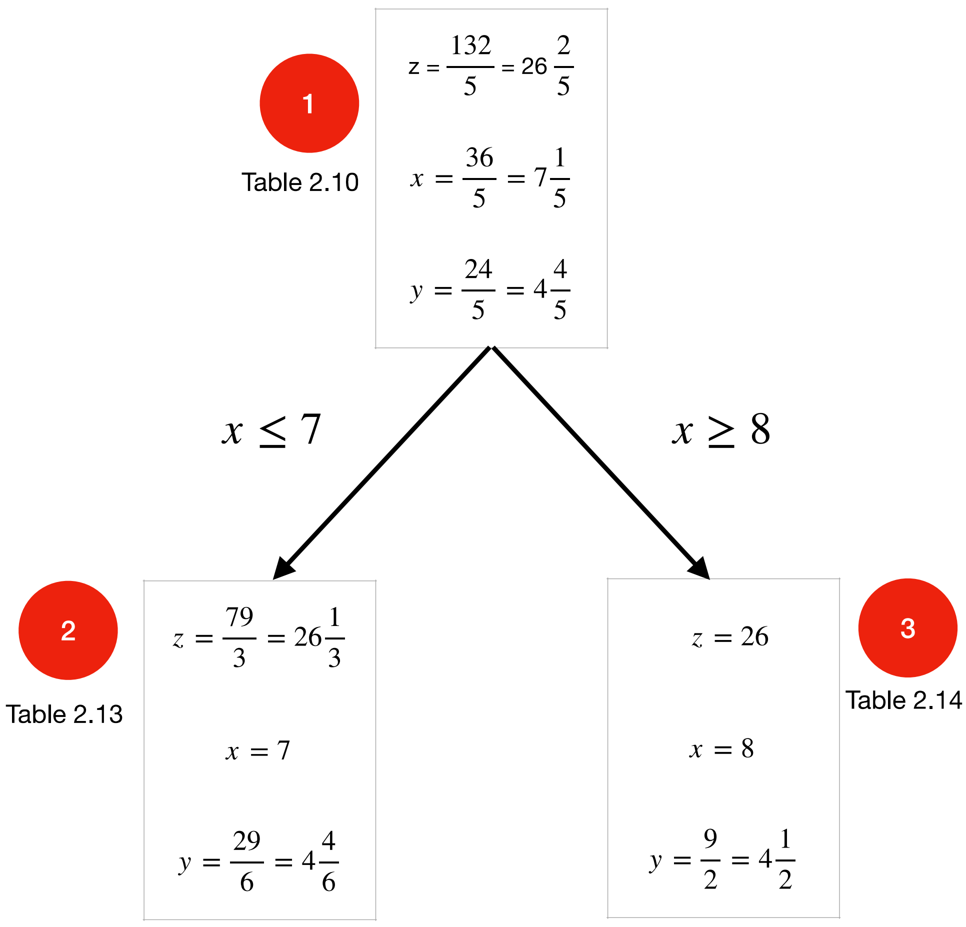
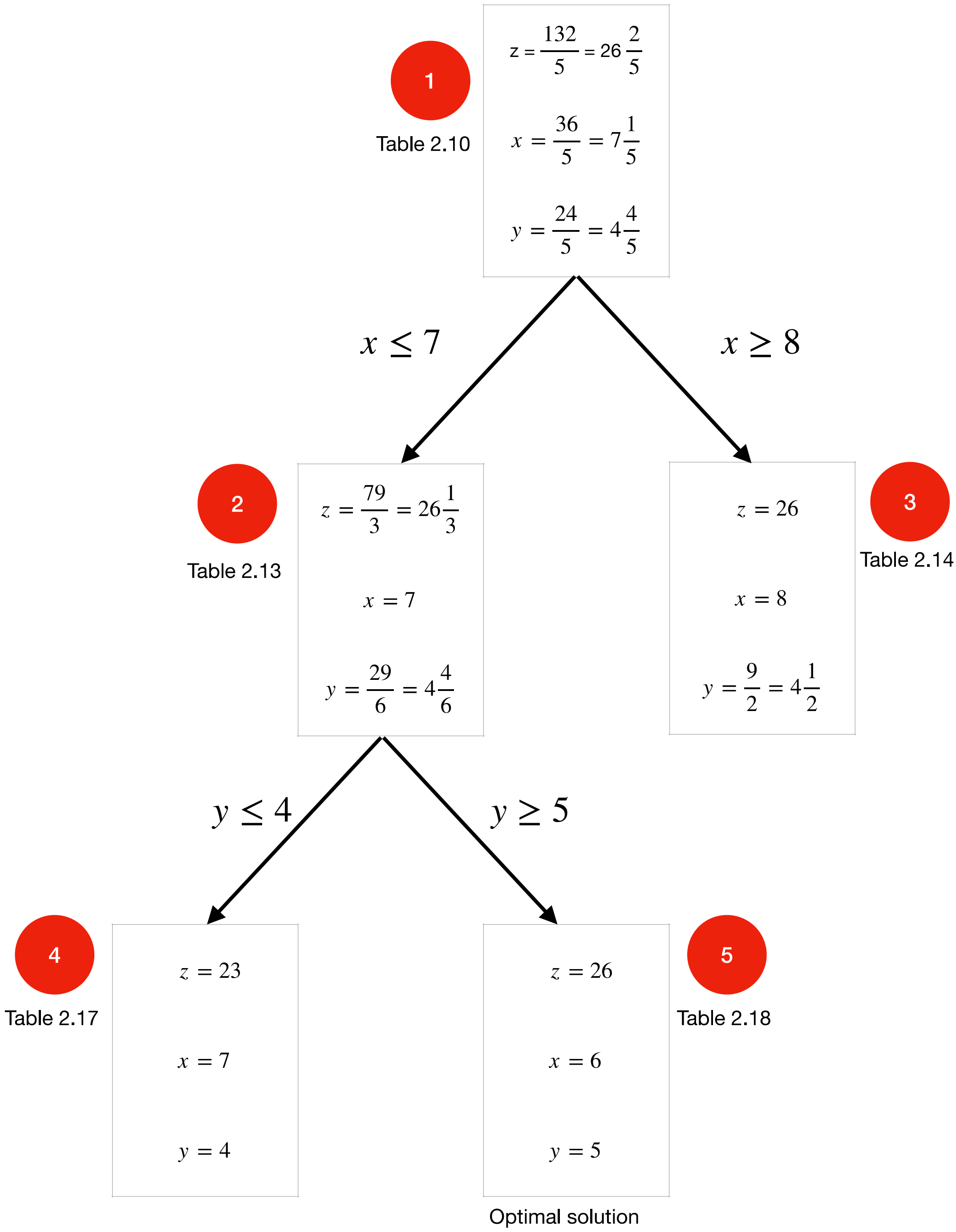


Figure 2.3

Second Branch in the Branch and Bound Tree for Example 2.1.2 Considering x as a Branching Variable.



2.1.4 Software for Integer Programming

The work of [26] mentioned a few software for solving IP and MIP, the two most commercial software mentioned in timetabling literature are:

IBM ILOG CPLEX Optimizer (commonly referred to as CPLEX) can solve integer and mixed programming problems that can run on different platforms. The CPLEX solvers have been used to solve large real-life optimization problems with millions of variables and constraints.

Gurobi Optimizer (the one we used for our timetabling problem) provides solvers for mixed integer solutions of linear, and quadratic programs. It uses advanced implementations of new MIP algorithms using parallel non-traditional search techniques and cutting planes.

Table 2.7

Second Step Tables for Example 2.1.2 Considering y as a Branching Variable.

(a)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	$\frac{2}{5}$	$\frac{1}{5}$	0	$\mathbf{\frac{132}{5}}$
y	0	1	$\frac{3}{10}$	$-\frac{1}{10}$	0	$\mathbf{\frac{24}{5}}$
x	1	0	$-\frac{4}{5}$	$\frac{3}{5}$	0	$\mathbf{\frac{36}{5}}$
u_2	0	0	$\mathbf{-\frac{3}{10}}$	$\mathbf{\frac{1}{10}}$	1	$\mathbf{-\frac{4}{5}}$

(b)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	$\frac{2}{5}$	$\frac{1}{5}$	0	$\mathbf{\frac{132}{5}}$
y	0	1	$\frac{3}{10}$	$-\frac{1}{10}$	-1	$\mathbf{\frac{24}{5}}$
x	1	0	$-\frac{4}{5}$	$\frac{3}{5}$	0	$\mathbf{\frac{36}{5}}$
u_2	0	0	$\mathbf{-\frac{3}{10}}$	$\mathbf{-\frac{1}{10}}$	1	$\mathbf{-\frac{1}{5}}$

(c)

