A preliminary Case Study on University Course Timetabling using Constraint Programming and Operations Research

Matthieu Py[1]
Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Étienne, LIMOS, 63000 Clermont-Ferrand, France

Mohamed Amir Keskes
Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Étienne, LIMOS, 63000 Clermont-Ferrand, France

Abstract

This paper addresses a real-world problem in university timetable management. We begin by describing the problem in detail, focusing on one particular teaching department, and then we propose a possible solution using constraint programming and operations research techniques. Finally, we discuss the advancements and trends in university timetable management over recent years.

2012 ACM Subject Classification Applied computing → Education; Theory of computation → Constraint and logic programming; Theory of computation → Discrete optimization; Applied computing → Operations research

Keywords and phrases Constraint Programming ; Operations Research ; Educational Scheduling ; University Timetable Management ; Case Study

Digital Object Identifier 10.4230/LIPIcs.Soft 2024.2024.5

Acknowledgements We want to thank the First International Workshop on Discrete Optimization with Soft Constraints for the opportunity to present this short introduction of our current and future work.

1 Introduction

Timetabling problems are one of the most common and well-known classes of constrained optimization problems. These problems concern all companies and organizations, including universities, particularly with the management of their teaching schedules. Although there are probably many scientific works on the subject, the practical use of constrained optimization techniques for the management of university timetables remains non-existent in many establishments.

This paper aims to present preliminary work on this topic that we are currently exploring in real-world context. For the next academic year, we are trying to explore the possibilities to improve the management of timetables within our teaching department. To this aim, we need to study what are the specific needs to be taken into account in this department. Based on this knowledge, we will propose a first solution in order to obtain a prototype for the next school year. In the longer term, we will end by asking some questions about the most effective techniques for solving this type of problem, and the different difficulties that may be encountered.

[1] corresponding author

First International Workshop on Discrete Optimization with Soft Constraints.
Editors: Simon de Givry and Javier Larrosa; Article No. 5; pp. 5:1–5:11
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This paper is organized as follows. We start with a preliminary Section 2 on one more restricted problematic which is the oral and written defenses of apprenticeship students. Then, Section 3 presents the university timetabling problem and specific needs of our teaching department. Section 4 presents the proposed solution for automatic timetable management. Section 5 offers discussions on the theme of timetable management in order to open up interesting perspectives of this preliminary work.

2 Preliminary: Oral and Written Defense for Apprenticeship Students

In this section, the case study is restricted to the oral and written defenses of apprenticeship students only. This will provide a framework for addressing the issues related to the overall scheduling challenges of the teaching department.

2.1 Data

There are between 30 and 50 students, between 10 and 15 teachers, and between 8 and 12 time slots.

Each student must be assigned a time slot for their oral defense (not for their written defense which is independent of the time slot). Each student has a supervisor (among the teachers) which is assigned to the oral defense and the written defense of the student. A second teacher must be assigned for each oral defense and for each written defense, and it is forbidden to have the same teach for both defenses.

Each time slot has a limited number of oral defenses (approximately 5) and corresponds to a specific day and hour. Each teacher must participate in exactly a certain number of defenses (usually twice the number of students they supervise). Some teachers have time slots during which they are unavailable. Certain specific constraints exist, such as a pair of teachers where at least one must be absent during certain time slots (but either one of the two).

The quality of the proposed schedule depends on several factors, including:

- The students involved have already had written and oral defenses in previous years. It is important to minimize assigning a teacher to a student if this assignment has already occurred before.
- It is necessary to group the teachers' participation as much as possible so that they make the fewest possible trips to conduct the defenses (as some teachers also work at other locations).

2.2 A first prototype using Hexaly

A first usable prototype has been implemented using the optimization solver Helaxy (formerly called LocalSolver) [3]. The prototype can be broken down into three parts:

- An Excel file [10] containing all the data.
- A computer program which models and proposes a solution to the timetabling problem.
- An Excel file which contains the proposed timetable.

We describe informally below the mathematical model (which is mathematically represented in a format similar to that of a Weighted Constraint Satisfaction Problem) and we use the optimization solver Hexaly (formerly called LocalSolver) [3] to solve a solution within a one-minut timeout.

