The Post-Enrollment Course Timetabling Problem with Flexible Teacher Assignments

Richard Hoshino and Zilei Liu

Northeastern University, Vancouver, British Columbia, Canada
{r.hoshino, liu.zile}@northeastern.edu

Abstract. Scholars in Operations Research have studied automated timetabling for the past sixty years, developing algorithms that assign students and teachers to courses and timeslots. In the Post-Enrollment Course Timetabling Problem (PECTP), we seek the optimal timetable that maximizes students getting enrolled in their requested courses.

Given the complexity of the NP-complete PECTP, most high schools pre-assign teachers to each section of each course, and then build their Master Timetable. This action reduces the number of feasible timetables, since no teacher can be teaching two courses in the same timeslot, and most teachers are required to have a non-teaching timeslot every day.

In this paper, we explain how we created the Master Timetable for a Canadian high school that intentionally does not pre-assign teachers to courses. Instead, each course has a set of possible teachers, and each teacher has a fixed number of courses they must be assigned. By providing flexible teacher assignments, this school increases the likelihood that all students get into the courses they select. Our final Master Timetable enrolls students in 99.8% of their requested courses (3557 out of 3565), which is just one shy of the provably-optimal upper bound.

Keywords: School Timetabling · Integer Programming · Optimization.

1 Introduction

Every high school in the world requires a Master Timetable, listing the complete set of offered courses, along with the timeslot and classroom for each section of that course. From the timetable, each instructor knows what they are teaching, where they are teaching, and when they are teaching.

Due to the complexity of creating a Master Timetable satisfying hundreds of constraints, Operations Research scholars have been analyzing the School Timetabling Problem (STP) since the 1960s [13], creating timetables for high schools around the world [21]. In the most basic version of the STP, the objective is to assign courses to teachers, timeslots, and classrooms, subject to the following constraints: a teacher cannot teach two courses in the same timeslot, no classroom can be used by two courses simultaneously, and each teacher has a set of unavailable teaching timeslots. This problem is NP-complete [5].

Given the challenge of creating a Master Timetable, some school administrators focus only on teacher requirements and preferences, disregarding the course requests of the students (i.e., the individuals most affected by the timetable). If a high school decides to offer only one section of courses X and Y, and both of these courses are placed in the same timeslot, then no student can take courses X and Y. Bad timetabling decisions have a significant impact on a student's future, especially when they are prevented from taking a course they need.

To advance the field of educational timetabling, a group of researchers introduced the Post-Enrollment Course Timetabling Problem (PECTP), to incorporate student course preferences into the STP [18]. The PECTP involves student-related hard constraints, such as ensuring that no student is enrolled in multiple sections of the same course, and the objective function is to maximize the number of occurrences where students are enrolled in their desired courses.

In the PECTP, points are awarded for enrolling students in any section of a desired course. For example, if there are three different sections of Calculus, a student wishing to take Calculus needs to be assigned to exactly one of these three sections.

The lead author has created 50 Master Timetables for various Canadian high schools over the past six years, using his published algorithms to solve large real-life PECTP instances for schools: using graph colouring [15] and large neighbourhood search [16]. In every high school but one (see Section 4), schools *pre-assign* teachers to every section of every course. This action limits the percentage of students who can get enrolled in their desired courses.

To illustrate the shortcoming of pre-assigning teachers to courses, consider a simple timetable with two timeslots, where each of four students (S_1 to S_4) wishes to enroll in two courses chosen among four course offerings (C_1 to C_4). Each course is taught by one of two teachers (T_1 or T_2).

Consider the following scenario where teachers are pre-assigned to their courses, and students pick two of the four courses.

Course	Assigned Teacher		Student	Requested Courses
C_1	T_1		S_1	C_1, C_3
C_2	T_1		S_2	C_1, C_4
C_3	T_2		S_3	C_2, C_3
C_4	T_2		S_4	C_2, C_4

Given that our timetable has two timeslots, and given that a teacher cannot be teaching two courses in the same timeslot, the timetable must either be the one on the left or the one on the right.

Course	Teacher	Timeslot		Course	Teacher	Timeslot
C_1	T_1	1		C_1	T_1	1
C_3	T_2	1		C_4	T_2	1
C_2	T_1	2		C_2	T_1	2
C_4	T_2	2		C_3	T_2	2

In the timetable on the left, students S_1 and S_4 can only take one of their two courses, since both of their requested courses are offered in the same timeslot. And in the timetable on the right, students S_2 and S_3 can only take one of their two courses, for the same reason. In both scenarios, only 6 out of 8 student course requests are satisfied, with two students unable to enroll in a desired course.

Now we simply ask: *must* these courses be taught by these teachers? Suppose C_2 and C_3 can be taught by either T_1 or T_2; we just need each teacher teaching exactly 2 out of the 4 courses. By swapping the teacher assignments for C_2 and C_3, we now have all 8 student course requests satisfied, increasing our success rate from 75% to 100%. Below is the optimal timetable, listing the teacher and students for each course in each timeslot.