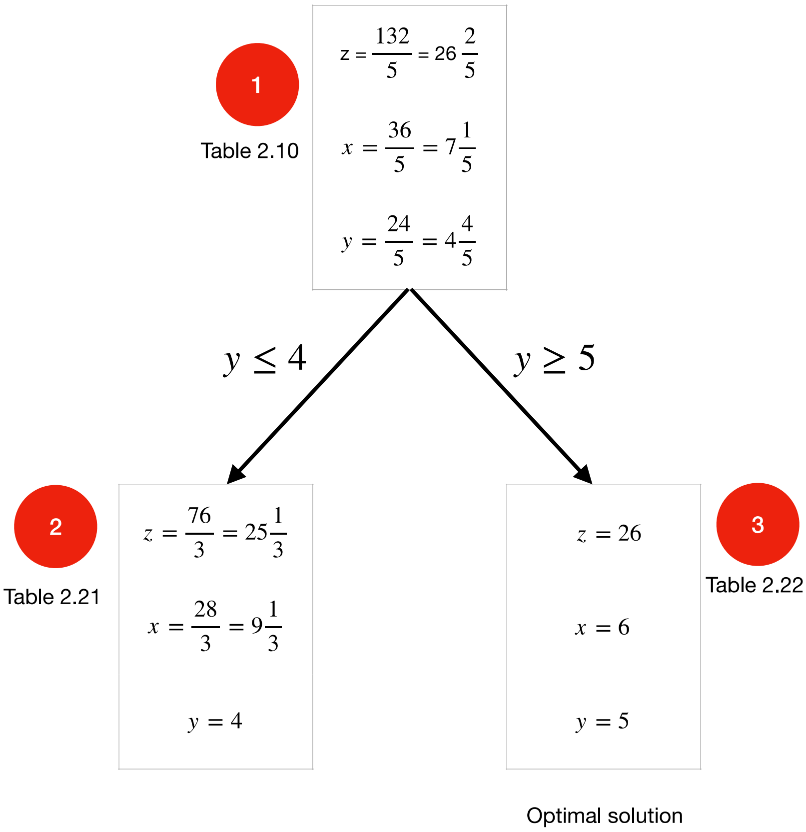
Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	0	$\frac{1}{3}$	$\frac{4}{3}$	$\mathbf{\frac{76}{3}}$
y	0	1	0	0	1	4
x	1	0	0	1	$-\frac{8}{3}$	$\mathbf{\frac{28}{3}}$
u_2	0	0	1	$\mathbf{-\frac{1}{3}}$	$\mathbf{-\frac{10}{3}}$	$\mathbf{\frac{8}{3}}$

(d)

Basic	x	y	u_1	u_2	u_3	Solution
z	0	0	1	0	2	26
y	0	1	0	0	-1	5
x	1	0	1	0	6	6
u_2	0	0	-3	1	-10	2

Figure 2.4

The Branch and Bound Tree for Example 2.1.2 Considering y as a Branching Variable



2.2 IP Formulation

Suppose we have the following sets:

1. $C = \{c_1, c_2, \ldots c_m\}$ be the set of m classes.

2. $T = \{t_1, t_2, \ldots t_n\}$ be the set of n teachers.

3. $P = \{p_1, p_2, \ldots p_l\}$ be the set of l possible periods.

4. $F = \{f_1, f_2, \ldots f_d\}$ be the set of d parents who have more than one student in the school.

And we have the non-negative matrix $R_{m\times n}$ called Requirement's matrix, where r_{ij} is the number of lectures given by teacher t_i to class c_j.

2.2.1 Hard Constrains

The simple mathematical formulation for the timetabling as in [8] is as follows:

Find

x_{ijk} $(i = 1, . . . , m; j = 1, . . . n; k = 1, . . . , l)$

s.t

$\sum_{k=1}^{l} x_{ijk} = r_{ij} \quad (i = 1,\ldots,m;\ j = 1,\ldots n)$ (2.18)

$\sum_{i=1}^{m} x_{ijk} \leq 1 \quad (j = 1,\ldots,n;\ k = 1,\ldots l)$ (2.19)

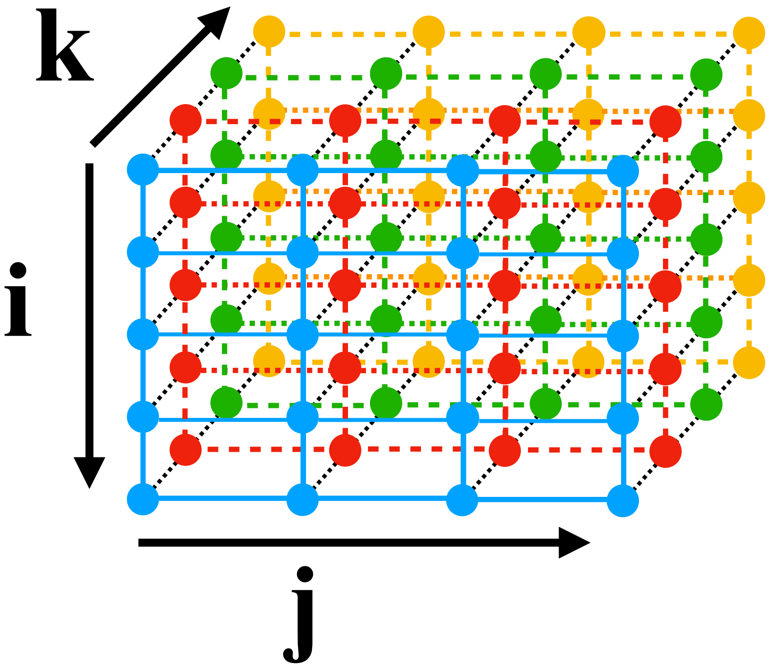
$\sum_{j=1}^{n} x_{ijk} \leq 1 \quad (i = 1,\ldots,m;\ k = 1,\ldots l)$ (2.20)

$x_{ijk} = 0 or \quad 1 \quad (i = 1, . . . , m; j = 1, . . . n; k = 1, . . . , l)$ (2.21)

where $x_{ijk} = 1$ if class c_i and teacher t_j meet at period p_k, and $x_{ijk} = 0$ otherwise. The

Figure 2.5

Matrix Representation of the Timetable.



$m \times n$ constraints (2.18) ensure that each teacher gives right number of lessons to each class. The $n \times l$ constraints (2.19) (resp. $m \times l$ constraints (2.20)) ensure that each teacher (resp. class) is involved in at most one lesson for each period and those are the hard constraints.

Suppose in constraints (2.19) $j = 1$ and $k = 1$ then

$$\sum_{i=1}^{m} x_{i11} = x_{111} + x_{211} + \cdots + x_{m11}$$

This is the summation for the teacher t_1 at period p_1 for all classes, this summation should be equal to 1 or 0. If it is more than 1 this means that t_1 has more than one class at p_1.

Here in Palestine, teachers can teach more than one subject for the same class. Such as teachers who teach grades from 1 to 4. Also, there is a number for how many lessons every teacher should teach. If a teacher’s number of lessons does not equal that number, he/she must teach another subject to reach it.

Since we care about distinguishing the type of subject that some teacher teaches for the same class, we make the teachers columns as teachers-subjects’ columns. So, constrains (2.19) become as follows:

$$\sum_{r\in J_j} \sum_{i=1}^{m} x_{irk} \leq 1 \quad (j = 1,\ldots,n;\ k = 1,\ldots l). \tag{2.22}$$

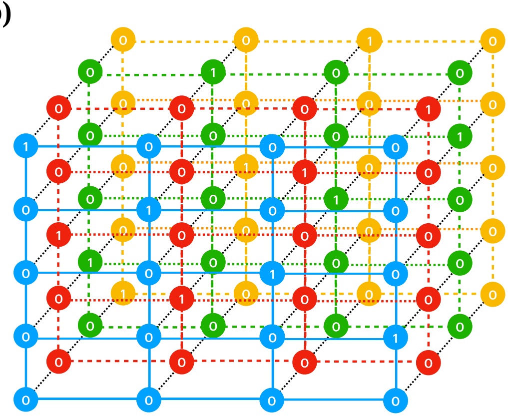
Figure 2.6

Example of Timetabling Matrix Representation.

(a)

Periods	Class #1	Class #2	Class #3	Class #4	Class #5
1	Math		Math	Arabic	Social Science
2	Arabic	Math	English	Math	
3	English	Art	Arabic		Science
4	Art	Arabic		Math	

(b)



where the set J_j is set of the indices of the columns assigned to teacher j.

For example. The matrix representation of the timetable in figure 2.6a is in figure 2.6b. Here in this example, there are 5 classes, 4 periods, and 4 teachers. Every color represents a teacher. As you can see, although some teachers teach different subjects, there is no need for new columns. We use the Teacher-Subject column if they teach the same class several subjects.

Also, here in Palestine there is a possibility of having teachers with half positions i.e., those teachers could be not available at their schools for specific days of the week. According to this and sine the day consists of a set of periods, we added the following constrains:

$$\sum_{l=1+(k-1)c}^{c+(k-1)c} \sum_{r\in J_j} \sum_{i=1}^{m} x_{irl} = 0 \quad \left(j = 1,\ldots, n_{half}\,;\ k \in K_j\right) \tag{2.23}$$

where c is the number of periods per day, n_{half} is the number of teachers with half positions, and K_j is the set of the indices of the days where the half position teacher j is not available.

2.2.3 Soft Constrains

We would like to add soft constraints to the model to make it more efficient; these constrains will minimize the cases where the subject is being taught more than once to any class during a day.