Decision variables:

The decision variables are all boolean and are the following ones:

- For each student and for each time slot: Is the oral defense of this student assigned to this time slot?
- For each student and for each teacher: Is this teacher assigned to be the second teacher in the oral defense of this student?
- For each student and for each teacher: Is this teacher assigned to be the second teacher in the written defense of this student?

(Hard) Constraints:

The constraints are the following ones:

- Each student must be assigned to exactly one time slot, one teacher for the oral defense (plus their supervisor) and one teacher for the written defense (plus their supervisor).
- Each time slot must contain at most a certain number of oral defenses.
- Each teacher can only be assigned to one oral defense per time slot.
- Each teacher cannot be assigned to the oral defense and the written defense of the same student.
- Each teacher must be assigned to the number of oral defenses given in the data.
- Each teacher cannot be assigned on time slot on which he is absent.
- ... (additional requests must be taken into account)

Objective Functions / Soft Constraints:

The objective function contains the following parts:

- For each teacher and for each student he/she have already evaluated in a previous defense, a penalty is applied if a teacher is assigned to this student.
- For each teacher, a penalty is applied for each day the teacher is assigned to at least one oral defense.
- ... (additional demands must be taken into account)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Student	Supervisor	Oral T.	Written T.		Student	Supervisor	Oral T.	Written T.		Stu
2													
3	Thursday 8h-10h		Student5	Teacher9	Teacher6	Teacher1		Student11	Teacher12	Teacher5	Teacher2		Stud
4													
5	Thursday 10h-12h		Student3	Teacher7	Teacher12	Teacher6		Student22	Teacher1	Teacher6	Teacher7		Stud
6													
7	Thursday 14h-16h		Student7	Teacher2	Teacher3	Teacher6		Student8	Teacher5	Teacher6	Teacher12		Stud
8													
9	Thursday 16h-18h		Student2	Teacher6	Teacher2	Teacher3		Student15	Teacher3	Teacher9	Teacher11		Stud
10													
11	Friday 8h-10h		Student10	Teacher6	Teacher8	Teacher3		Student18	Teacher9	Teacher11	Teacher8		Stud
12													
13	Friday 10h-12h		Student6	Teacher3	Teacher9	Teacher7		Student19	Teacher7	Teacher6	Teacher12		Stud
14													
15	Friday 14h-16h		Student4	Teacher11	Teacher8	Teacher5		Student13	Teacher6	Teacher9	Teacher8		Stud
16													
17	Friday 16h-18h		Student1	Teacher8	Teacher9	Teacher7		Student14	Teacher6	Teacher4	Teacher5		Stud

Figure 1 An example of generated timetabling solution for the Oral and Written Defense for Apprenticeship Students

2.3 Results and Post-Optimization

An example of result given by the prototype is given in Figure 1. However, it is common to have to modify the schedule several times depending on new constraints (for example teacher unavailability). An additional part of the objective function has consequently be added, which is minimizing the number of modifications of the solution if an update is needed. An example of a such update is given in Figure 2, after Teacher 11 was known absent on Friday.

	Student	Supervisor	Oral T.	Written T.	Student	Supervisor	Oral T.	Written T.	Stud
Thursday 8h-10h	Student5	Teacher9	Teacher6	Teacher3	Student11	Teacher12	Teacher5	Teacher6	Stude
Thursday 10h-12h	Student3	Teacher7	Teacher12	Teacher8	Student26	Teacher11	Teacher6	Teacher7	Stude
Thursday 14h-16h	Student4	Teacher11	Teacher12	Teacher8	Student7	Teacher2	Teacher3	Teacher6	Stud
Thursday 16h-18h	Student2	Teacher6	Teacher2	Teacher3	Student15	Teacher3	Teacher9	Teacher5	Stude
Friday 8h-10h	Student10	Teacher6	Teacher8	Teacher9	Student18	Teacher9	Teacher1	Teacher11	Stude
Friday 10h-12h	Student6	Teacher3	Teacher1	Teacher9	Student19	Teacher7	Teacher6	Teacher2	Stude
Friday 14h-16h	Student13	Teacher6	Teacher8	Teacher9	Student21	Teacher1	Teacher7	Teacher12	Stude
Friday 16h-18h	Student1	Teacher8	Teacher9	Teacher4	Student14	Teacher6	Teacher4	Teacher7	Stude

Figure 2 An example of generated timetabling solution for the Oral and Written Defense for Apprenticeship Students after the modification of a first version

3 Case study and presentation of the problem

In this section, we focus on the study of the global needs of one teaching department for one semester. Currently, timetabling is managed by a teacher whose are of expertise does not include mathematics or computers science. This colleague must manage two complementary aspects of timetable management:

- The forecast part: this involves planning the schedules in advance for future weeks
- The adaptive part: it involves modifying the forecast schedule so that it adapts to daily unforeseen events that affect the current schedule.

The objective of this current work is to lighten and help reduce the important management load of the timetabling task, especially in the forecast aspect.

3.1 Students

There are between 80 and 160 students each year studying in the teaching department.

A student belongs to:

- One year of teaching: first year, second year, third year or special year (the special year allows you to follow the first two years of teaching in a single year).
- One group of Supervised Work (each year contains between one and two groups of supervised work)
- A group of Practical Work (each year contains between one and four groups of practical work)
- A language group: each year contains between one and two groups for Spanish and each of the other languages, German, Italian and French, has one group containing all the students who participate whatever their year.

- Each student is either an Apprentice or an Intern. Apprentices alternate between weeks at university and weeks in which they work in a company. Interns remain at the university most of the year but have specific period reserved for an internship to be conducted within a company. First-year students are all interns. In second and third years, students choose between being an apprentice or being an intern.

We deduce from this information a set of student profiles P, where each student profile $p \in P$ is the non-empty intersection of a Year, a Supervised Work group, a Practical Work group, a Language group and of professional status (Apprentice or Intern). Each student profile can have unavailability and requests, like avoiding to have gaps between sessions on the same day (except for lunch).

3.2 Teaching Staff

One of the particularities of the training is the diversity of the instructors who provide the course sessions. In addition to teachers and teacher-researchers, many sessions are provided by external speakers, most often professionals from companies related to the areas of training.

Each instructor has constraints and requests, particularly linked to their status. For example:

- External speakers must travel to the training site as little as possible, which means grouping their course sessions together as much as possible.
- Teacher-researchers have recurring days dedicated to research.

Teachers and teacher-researchers assigned to the department site can have unavailabilities. Certain special cases must also be treated, including for example:

- Individual preferences: some teachers prefer to start early in the morning while others prefer to finish late in the evening.
- Family or personal constraints: For instance, we have a couple of teacher-researcher parents, at least one of whom must be free when it is time to drop the children off at school.
- The teaching load must be smoothed as much as possible over the year, which can be difficult because of apprentice students.

3.3 Classrooms

We can estimate at first glance that the classrooms are completely classic: amphitheater, Supervised Working Rooms and Practical Working Rooms. However, the use of the rooms differs depending on its specific characteristics. For example, one room is specifically designed for English Practical Work. This will be developed in the subsection on course sessions.

3.4 Calendar

We are currently interested in the autumn semester, which extends from approximately the beginning of September to the end of January (with some nuances depending on the years of students). Each student profile (year, apprentice or intern, etc.) has its own calendar, with its weeks of presence and absence.

Each week $w \in W$ has the following information:

- The number of this week in the calendar
- For each year, the maximum number of working hours of this year in the week.

3.5 Teachings and Sessions

The autumn semester contains approximately 100 distinct courses, each course was composed of a certain number of sessions. Each session belongs to a course, with the exception of a few specific sessions, for example the back-to-school session. Depending on the number of groups, we obtain approximately 1500 sessions to be placed over the semester.

Each session $s \in S$ has the following information:

- An identifier.
- The name of the overlying course. For simplification purposes, we consider a "Miscellaneous" course containing isolated sessions.
- A duration, in minutes. Most sessions last between 30 and 240 minutes.
- The list of student profiles who must follow this session.
- The list of speakers who must lead this session. Some sessions, such as exams, do not have an assigned speaker but have a list of teachers who may be assigned to that session.
- A list of rooms that can be used for this session, and the number of rooms required.
- Information specific to the session, for example if it is an exam.