Let us define the following variable:

$$z_{ijk} = \sum_{l=1+(k-1)c}^{c+(k-1)c} x_{ijl}\ (i = 1,\cdots,m; j = 1,\cdots,n; k = 1,\cdots,5)$$

here $z_{ijk} \leq c$, we want to minimize the cases where $z_{ijk} > 1$. To do so, we use the Big M method and introduce an indicator variable u_{ijk} and a new variable Z_{ijk}.

We can formulate it as follows:

$$z_{ijk} = \begin{cases} z_{ijk}, & if\ u_{ijk} = 1 \\ 0, & if\ u_{ijk} = 0 \end{cases} \tag{2.24}$$

with the following inequalities:

$$z_{ijk} \geq 1 - c\,.(1 - u_{ijk}) \tag{2.25}$$

$$z_{ijk} \leq 1 + c\,.u_{ijk} \tag{2.26}$$

Then we want to minimize:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{5} Z_{ijk} \tag{2.27}$$

2.2.4 Minimize the Conflicts in the Main Materials

Let $C_f i$ be the set of the indices of the students' classes for the parent f_i and M is the set of indices of columns of the teachers of the main subjects.

Let us define the following variable:

$$y_{rk} = \sum_{i \in C_{f_r}} \sum_{j \in M} x_{ijk}$$

Here y_{rk} indicates the number of devices the parent f_r needs for his children to attend the lessons of the main subjects in period p_k. Then we want to minimize the cases where y_{rk}>1. i.e., we want to minimize

$$\sum_{r=1}^{d} \sum_{k=1}^{l} (y_{rk} - 1) \quad if\ \ y_{rk} > 1 \tag{2.28}$$

($y_{rk} - 1$ indicates the number of extra devices needed at period p_k for parent f_r). So, as we did before we are going to use the Big M method and introduce an indicator variable e_{rk} and a new variable Y_{rk}.

The formulation would be as follows:

$$Y_{rk} = \begin{cases} y_{rk}, & if\ e_{rk} = 1 \\ 0, & if\ e_{rk} = 0 \end{cases} \quad (2.29)$$

with the following inequalities:

$$y_{\ rk} \geq 1 - BigM\,.(1 - e_{rk}) \quad (2.30)$$

$$y_{rk} \leq 1 + BigM\,.\,e_{rk} \quad (2.31)$$

and (2.30) will be transformed into:

$$\sum_{r=1}^{d}\sum_{k=1}^{l}(Y_{rk} - e_{rk}) \quad (2.32)$$

2.2.5 Objective Function for the Model

Combining (2.27) and (2.32) will give us the objective function which we want to minimize. Moreover, we want to give priority to the number of conflicts in the objective function. Therefore, we added a penalty equal to 0.4 for (2.27) to make the second part less weighted.

$$\sum_{r=1}^{d}\sum_{k=1}^{l}(Y_{rk} - e_{rk}) + 0.4\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{5}Z_{ijk} \quad (2.33)$$

2.2.6 The IP Model

Min

$$\sum_{r=1}^{d}\sum_{k=1}^{l}(Y_{rk} - e_{rk}) + 0.4\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{5}Z_{ijk}$$

Subject to:

$$\sum_{k=1}^{l}x_{ijk} = r_{ij} \qquad (i = 1,\ldots,m;\ j = 1,\ldots n)$$

$$\sum_{r\in J_j}\sum_{i=1}^{m}x_{irk} \leq 1 \qquad (j = 1,\ldots,n;\ k = 1,\ldots l)$$

$$\sum_{j=1}^{n} x_{ijk} \leq 1 \qquad (i = 1, \ldots, m;\ k = 1, \ldots l)$$

$$\sum_{l=1+(k-1)c}^{c+(k-1)c} \sum_{r \in J_j} \sum_{i=1}^{m} x_{irl} = 0 \quad \left(j = 1, \ldots, n_{half}\ ;\ k \in K_j\right)$$

$$z_{ijk} = \sum_{l=1+(k-1)c}^{c+(k-1)c} x_{ijl}\ (i = 1, \cdots, m; j = 1, \cdots, n; k = 1, \cdots, 5)$$

$$z_{ijk} \geq 1 - c\,.\left(1 - u_{ijk}\right)$$

$$z_{\ ijk} \leq 1 + c\,.\,u_{ijk}$$

$$y_{rk} = \sum_{i \in C_{f_r}} \sum_{j \in M} x_{ijk}$$

$$y_{rk} \geq 1 - BigM\,.(1 - e_{rk})$$

$$y_{\ rk} \leq 1 + BigM\,.\ e_{rk}$$

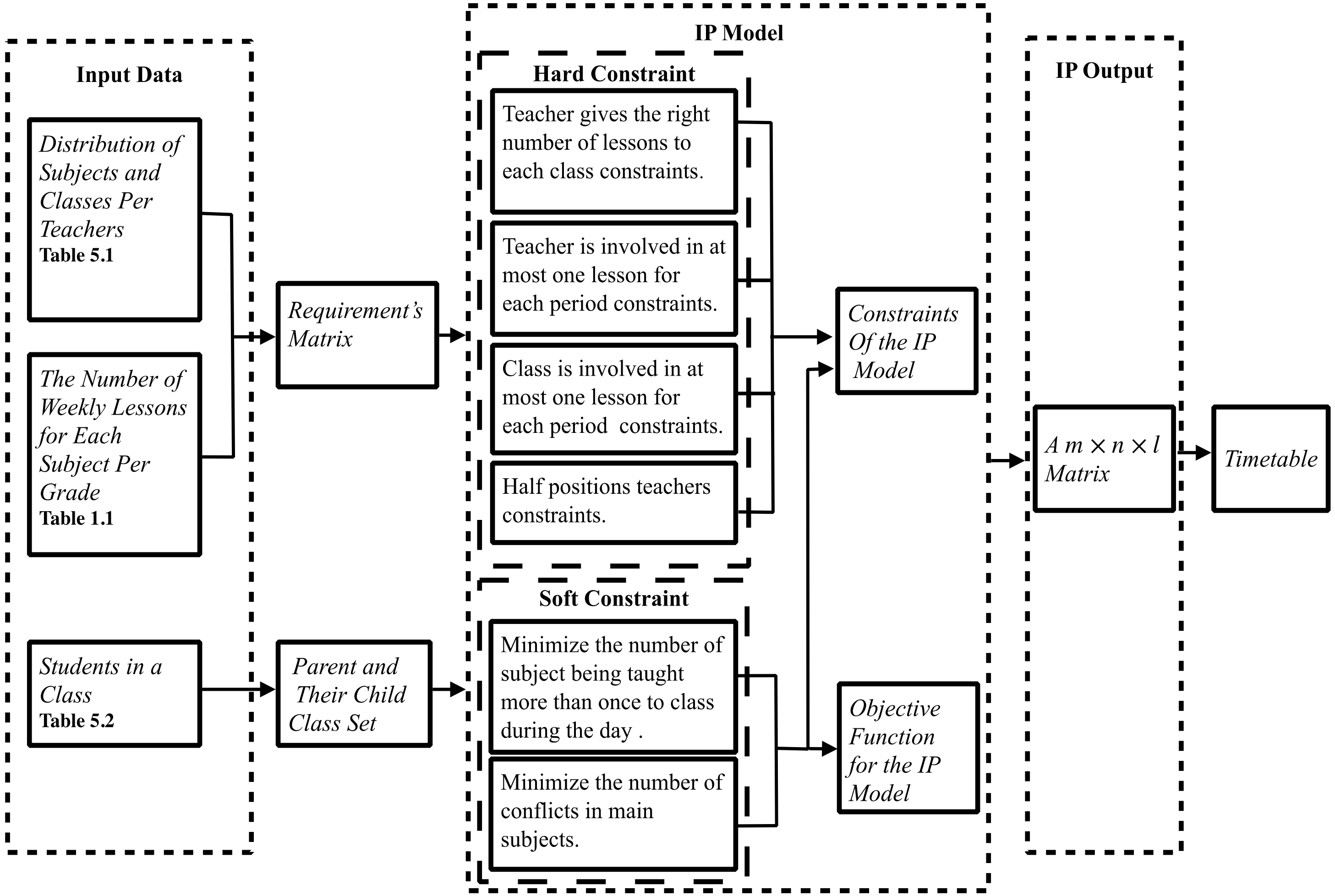
$$x_{ijk} = 0 \quad or\ 1$$

$$z_{ijk} = \begin{cases} z_{ijk}, & if\ u_{ijk} = 1 \\ 0, & if\ u_{ijk} = 0 \end{cases}$$

$$Y_{rk} = \begin{cases} y_{rk}, & if\ e_{rk} = 1 \\ 0, & if\ e_{rk} = 0 \end{cases}$$

Figure 2.7

Flow Chart of the IP Model



2.3 Case Study

In this section, we are going to illustrate all the study using a complete example. We will explain how the constraints and the equations in the previous section work to come out with an optimal schedule.

To illustrate the whole steps of the work and how they are applied, we begin with Table 2.8a that contains a list subject for 3 different classes where each number in the table indicates the number of sessions/weeks. For example, *Arabic/ClassA* = 4 means that the Arabic subject for Class A will have 4 sessions/week. The same table, part b, depicts the same subjects and the names of teachers for each class. For example, the teacher Jihan that teaches Arabic for Class A will be assigned 4 sessions per week to teach this subject. However, part C of the same table has 3 different families: Family 1, 2 and 3 with $*$ sign to indicate that the family has a child in the given class. For example, Family 1 has 2 children in Classed A and B.

Now, after illustrating the list of data that will be input for the algorithm, we can model them in the fallowing sets:

1. $C = \{A, B, C\}$ be the set of the 3 classes.

2. $T = \{Jihan, Fatima, Eman, Maysa, Aman, Ruba, Safa, Nada, Ghson, Alaa, Maha, Salam\}$ be the set of the 12 teachers.

3. $P = \{p_1, p_2, \ldots p_{25}\}$ be the set of 25 possible periods (5 days and 5 periods per a day).

4. $F = \{f_1, f_2, f_3\}$ be the set of 3 parents whose children's classes are shown in Table 2.8c.

Table 2.8d represents a 3 × 13 matrix, the column and the row header are only for explanation purpose, which is the requirements matrix mentioned in the previous section. This matrix contains the initial data taken from the Table 2.8a and 2.8b. For example, the teacher Eman in Table 2.8b teaches Class A and Class B, and she does not teach Class C. Then, under this teacher column, there is the number 3 in the row of Classes A and B which is taken from Table 2.8a. However, number 0 in Class C means that this teacher does not teach this class. Another example, the teacher Jihan teaches Class A two subjects: the Religious Studies and Arabic. As we mentioned earlier in the previous section, we need to distinguish between those subjects because one of them is a main subject and the other is not. That is why we have put them into two columns, and that is why the matrix dimension is 3 × 13 not 3 × 12. This is indicated by Jihan-R and Jihan-A to reflect that she teaches Religious Studies and Arabic subjects.

Table 2.8

The Input for the Example

(a) *The Number of Lessons for Each Subject and Class* **(b)** *The Name of the Teacher for Each Subject and Class*

Subject	Class A	Class B	Class C	Subject	Class A	Class B	Class C
Religious Studies	2	2	2	Religious Studies	Jihan	Ghson	Ghson
Arabic	4	4	4	Arabic	Jihan	Jihan	Jihan
English	2	2	2	English	Fatima	Fatima	Fatima
Mathematics	3	3	3	Mathematics	Eman	Eman	Alaa
Science	2	2	2	Science	Maysa	Maysa	Maha
Social Sciences	2	2	2	Social Sciences	Aman	Aman	Aman
Art	1	1	1	Art	Ruba	Ruba	Ruba
Sport	1	1	1	Sport	Safa	Safa	Safa
Technology	**1**	**1**	**1**	**Technology**	**Nada**	**Nada**	Salam

(c) The Set of The Parents and Their Children Class

Families	**Class** A	**Class** B	**Class** C
Family 1	*	*	
Family 2	*		*
Family 3		*	*

(d) The Summarizing of Data Appear in Tables 2.8a and 2.8b.

	Jihan-R	Jihan-A	Fatima	Eman	Maysa	Aman	Ruba	Safa	Nada	Ghson	Alaa	Maha	Salam
Class A	2	4	2	3	2	2	1	1	1	0	0	0	0
Class B	0	4	2	3	2	2	1	1	1	2	0	0	0
Class C	0	4	2	0	0	2	1	1	0	2	3	2	1

To find the solution for this example, we need to find the values of the 975 binary numbers, which are the entries for the matrix ***X*** whose dimension is 3 × 13 × 25.

2.3.1 Hard Constraints

Now, Equation 2.18 will be applied on the collected 39 constrains (3 × 13) as follows:

1. $\sum_{k=1}^{25} x_{1,1,k} = x_{1,1,1} + x_{1,1,2} + \cdots + x_{1,1,25} = r_{1,1} = 2$

2. $\sum_{k=1}^{25} x_{1,2,k} = x_{1,2,1} + x_{1,2,2} + \cdots + x_{1,2,25} = r_{1,2} = 4$

39. $\sum_{k=1}^{25} x_{3,13,k} = x_{3,13,1} + x_{3,13,2} + \ldots + x_{3,13,25} = r_{3,13} = 1$

Also, the 325 constraints after applying equation 2.19 would look like this:

1. $\sum_{i=1}^{3} x_{i,1,1} = x_{1,1,1} + x_{2,1,1} + x_{3,1,1} \leq 1$

2. $\sum_{i=1}^{3} x_{i,1,2} = x_{1,1,2} + x_{2,1,2} + x_{3,1,2} \leq 1$

325. $\sum_{i=1}^{3} x_{i,13,25} = x_{1,13,25} + x_{2,13,25} + x_{3,13,25} \leq 1$

But since column 1 and 2 in the requirements matrix are for the same teacher, and we do not want this teacher to have more than one lesson in every period. The constraints for this teacher after applying equation 2.22 with the set $J_1 = \{1, 2\}$ will be:

1. $\sum_{r \in J_1} \sum_{i=1}^{3} x_{i,\mathrm{r},1} = \sum_{i=1}^{3} x_{i,1,1} + \sum_{i=1}^{3} x_{i,2,1} \leq 1$

2. $\sum_{r \in J_1} \sum_{i=1}^{3} x_{i,\mathrm{r},2} = \sum_{i=1}^{3} xi,1,2 + \sum_{i=1}^{3} x_{i,2,2} \leq 1$

25. $\sum_{r \in J_1} \sum_{i=1}^{3} x_{i,\mathrm{r},25} = \sum_{i=1}^{3} x_{i,1,25} + \sum_{i=1}^{3} x_{i,2,25} \leq 1$

This reduces the number of constraints to 300 (25×12) instead of 325 (25×13) constraints after merging the first two columns since they belong to the same teacher.

The following 75 constraints are generated after applying equation 2.20:

1. $\sum_{j=1}^{13} x_{1,j,1} = x_{1,1,1} + x_{1,2,1} + \ldots + x_{1,13,1} \leq 1$

2. $\sum_{j=1}^{13} x_{1,j,2} = x_{1,1,2} + x_{1,2,2} + \ldots + x_{1,13,2} \leq 1$

75. $\sum_{j=1}^{13} x_{1,j,25} = x_{3,1,25} + x_{3,2,25} + \ldots + x_{3,13,25} \leq 1$

Finally, constraints related to equation 2.23 which are for the set of teachers who with half position situations. Let us suppose the teachers in half position are Ruba, Salam and Maysa, and their working days as follows:

1. Ruba and Salam: Sunday, Tuesday and Thursday.

2. Maysa: Monday and Wednesday.

If we give the days of the week the indices 1,2,3,4 and 5, starting from Sunday up to Thursday, then the set of the indices for the days that they are not working in the pretended school for Ruba and Salam is {2, 4} and for Maysa is {1, 3, 5}. Ruba has 7 as index column in the requirements matrix, so the constraints after applying equation 2.23 for this teacher will be like this:

For $k = 2$:

$$\sum_{l=6}^{10} \sum_{i=1}^{3} x_{i,7,l} =$$

$$\sum_{l=6}^{10} x_{1,7,l} + x_{2,7,l} + x_{3,7,l} =$$

$$x_{1,7,6} + x_{2,7,6} + x_{3,7,6} + x_{1,7,7} + x_{2,7,7} + x_{3,7,7} + . + x_{1,7,10} + x_{2,7,10} + x_{3,7,10} = 0$$

For $k = 4$:

$$\sum_{l=16}^{20} \sum_{i=1}^{3} x_{i,7,l} =$$

$$\sum_{l=16}^{20} x1,7,l + x2,7,l + x3,7,l =$$

$$x_{1,7,16} + x_{2,7,16} + x_{3,7,16} + x_{1,7,17} + x_{2,7,17} + x_{3,7,17} + . + x_{1,7,20} + x_{2,7,20} + x_{3,7,20} = 0$$

Ruba has only one column so there is no need for the second summation, and we put $c = 5$ since we suppose that there are 5 periods per the day. It is the same for Salam except the column index is equal to 13 instead of 7.

For Maysa, whose index column is 5, we need three constrains:

$k = 1$

$$\sum_{l=1}^{5} \sum_{i=1}^{3} x_{i,5,l} = 0$$

$k = 3$

$$\sum_{l=11}^{15} \sum_{i=1}^{3} x_{i,5,l} = 0$$

$k = 5$

$$\sum_{l=21}^{25} \sum_{i=1}^{3} x_{i,5,l} = 0$$

2.3.2 Results

Now, we want to find the solution that optimize the objective function appears in 2.36. We get 4 feasible solutions after running the algorithm with the following results: Data in the Table 1 in Appendix B indicates two optimal tables with the minimal number of conflicts. The two 8 numbers appear in the table indicates that we have two identical solutions equal in their objective values and they different in terms of sessions distributions/5 days. Both timetables can be used as a solution, where each one of them has the same minimal number of conflicts.

Tables 2a and 3a in Appendix B represent the generated schedules for the 5 days for all both subjects and teachers with their corresponding sessions per class per subject. However, we denote the conflict subjects by bold in which each two conflicted subjects appear in the same row. Of course, these conflicts are per family, i.e., each conflicted subjects are related to one family who has 2 or more children. This conflict can be solved by increasing the number of computing machines for that family.