Sessions can be affected by specific constraints, including but not restricted to:

- Precedence constraint: in many cases, one session must completed before starting another session
- Proximity constraint: the time gap between two sessions must be within a certain interval of time
- Already-planned-session: for some sessions, we already know when the session will be allocated.

4 Towards a first prototype

The initial objective is to obtain a usable prototype, which will then gradually evolve over the years. The prototype will be broken down into three parts:

- An Excel file [10] containing all the problem data.
- A computer program which models and proposes a solution to the timetabling problem.
- An Excel file which contains the proposed timetable.

To solve the University Timetabling Problem, for this first prototype we propose to build a mathematical model of this real-world problem (in a format similar to that of a Weighted Constraint Satisfaction Problem) and to find a solution using the optimization solver Hexaly (formerly called LocalSolver) [3]. Given the large number of sessions to be placed, the modeling and resolution of the problem was divided into independent parts summarized in the Algorithm 1:

- **Planning:** For each session, we decide (only) on which week this session will take place
- **Scheduling:** For each week and for each session planned for this week, we decide on which day, at what time and in which room(s) this session will take place.

Algorithm 1 University timetabling algorithm

Require: University Timetabling Data
Ensure: For each session, the week, day, time and room(s) affected to this session
```
Solve the Planning Problem
for all Week in the Calendar do
    Solve the Scheduling Problem for every sesssions affected to this week
end for
```

Below is a brief overview of the proposed mathematical models.

4.1 Planification

The first part of the proposed algorithm is to assign, for each session, the week on which the session will be executed.

Decision variables:

Decision variables are the following ones:

- For each session and for each week, is this session executed on this week?

(Hard) Constraints:

Constraints are the following ones:

1. Each session must be executed on exactly one week.
2. For each week and each student profile, the number of hours affected to this profile in this week cannot exceed a maximum amount.
3. Similarly, for each week and each instructor, the number of hours affected to this instructor in this week cannot exceed a maximum amount.
4. For each week and each classroom, the number of hours affected to this classroom in this week cannot exceed a maximum amount.
   - As we have not yet chosen the classroom of each session at this stage, we temporarily consider that if a session needs one classroom among two possibilities for two hours, it is the same as asking for both classrooms for one hour. This simplification will have to be removed in later work.
5. We must respect every specific constraint requested about the sessions, the instructors, the students, ...

Objective Functions / Soft Constraints:

The current objective function enables to compress the timetable as much as possible, i.e. minimize the weighted sum by week ID of session assignments.

4.2 Scheduling

After every session is allocated to a week, we iteratively compute on every week, and on every session affected to this week, the day, the time and the classroom of this session. If some sessions cannot be successfully scheduled, they are rescheduled for the next available week. The time is discretized on time periods which lasts 30 minutes.

Decision variables:

We define the following decision variables:

- For each session affected to this week, is this session successfully scheduled?
- For each session affected to this week and for each day, is this session scheduled on this day?
- For each session affected to this week and for each time period, does this session start at this time period?
- For each session affected to this week and for each classroom, is this session using this classroom?

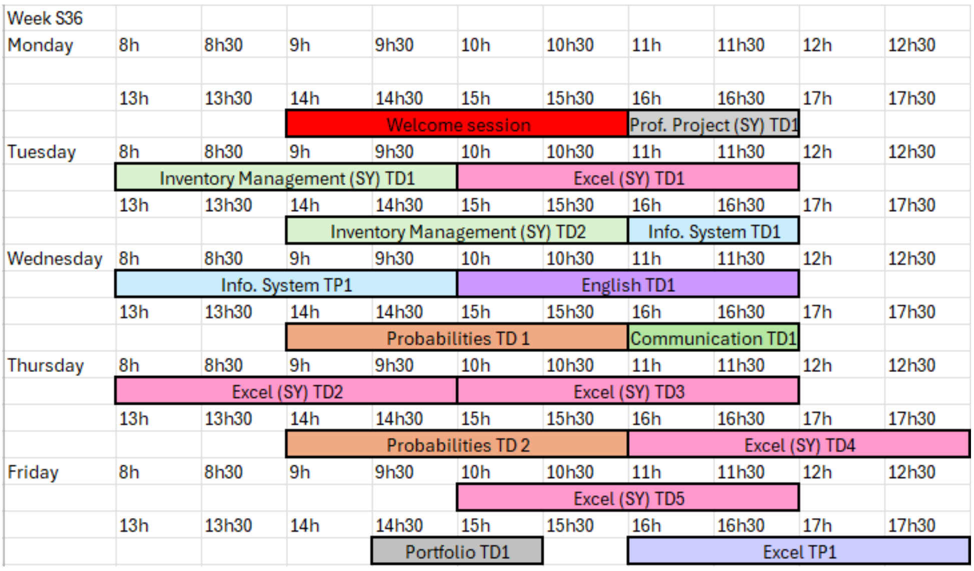
(Hard) Constraints:

Constraints are the following ones:

1. Each successfully scheduled session must be affected to exactly one day/time.
2. Each student profile can participate in at most one session at a time.
3. Similarly, each instructor can participate in at most one session at a time.
4. Similarly, each classroom can be used by at most one session at a time.
5. We must respect every specific constraint about the sessions, the instructors, the students, ...

Objective Functions / Soft Constraints:

Two objective functions are identified in this situation. The first objective function is to minimize the number of delayed session (i.e. maximize the number of successfully scheduled sessions). The second one is to satisfy user demands as much as possible (students and instructors).



Week S36										
Monday	8h	8h30	9h	9h30	10h	10h30	11h	11h30	12h	12h30
	13h	13h30	14h	14h30	15h	15h30	16h	16h30	17h	17h30
			Welcome session				Prof. Project (SY) TD1			
Tuesday	8h	8h30	9h	9h30	10h	10h30	11h	11h30	12h	12h30
	Inventory Management (SY) TD1				Excel (SY) TD1					
	13h	13h30	14h	14h30	15h	15h30	16h	16h30	17h	17h30
			Inventory Management (SY) TD2				Info. System TD1			
Wednesday	8h	8h30	9h	9h30	10h	10h30	11h	11h30	12h	12h30
	Info. System TP1				English TD1					
	13h	13h30	14h	14h30	15h	15h30	16h	16h30	17h	17h30
			Probabilities TD 1				Communication TD1			
Thursday	8h	8h30	9h	9h30	10h	10h30	11h	11h30	12h	12h30
	Excel (SY) TD2				Excel (SY) TD3					
	13h	13h30	14h	14h30	15h	15h30	16h	16h30	17h	17h30
			Probabilities TD 2				Excel (SY) TD4			
Friday	8h	8h30	9h	9h30	10h	10h30	11h	11h30	12h	12h30
					Excel (SY) TD5					
	13h	13h30	14h	14h30	15h	15h30	16h	16h30	17h	17h30
				Portfolio TD1			Excel TP1			

Figure 3 An example of generated timetabling solution for the Special Year Students

5 Discussions

5.1 Towards a reliable and efficient prototype for our case study

We present in Figure 3 an example of what we are able to produce. This example shows, for one student profile, their planning for the first week of the calendar (we translated the information on the Figure in English compared to the real names). Despite this encouraging results and our abilities to satisfy every constraints of the problem, a lot of work still has to do to to respect the wishes of users and the results remains not good enough. In particular, the overall timetable contains many empty slots (students have classes before the slot and after the slot, but not during the slot), empty slots caused by some bad assignments taken by the planification part of the process. Moreover, the identification of the specific needs of the teaching department, the collect of useful data, the creation of the mathematical model and the verification of the correctness of the implementation are currently being carried out.

Once this first step has been completed, it would be interesting to compare this first prototype with other techniques. The state of the art on this theme remains to be carried out but we can already cite some potential directions, like Meta-heuristics [1, 13, 14, 15], Linear Programming [6, 11, 12], Constraint Satisfaction Problem [16], and Boolean Satisfiability Problem [7, 8] among others [2, 4, 5, 9].

5.2 Multiplicity of actors, an obstacle to practical application

From our meager experience in a few French universities, the impression remains that very often, timetables were drawn up almost entirely by hand. One of the obstacles can be the multiplicity of the actors. Indeed, there are different actors involved in the university timetable management:

- There is at least one person who manages the schedules. This is often a person who has experience in management and human contact, sometimes a member of the administrative staff. It is rarely a person with skills in decision-making processes like constraint programming or operational research.