Chapter Three

Experimental Tests

In this chapter, we are conducting an experimental evaluation for the proposed model to test its accuracy in mapping studying sessions to teachers and minimizing the number of conflicts that might exist. Gurobi 9.1.1 1[3] (Optimization Software) is used as MIP solver with default settings on an Intel Core i5 2.5 GHz Dual-Core machine with 8 GB of RAM under a 64-mac operating system. The algorithm is implemented using Jupyter notebook 6.1.4 with Python 3.8.11. The conducted experimental tests are done with virtual and real data.

3.1 Virtual Data: Conflicts Comparisons

To estimate the usefulness of our work, we conducted an experimental test to compare between the manual and automated construction of lessons schedule for online learning in terms of final number of conflicts after generating a feasible schedule. To do so, we generated random data for 2 schools, each of which contains 5 classes (from grade 5 to 9) and a set of 8 teachers as well as the list of elementary learning subjects. To make the experimental test more real, we fixed up a set of families with more than one child (2 children) to simulate a real-life example with possible conflicts within the same family. We repeated the test 3 times with different number of children per family. We manually prepare the schedules and then we used the code to generate the schedules for the same data.

Table 3.1 represents the amounts of conflicts for constructing the schedule manually and by the code. We have two cases: *Case 1* represents a mix of teachers in which part of the are teaching *Arabic* subject, part of them teaches two subjects *Physical Education* and *Arts*, and finally part of them teaches in two different schools. *Case 2* is like *Case 1* but with different permutations of teachers. All of this is to make the simulated experiments more realistic.

After preparing the suitable data for the test, we manually worked on preparing the weekly schedule for each class and computed the number of conflicts.

[3] https://support.gurobi.com/hc/en-us

We need to mention that we computed the number of conflicts in a separate code. There were two codes to compute the number of conflicts: one computed the conflicts from the matrix obtained from Gurobi and another from the timetable. We made this to make our work accurate. Also, we used the second one to compute the number of conflicts for the real timetable.

Table 3.1

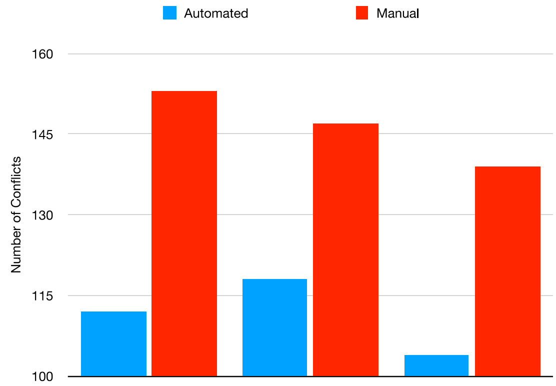
Number of Conflicts for the Schedule Construction: The Manual and by the Code.

	Manual	Code
	112	153
case1	118	147
	104	139
	104	143
case2	115	154
	101	142

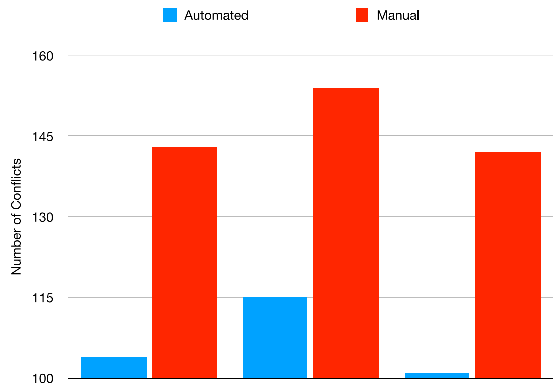
Figure 3.1

Manual vs Automated Schedule Conflicts

(a) Case1



(b) Case2



It is clear from the data in the previous table that using the code leads to a smaller number of conflicts. Figure 3.1 reflects the comparison between the manual and the code schedule construction number of conflicts.

Figure 3.2

The Graph for Time Analysis.

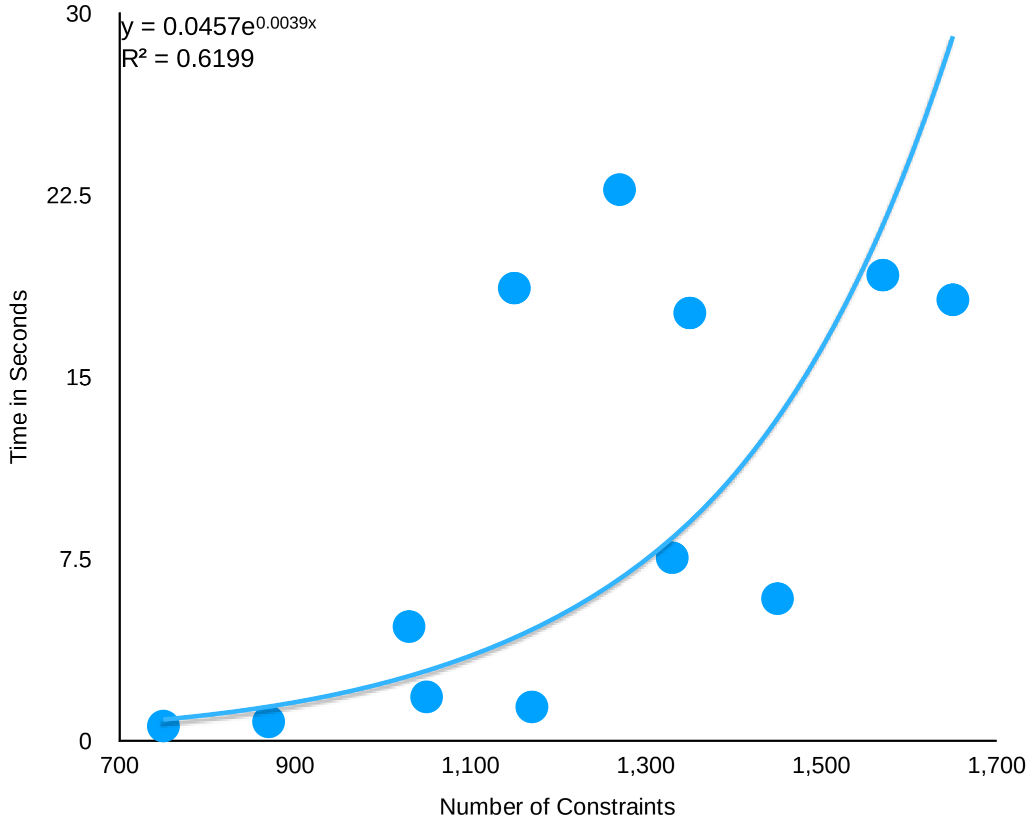


Figure 3.2 shows the graph for time analysis, and we fitted the data to an exponential model [26].

3.2 Real Data Set

Working on preparing lessons schedule is both time and effort consuming. So, making a comprehensive experimental test that includes big amount of manual schedule construction to be compared with the code one is not an easy and feasible task. Therefore, we executed the first part of testing that relied on virtual data. Just to introduce the reader with the importance of code-based construction of the schedule. However, in this part of the test we relied on real data collected from several schools to be used in this part of experimental test.

3.2.1 Large Data Set

The data here is collected from the north area of Nablus city, specifically from four schools. Table 3.2a shows how many classes and teachers, and which grades each school has.

Table 3.2b shows the characteristic for the whole data. Here:

1. m refers to the number of classes.
2. N to the number of teachers.
3. n_{half} to the number of the half position teachers.
4. n to the number of teachers columns of the requirements matrix.
5. c to the number of periods per day.
6. l to the number of periods per week.
7. d refers to the size of the parents set.

This parent's set has been taken from student data Table 3 in Appendix A. Moreover, in the academic year 2020-2021, the ministry of education applied blended learning, as we mentioned earlier. In this year, the ministry spread the lessons into two weeks rather than one week. Approximately 50% of lessons were taught throw the week. So, to simulate this situation, we prefer to set the percentage of the number of lessons in Table 1 in Appendix A to 60%.

Table 3.2

(a) *Tested Data.*

The name of the school	Grades	The number of teachers	The number of classes
Burqa Basic School for Girls	1 - 4	8	6
Burqa Basic School for Boys	1- 6	17	11
Burqa Secondary School for Girls	5-12	22	13
Burqa Secondary School for Boys	7-12	20	11
Total		67	41

(b) *Main Characteristic of the Tested Data.*

m	N	n_{half}	n	c	l	d
41	67	12	129	5	25	551

We experiment two cases:

1. The case as it is, i.e., Timetable for the four schools applying the hard constraints and the soft constraints for the whole classes and equation 2.30 for grades from 1st to 9th grades.

2. If we pretend that we know the number of the devices in each family or at least some families have two devices, we would like to see the effect of this knowledge on the objective function.

Table 1 in Appendix C shows the results for the two cases above. For case 2, we ran the code three times. The average of the number of conflicts for the three runs equals to 922.6. From case 2, if we know that there are some families that have more than one device, we could reduce the number of conflicts by 33.8%.

Here, the optimality gap found by Gurobi g considers the cost of the best solution found b (Objective Value) and the best objective bound l (Objective Bound). It is computed as $\frac{|b-l|}{|b|}$. If $b = 0$ then there is no solution and then $g = \infty$. If the optimal solution found, $b = l$, then $g = 0$. i.e., as the solution gets closer to the optimal solution, the gap gets closer to zero.

3.2.2 Real Timetable and the Automated One

At the beginning, we wanted to make several conflicts comparison between the four real timetables of the participated schools during the lockdown and the ones generated by the code. However, this comparison would be illogical since the lesson periods (teaching sessions) in each school have different time slots. Instead of that, we considered one of the four schools to execute the comparison (Burqa Secondary School for Girls).

Here is how this comparison is done: First, we took the real timetable for this school which is Table 3 in Appendix C. Then, we computed the number of conflicts for this timetable considering only grades from 5 to 9. Table 2 in Appendix C shows that the real one has 376 conflicts, approximately 4.37 conflicts for each family as an average. Second, we made a requirements matrix from this timetable, then we applied the algorithm (IP) on the data for this school using the requirements matrix we got from the real one. Table 4 in Appendix A shows the resulted timetable. Table 2 in Appendix C displays the number of conflicts for this timetable, which is 224, approximately 2.6 conflicts for each family as an average. Figure 1 in Appendix D depicts the degradation in the objective value during time. The figure illustrates how the number of conflicts decreases dramatically within the first time slots then after the time 200 seconds the change gets stable.

Figure 3.4 shows a sample output of running the Gurobi[4] code that describes the output as follows: The MIP log can be divided into three sections: the presolve section, the simplex progress section, and the summary section.

- Presolve Section:

Here, presolve removes 13873 rows and 26549 columns. The Presolved line shows the size of the model that is passed to the branch-and-cut algorithm and Variable types line the types of remaining variables.

- Progress Section:

In this section the MIP log tracks the progress of the branch-and-cut search, which involves several different steps.

[4] https://www.gurobi.com/documentation/9.1/refman/mip_logging.html

First, the line Found heuristic solution indicates that the Gurobi heuristics found an integer-feasible solution before the root relaxation was solved. Next, the Root relaxation line displays the solution for this step. The final section in the progress section provides information on the branch-and-cut tree search. This section has four columns: Nodes, Current Node, Objective Bounds, and Work. You can refer to the website for more details about each column. One thing we need to mention here is the letter H. This letter means that Gurobi found a new feasible solution by MIP heuristic.

- Summary Section

In this section, the summary information can be found when the MIP solver is finished, or the time limit is reached. Here, we set the time to 30 minutes (1800 seconds). The last line provides information about the best objective found, the best bound, and the gap.

Figure 3.3

Gurobi Output

```
Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (mac64)
Thread count: 2 physical cores, 4 logical processors, using up to 4 threads
Optimize a model with 26938 rows, 38125 columns and 99340 nonzeros
Model fingerprint: 0xd9fd4c71
Model has 6560 general constraints
Variable types: 0 continuous, 38125 integer (28000 binary)
Coefficient statistics:
  Matrix range     [1e+00, 6e+00]
  Objective range  [4e-01, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 6e+00]
Presolve removed 13873 rows and 26549 columns
Presolve time: 0.25s
Presolved: 13065 rows, 11576 columns, 38103 nonzeros
Variable types: 0 continuous, 11576 integer (8419 binary)
Found heuristic solution: objective 423.4000000

Root relaxation: objective 1.085000e+02, 16123 iterations, 2.78 seconds
Total elapsed time = 6.06s

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  108.50000    0 1195  423.40000  108.50000  74.4%     -    7s
H    0     0                    420.6000000  108.50000  74.2%     -    7s
     0     0  108.50000    0 1678  420.60000  108.50000  74.2%     -    9s
     0     0  108.50000    0 1645  420.60000  108.50000  74.2%     -    9s
     0     0  109.10000    0 1302  420.60000  109.10000  74.1%     -   18s
     0     0  109.10000    0 1483  420.60000  109.10000  74.1%     -   24s


 26618 17042     cutoff   90       226.40000  205.72545  9.13%   304 1589s
 26811 17142  207.33234   36 1894  226.40000  205.74080  9.13%   305 1600s

 30105 18754  209.55247   47 1781  226.40000  205.93493  9.04%   317 1774s
 30243 18940  220.71994   69 1424  226.40000  205.94398  9.04%   318 1785s
 30510 18998  218.26651   67 1386  226.40000  205.95567  9.03%   319 1794s
 30704 19053  207.30069   41 1782  226.40000  205.96191  9.03%   319 1800s

Cutting planes:
  Gomory: 261
  Cover: 9
  Implied bound: 151
  MIR: 92
  StrongCG: 3
  Flow cover: 206
  Zero half: 519
  Relax-and-lift: 8

Explored 30846 nodes (10028661 simplex iterations) in 1800.03 seconds
Thread count was 4 (of 4 available processors)

Solution count 10: 226.4 227.4 228.4 ... 237.4

Time limit reached
Best objective 2.264000000000e+02, best bound 2.060000000000e+02, gap 9.0106%
```

Chapter Four

Conclusion

In this thesis, we presented a formulation of the school timetabling problem. We designed this problem to handle situations like the ones the Palestinian schools experienced during the COVID-19 pandemic. During the COVID-19 pandemic and the academic year 2020- 2021, the schools adopted the e-learning strategy. Our formulation tries to help in solving one of the e-learning problems, which is the conflicts of lessons sessions due to the lack of students' devices. We formulate an IP problem with minimum hard constraints for this timetable and an objective function that can reduce the conflicts in the main subjects for grades from 1 to 9. Moreover, we tested the method using Gurobi as an IP solver on virtual and real data, and we had a good result. For the virtual data, the IP algorithm managed to reduce the number of conflicts by 26%. Also, for the real data, the IP algorithm decreased the number of conflicts by 40%.

List of Abbreviations

Abbreviation	Meaning
B&B	Branch and Bound
IP	Integer Programming
LP	Linear Programming
MIP	Mixed Integer Programming

References

[1] Dhawan S. Online Learning: A Panacea in the Time of COVID-19 Crisis. Journal of Educational Technology Systems. 2020;49(1):5-22. Available from: https://doi.org/10.1177/0047239520934018.

[2] Lassoued Z, Alhendawi M, Bashitialshaaer R. An Exploratory Study of the Obstacles for Achieving Quality in Distance Learning during the COVID-19 Pandemic. Education Sciences. 2020;10(9). Available from: https://www.mdpi.com/22277102/10/9/232.

[3] Castano AM, Hernandez J, Llanos AM. Kids Today: Remote Education in the timeof COVID-19. ArXiv. 2020;abs/2010.07295.

[4] Fojtik R. Problems of Distance Education. International Journal of Information andCommunication Technologies in Education. 2018 05;7:14-23.

[5] Wren A. Scheduling, timetabling and rostering — A special relationship? In: Burke E, Ross P, editors. Practice and Theory of Automated Timetabling. Berlin, Heidelberg: Springer Berlin Heidelberg; 1996. p. 46-75.

[6] Burke E, de Werra D, Kingston J. Applications to timetabling. Handbook of Graph Theory. 2004:445-74. Available from: http://infoscience.epfl.ch/record/88640.

[7] Kristiansen S, Stidsen T. A Comprehensive Study of Educational Timetabling - a Survey. No. 8.2013 in DTU Management Engineering Report. DTU Management Engineering; 2013.

[8] Schaerf A. A Survey of Automated Timetabling. Artif Intell Rev. 1999 Apr;13(2):87–127.

[9] Tan JS, Goh SL, Kendall G, Sabar NR. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Systems with Applications. 2021;165:113943.

[10] Pillay N. A survey of school timetabling research. Annals of Operations Research. 2013 02;218.

[11] Burke EK, Kendall G. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. 2nd ed. Springer Publishing Company, Incorporated; 2013.

[12] Lawrie NL. An integer linear programming model of a school timetabling problem.The Computer Journal. 1969 01;12(4):307-16. Available from: https://doi.org/10.1093/comjnl/12.4.307.

[13] Birbas T, Daskalaki S, Housos E. Timetabling for Greek High Schools. The Journal of the Operational Research Society. 1997 2021/09/28/;48(12):1191-200. Full publication date: Dec., 1997. Available from: https://doi.org/10.2307/3010749.

[14] Birbas T, Daskalaki S, Housos E. School timetabling for quality student and teacherschedules. J Scheduling. 2009 04;12:177-97.

[15] Santos H, Uchoa E, Ochi L, Maculan N. Strong bounds with cut and column genera-tion for class-teacher timetabling. Annals of Operations Research. 2012 04;194:399-412.

[16] Ribic´ S, Konjicija S. A two phase integer linear programming approach to solvingthe school timetable problem. In: Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces; 2010. p. 651-6.

[17] Papoutsis K, Valouxis C, Housos E. A column generation approach for the timetabling problem of Greek high schools. Journal of The Operational Research Society - J OPER RES SOC. 2003 03;54:230-8.

[18] Kristiansen S, Sørensen M, Stidsen T. Integer programming for the generalized highschool timetabling problem. Journal of Scheduling. 2015 08;18.

[19] Post G, Kingston J, Ahmadi S, Daskalaki S, Gogos C, Kyngas J, et al. XHSTT: AnXML archive for high school timetabling problems in different countries. Annals ofOperations Research. 2011 07;218:295-301.