- There is at least one person, or an external company, that manages the tool that displays timetables. Similarly, this person rarely has skills in decision-making processes like constraint programming or operational research. His/Her priority is to secure the process of displaying and accessing timetables, and consequently the automation of their creation often remains in contradiction.
- Finally, the people with knowledge in constraint programming or operational research are often not involved in the timetabling management process.

How to bring together these people to ensure a better utilization of constraint programming and operations research in order to improve decision-making in university timetabling? This seems to be a relevant question.

5.3 How to generalize a such work to several teaching departments?

Another important part of our work will also aim to tackle the generalization of timetabling management for several teaching departments. Automating this task across multiple teaching departments presents significant challenges due to the unique and diverse needs of each department. While one department may have to build a schedule and repeat it on several weeks, others can aim to create a different schedule every week. The variability in course structures means that a one-size-fits-all solution is very difficult to put in practice. Developing such a system necessitates a deep understanding of the distinct operational nuances of each department, making the task complex and resource-intensive.

References

1 Sina Abdipoor, Razali Yaakob, Say Leng Goh, and Salwani Abdullah. Meta-heuristic approaches for the University Course Timetabling Problem. *Intelligent Systems with Applications*, 19:200253, 2023.

2 Hamed Babaei, Jaber Karimpour, and Amin Hadidi. A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59, 2015.

3 Thierry Benoist, Bertrand Estellon, Frédéric Gardi, Romain Megel, and Karim Nouioua. Localsolver 1.x: a black-box local-search solver for 0-1 programming. *4OR - A Quarterly Journal of Operations Research*, 9(3):299–316, 2011.

4 Edmund K. Burke, Jakub Marecek, Andrew J. Parkes, and Hana Rudová. Decomposition, reformulation, and diving in university course timetabling. *Computers and Operations Research*, 37(3):582–597, 2010.

5 Mei Ching Chen, San-Nah Sze, Say Leng Goh, Nasser R. Sabar, and Graham Kendall. A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. *IEEE Access*, 9:106515–106529, 2021.

6 Sophia Daskalaki, Theodore Birbas, and Efthymios Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135, 2004.

7 Emir Demirovic, Nysret Musliu, and Felix Winter. Modeling and solving staff scheduling with partial weighted maxsat. *Annals of Operations Research*, 275(1):79–99, 2019.

8 Alexandre Lemos, Pedro T. Monteiro, and Inês Lynce. Introducing UniCorT: an iterative university course timetabling tool with MaxSAT. *Journal of Scheduling*, 25(4):371–390, 2022.

9 Michael Lindahl, Matias Sørensen, and Thomas R. Stidsen. A fix-and-optimize matheuristic for university timetabling. *Journal of Heuristics*, 24(4):645–665, 2018.

10 Microsoft Corporation. Microsoft Excel. https://office.microsoft.com/excel , 2024.

11 Mozhgan Mokhtari, Majid Vaziri Sarashk, Milad Asadpour, Nadia Saeidi, and Omid Boyer. Developing a Model for the University Course Timetabling Problem: A Case Study. *Complex.*, 2021.
12 Efstratios Rappos, Eric Thiémard, Stephan Robert, and Jean-François Hêche. A mixed-integer programming approach for solving university course timetabling problems. *Journal Of Scheduling*, 25(4):391–404, 2022.
13 Amin Rezaeipanah, Samaneh Sechin Matoori, and Gholamreza Ahmadi. A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. *Applied Intelligence*, 51(1):467–492, 2021.
14 Ting Song, Sanya Liu, Xiangyang Tang, Xicheng Peng, and Mao Chen. An iterated local search algorithm for the University Course Timetabling Problem. *Applied Soft Computing*, 68:597–608, 2018.
15 Kadri Sylejmani, Edon Gashi, and Adrian Ymeri. Simulated annealing with penalization for university course timetabling. *Journal of Scheduling*, 26(5):497–517, 2023.
16 Christos Valouxis and Efthymios Housos. Constraint programming approach for school timetabling. *Computers And Operations Research*, 30(10):1555–1572, 2003.