[20] Árton P Dorneles, de Araújo OCB, Buriol LS. A column generation approach to high school timetabling modeled as a multicommodity flow problem. European Journal of Operational Research. 2017;256(3):685-95. Available from: https://www.

sciencedirect.com/science/article/pii/S0377221716305392.

[21] Tassopoulos IX, Iliopoulou CA, Beligiannis GN. Solving the Greek school timetabling problem by a mixed integer programming model. Journal of the Operational Research Society. 2020;71(1):117-32. Available from: https://doi.org/10.1080/01605682.2018.1557022.

[22] Even S, Itai A, Shamir A. On the complexity of timetable and multi-commodity flow problems. In: 16th Annual Symposium on Foundations of Computer Science (sfcs 1975); 1975. p. 184-93.

[23] Post G, Ahmadi S, Daskalaki S, Kingston J, Kyngas J, Nurmi K, et al. An XML format for benchmarks in High School Timetabling. Annals of Operations Research.2012 04;194:385-97.

[24] A´ rton P Dorneles, de Arau´jo OCB, Buriol LS. A fix-and-optimize heuristic for the high school timetabling problem. Computers and Operations Re- search. 2014;52:29-38. Available from: https://www.sciencedirect.com/science/article/pii/S0305054814001816.

[25] Fonseca GHG, Santos HG, Carrano EG. Integrating matheuristics and metaheuristics for timetabling. Computers and Operations Research. 2016;74:108-17. Available from:https://www.sciencedirect.com/science/article/pii/S0305054816300879.

[26] Carter MW, Price CC, Rabadi G. Operations Research: A Practical Introduction. Second edition ed. Chapman and Hall/CRC.; 2019.

[27] Taha HA. Operations Research: An Introduction (8th Edition). USA: Prentice-Hall,Inc.; 2006.

[28] Kolman B, Beck RE. Elementary Linear Programming with Applications (Second Edition). Second edition ed. San Diego: Academic Press; 1995.

Appendices

Appendix A

Data

Table A-1

A Sample of the Number of Weekly Lessons for Each Subject Per Grade.

Subject	1st Grade	2nd Grade	3rd Grade	4th Grade	5th Grade	6th Grade	7th Grade	8th Grade	9th Grade
Religious studies	3	3	3	3	4	4	4	4	4
Arabic	10	10	9	9	7	7	7	6	6
English	3	3	3	3	4	4	4	5	5
Mathematics	6	6	6	6	5	5	5	5	5
Science	-	-	3	3	4	4	5	5	5
civic education	5	5	3	3	-	-	-	-	-
Social sciences	-	-	-	-	4	4	3	3	3
Art	-	-	-	-	2	2	1	1	1
Sport	1	1	1	1	2	2	1	1	1
Technology	-	-	-	-	2	2	2	2	2
Elective Subjects	-	-	-	-	-	-	2	2	2
Summation	28	28	28	28	34	34	34	34	34

Table A-2

Sample Distribution of Subjects and Classes Per Teachers

Class	R-Studies	Arabic	English	Math	Science	S-Sciences	Art	Sport	Tech
5a	Jihan	Jihan	Fatima	Eman	Maysaa	Aman	Ruba	Safa	Nada
5b	Ghsoon	Jihan	Fatima	Eman	Maysaa	Aman	Ruba	Safa	Nada
6	Ghsoon	Jihan	Fatima	Alaa	Maha	Aman	Ruba	Safa	Salam

Table A-3

A Sample Students in a Class.

Student Name	Student ID Number	Gender	Date Birth	ofParentName	Parent ID Number	Relation	Place of Birth
Eqbal Ahmed	427......	Female	DD 2010	MMAhmed	992......	Father	Nablus
Alaa Shaker	427......	Female	DD 2010	MMShaker	941......	Father	Nablus
Basma Emad	427......	Female	DD 2010	MMEmad	850......	Father	Nablus
Tala Mohamed	431......	Female	DD 2011	MMMohamed	942......	Father	Nablus
Toqa Waheed	427......	Female	DD 2010	MMWaheed	959......	Father	Nablus

Appendix B

Case Study

Table B-1

Tables' Titles with Their Corresponding Objective Values.

Table Number	Objective Function Value.
Table1	10
Table2	9.8
Table3	8
Table4	8

Table B-2

The First Optimal Generated Sessions Schedules.

(a) *Subjects schedule for the 5 days.*

The day	Class A	Class B	Class C
Sun	Art	**Arabic**	**English**
	Religion	Math	Tec
	Arabic	Tec	Religion
		English	**Arabic**
	Social	Sport	Math
Mon	**Science**	**Math**	
		Science	**English**
	English		**Math**
			Science
	Arabic	Religion	
Tue	Math	Social	Religion
	Social	Art	Science
	English		**Arabic**
		Arabic	Sport
	Religion	Math	Art
Wed	**Arabic**	**Science**	Social
		Arabic	
			Arabic
	Math		
	Science		
Thu	**Arabic**	Social	**Math**
		English	
	Tec		Arabic
	Sport	Arabic	
	Math	Religion	Social

(b) *Teachers schedule for the 5 days.*

The day	Class A	Class B	Class C
Sun	Ruba	Jihan	Fatima
	Jihan	Eman	Salam
	Jihan	Nada	Ghson
		Fatima	Jihan
	Aman	Safa	Alaa
Mon	Maysa	Alaa	
		Maysa	Fatima
	Fatima		Alaa
			Maha
	Jihan	Ghson	
Tue	Eman	Aman	Ghson
	Aman	Ruba	Maha
	Fatima		Jihan
		Jihan	Safa
	Jihan	Eman	Ruba
Wed	Jihan	Maysa	Aman
		Jihan	
			Jihan
	Eman		
	Maysa		
Thu	Jihan	Aman	Alaa
		Fatima	
	Nada		Jihan
	Safa	Jihan	
	Eman	Ghson	Aman

Table B-3

The Second Optimal Generated Sessions Schedules.

(a) *Subjects schedule for the 5 days.*

The day	Class A	Class B	Class C
Sun	Art	Religion	Math
	Math		**Arabic**
	Religion	English	Social
			Arabic
		Arabic	**English**
Mon	**Science**	**Math**	
	English	Tec	
	Religion	Science	
			Math
	Arabic		
Tue	Arabic	Art	Tec
	Math		**Math**
		Math	Religion
		Sport	Arabic
		Arabic	Social
Wed	Tec	Arabic	Sport
	Sport	**English**	**Arabic Science**
	Science		
	Arabic	Social	
	Social	Science	
Thu	Social	Math	Religion
	Arabic		**English**
		Arabic	
	English	Religion	Art
	Math	Social	**Science**

(b) *Teachers schedule for the 5 days.*

The day	Class A	Class B	Class C
Sun	Ruba	Ghson	Alaa
	Eman		Jihan
	Jihan	Fatima	Aman
			Jihan
		Jihan	Fatima
Mon	Maysa	Eman	
	Fatima	Nada	
	Jihan	Maysa	
			Alaa
	Jihan		
Tue	Jihan	Ruba	Salam
	Eman		Alaa
		Eman	Ghson
		Safa	Jihan
		Jihan	Aman
Wed	Nada	Jihan	Safa
	Safa	Fatima	Jihan
	Maysa		Maha
	Jihan	Aman	
	Aman	Maysa	
Thu	Aman	Eman	Ghson
	Jihan		Fatima
		Jihan	
	Fatima	Ghson	Ruba
	Eman	Aman	Maha

Appendix C

Results

Table C-1

The Results of the Two Cases Using a Time Limit of 12 Hours.

Case	Objective Value	Objective Bound	Gap	Number of Conflict
1	1403.4	1117.8	20.4%	1395
	985.8	759.1	23.0%	977
2	949.6	729.2	23.2%	940
	859.8	720.8	16.2%	851

Table C-2

Comparison Results Between the Real Timetable and the Timetable Obtained After Implementing the IP.

Case	Number of Conflicts	d	The Average Number of Conflict for the Family per Week
Real	376	86	4.37
IP	224	86	2.60

Table C-3

Real Timetable for Burqa Secondary School for Girls

Day		5a	5b	6	7a	7b	8a	8b	9a	9b	10	11a	11b
Sunday	8:20:09	Arabic	Arabic	English	English	English	Math	Math	Math	Math	Math	Tech	Physic
	9-9:40	Math	Religion	Science	Arabic	Arabic	English	English	Tech	Tech	Chemistry	English	Math
	10-10:40	Religion	Math	Tech	Science	Science	S-Sciences	S-Sciences	Arabic	Arabic	Physic	Arabic	Arabic
	10:50-11:30	English	English	Math	S-Sciences	S-Sciences	Arabic	Arabic	Religion	Religion	Biology	Math	Chemistry
	11:40-12:20	Science	Science	Arabic	Religion	Religion			Math	Math	English	History	Tech
	12:20 -1				Math	Math							
Monday	8:20:09	Math											
	9-9:40	Tech	Tech	English	Arabic	Arabic	Math	Math	S-Sciences	S-Sciences	English		Physic
	10-10:40	Science	Science	Arabic	Math	Math	Arabic	Arabic	English	English	Biology	Math	Chemistry
	10:50-11:30	Arabic	Arabic	Math	Tech	Tech	English	English	Religion	Religion	S-Sciences	Religion	Religion
	11:40-12:20		Math	Sport	English	English	Religion	Religion	Arabic	Arabic	Chemistry	Arabic	Arabic
Tuesday	9-9:40			Science	S-Sciences	S-Sciences	Tech	Tech	English	Math	English	Tech	Physic
	10-10:40	S-Sciences	S-Sciences	Religion	Religion	Religion	English	English	Science	Science	Math	Math	Tech
	10:50-11:30	English	English	Arabic	Science	Science	Science	Science	Math	English	Tech	History	Math
	11:40-12:20	Religion	Math		Sport	Sport			S-Sciences	S-Sciences	Arabic		Biology
Wednesday	9-9:40	Math	Religion	English	Tech	Tech	Math	Math	Religion	Religion	Physic	Geography	
	10-10:40	Science	Science	Math	English	English	Religion	Religion	Arabic	Math	Tech	History	Biology
	10:50-11:30	Tech	Tech	S-Sciences	Religion	Religion	Sport	Sport	Math	Arabic	English	English	Math
	11:40-12:20	Sport	Sport		Arabic	Arabic	Science	Science	Science	Science	S-Sciences	Arabic	Arabic
Thursday	8:20:09									Math			
	9-9:40		Math	Arabic	Science	Science	Tech	Tech	English	English	Physic	Tech	Chemistry
	10-10:40	Arabic	Arabic	Tech	Math	Math	Arabic	Arabic	Math	Art	Math	Religion	Religion
	10:50-11:30	Math		Math	Arabic	Arabic	Science	Science	Tech	Tech	English	English	English
	11:40-12:20				Art	Art	S-Sciences	S-Sciences	Math	Arabic	Religion		Physic
	12:20 -1										Arabic		

Table C-4

The Result of Optimization Timetable Using Time Limit Half an Hour

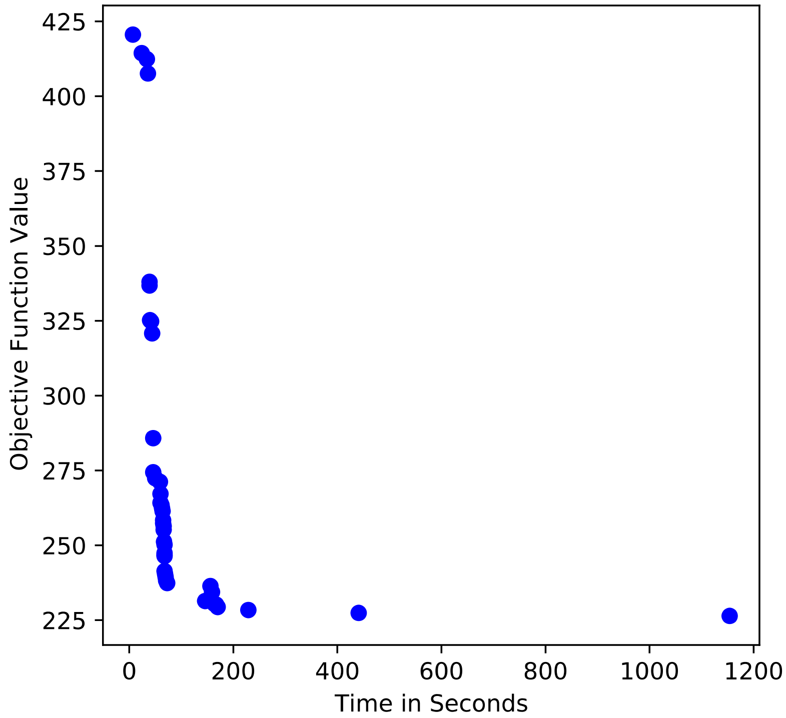
0		5a	5b	6	7a	7b	8a	8b	9a	9b	10	11a	11b
Sunday	9 - 8:20	Tech	English	Tech	S-Sciences	Religion	Arabic	Math	Math	English	Math	History	
	(9-9:40)		Arabic	Sport	Art	Tech	Science	English	English	Math	Arabic		Arabic
	(10-10:40)	Math		Science	Math	Science	Sport		S-Sciences	Religion	Religion		Physic
	(10:50-11:30)	English		Math	Arabic	Math		Religion	Tech	Art	English	Math	English
	(11:40-12:20)	Sport	Math		Religion	S-Sciences	Math	Science	Arabic	Arabic	S-Sciences	Tech	Math
	(12:20 -1)	Science	Math	English	Science	Religion	S-Sciences	Tech	Arabic		Physic	English	
Monday	9 - 8:20	Math		Arabic	English	Math		S-Sciences	Religion		Math		Tech
	(9-9:40)	Religion	Tech		Religion	Arabic	Arabic	English	English	Math	Chemistry		Chemistry
	(10-10:40)		Science		Tech	Sport	English	Science	Math	Arabic	English	Religion	
	(10:50-11:30)	Arabic		Math	Arabic	Science	Tech	Religion		Religion	S-Sciences	Geography	Arabic
	(11:40-12:20)		Arabic		Religion	S-Sciences	Science	Math	Math	Science	English	English	Biology
Tuesday	(9-9:40)	Math		Arabic	Science	Arabic	Religion		Tech	S-Sciences		Tech	Arabic
	(10-10:40)	Arabic	Sport	Math	Math	English	Tech	S-Sciences	Religion	Math	Physic	Arabic	
	(10:50-11:30)		Math	Religion		Religion	Math	Arabic	Science	Arabic	Tech		Math
	(11:40-12:20)				Arabic	Science	English	Tech	English	English		Math	Physic
Wednesday	(9-9:40)		Science	S-Sciences			Math	English	Math	Science	Math		Religion
	(10-10:40)		Science		Sport	Tech	English	Arabic	Math	English	Biology	Religion	Physic
	(10:50-11:30)	Science	English	Arabic	Math		S-Sciences	Sport	Arabic	Tech	English	Arabic	Chemistry
	(11:40-12:20)	Science		English	English	Arabic				S-Sciences	Physic		Tech
Thursday	9 - 8:20	S-Sciences	Religion	Math	Tech	English		Math	Religion	Math	Chemistry	History	Biology
	(9-9:40)	Arabic	Math	English	S-Sciences			Science		Tech	English	Tech	Chemistry
	(10-10:40)	Math	S-Sciences	Arabic	Arabic	English	Religion		S-Sciences	Religion	Tech	Arabic	
	(10:50-11:30)	Tech	Arabic	Tech		Art	Science	Arabic	Science	Math	Arabic	English	Math
	(11:40-12:20)	English	Religion	Science	Science	Arabic						Math	Physic
	(12:20 -1)	Religion	Tech		English	Math	Arabic		Math	Arabic	Biology	History	Religion

Appendix D

Figure of Study's

Figure D-1

The Change of Objective Function Value with Time.



جـامعـــة النجاح الوطنية

كليـــة الدراســـات العليـــا

خوارزمية أتمته البرنامج المدرسي في ظل الوباء : المدارس الفلسطينية كحالة اختبار

إعداد

ايمان وجيه عثمان مخيمر

إشراف

د. أمجد هواش

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجه الماجستير في الحوسبة المتقدمة، من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس– فلسطين.

2022

خوارزمية أتمته البرنامج المدرسي في ظل الوباء: المدارس الفلسطينية كحالة اختبار

اعداد

ايمان وجيه عثمان مخيمر

إشراف

د. أمجد هواش

الملخص

خلال جائحة COVID-19، تم اقتراح التعلم عن بعد كحل حيوي لمواصلة عملية التدريس والتعلم، ولإبقاء الطلاب والمعلمين في المدارس أو الجامعات على اتصال، ولتجنب العدوى المحتملة بينهم. ومع ذلك وعلى الرغم من جميع الإيجابيات ودوره في القضاء على فرصة العدوى المحتملة، إلا أنه يعاني من عدة عيوب (في بعض المجتمعات) منها: نقص خبرة التعلم عن بعد لكل من الطلاب والمعلمين، والحاجة إلى تحفيز عال للطلاب، وكذلك الحاجة إلى عدد كاف من الأجهزة خاصة إذا كان لدى الأسرة أكثر من طالب مدرسة أو جامعة. الغرض من هذه الدراسة هو اقتراح حل للمشكلة الأخيرة من خلال الجدولة المناسبة لجلسات الفصول الدراسية مع مراعاة جميع العوامل المؤثرة مثل عدد الدروس لكل معلم وعدد الإخوة الطلاب، وعدد الأجهزة في كل أسرة. وقد تم تطبيق الدراسة على 4 مواد دراسية أساسية: اللغة العربية، واللغة الإنجليزية، والعلوم، والرياضيات. وتم تجريبها على 4 مدارس فلسطينية. حيث قمنا بنمذجة المشكلة واستخدمنا البرمجة الصحيحة في حلها وباستخدام برنامج Gurobi. وتم اجراء اختبارات تجريبية شاملة لمقارنة بين عملنا والإعداد اليدوي لجدولة الدروس وتمكنا من خلالها تحقيق نتائج واعدة.

الكلمات المفتاحية: الجدول الزمني للمدرسة، الجدولة، البرمجة الصحيحة.