Course	Teacher	Timeslot	Enrolled Students
C_1	T_1	1	S_1, S_2
C_2	T_2	1	S_3, S_4
C_3	T_1	2	S_1, S_3
C_4	T_2	2	S_2, S_4

Ideally, schools would not pre-assign teachers to every section of every course, and instead allow the timetabling algorithm to find the best assignment of teachers to course sections to maximize students getting enrolled in their desired courses. Later in the paper we will demonstrate the effectiveness of this approach for a Canadian high school where numerous courses have a set of possible teachers who can be assigned to that course.

This paper proceeds as follows. In Section 2, we provide a brief literature review on related work. In Section 3, we describe our solution to the PECTP by formulating it as an Integer Linear Program (ILP) with five-dimensional binary variables. In Section 4, we apply our model to generate the Master Timetable for a Canadian high school, which succeeded in satisfying 3557 out of 3565 course requests from the 403 students. In Section 5, we conclude the paper with directions for future research.

2 Related Work

Educational timetabling is split into three main areas: high school timetabling, university course timetabling, and university examination timetabling [2].

High school timetabling problems involve the scheduling of all courses, ensuring the satisfaction of hard constraints, such as preventing teachers from being assigned to multiple courses at the same time [2]. The introduction of the XHSTT format [23] provides standardized specifications for addressing high school timetabling instances. Moreover, a variety of solution techniques and approaches have been developed to tackle this problem. As described in a recent survey paper [27], these methods include algorithms based on integer programming [1, 3, 8, 17, 25, 28], tabu search [19], simulated annealing [29], matheuristics approaches [4] [7] [24] and metaheuristic algorithms such as adaptive large neighbourhood search [26] and variable neighbourhood search [6].

While metaheuristic algorithms are capable of generating high-quality solutions in a short amount of time, they do not guarantee optimal results. In contrast, mathematical optimization techniques based on integer programming can find optimal solutions, particularly for smaller problem instances [8]. However, because timetabling is an NP-hard problem [21], mathematical optimization methods become less effective for medium- and large-sized problems [3, 24]. As a result, researchers have integrated exact mathematical optimization methods with metaheuristics in an approach known as matheuristics [27].

At some educational institutions, students are divided into cohorts, where they complete the same set of courses with everybody else in that cohort. In this case, an optimal timetable can be easily generated by solving an integer program [14, 17]. This is consistent with the observation that many instances in the XHSTT benchmark dataset have been solved to proven optimality [2]. A few of the XHSTT instances do involve the assignments of teachers to courses.

Alternatively, some high schools are organized similarly to universities, where students are not divided into fixed cohorts but instead choose their own set of desired courses. These timetabling problems fall under the category of University Course Timetabling, specifically the Post-Enrollment Course Timetabling Problem (PECTP), which was introduced in 2007 as one of the tracks of the International Timetabling Competition (ITC) [18].

PECTP follows a distinct formulation from XHSTT and has two benchmark datasets from ITC-2002 and ITC-2007 [2]. The same heuristics used to solve XHSTT instances can be applied to PECTP with appropriate adjustments. Since PECTP instances involve significantly more decision variables due to each student selecting a set of courses, researchers are rarely able to find the optimal solution. According to [2], the-state-the-art solution of ITC-2007 is achieved by local search methods, namely tabu search [20] and simulated annealing [9–11]

Most research on PECTP focuses on timetabling at universities, and all of the benchmark instances are artificial as they were obtained by a generator [2]. For example, ITC-2019 focuses exclusively on university timetabling.

Several researchers, including the lead author of this paper, have solved PECTP instances based on actual data sent from various high schools. Our PECTP instances have been tackled using matheuristics, through a bundling metaheuristic method [15] and Large Neighbourhood Search [16]. These Canadian PECTP instances are less complex than the benchmark ITC-2007 instances, as they have fewer students. Thus, a matheuristic approach is well-suited, as we can find an optimal (or close-to-optimal) solution in a short amount of time.

In this paper, we ask a question that we have not yet seen in the Educational Timetabling literature: "what if teacher assignments are flexible?" By specifying a set of possible teachers for each course, can we figure out the optimal assignment of teachers to courses to maximize students getting enrolled in their desired courses? Answering this question leads to better solutions, as we saw in the simple example in Section 1 with $|S| = 4$ students, $|C| = 4$ courses, and $|T| = 2$ teachers, where we improved our success rate from 75% to 100%.

We now provide our solution to the PECTP with flexible teacher assignments.

3 Mathematical Model

Each course c has one or more sections s. We define the binary decision variable $X_{s,c,d,p}$ to equal 1 if section s of course c is scheduled on day d in period p. Otherwise, $X_{s,c,d,p} = 0$.

We define the pair (s, c) as an *event* and each pair (d, p) pair as a *timeslot*. Each event is taught by one or more teachers and is offered in one or more rooms.

A school's timetable is a d-day repeating cycle, with p periods in each day. In our real-life school example in Section 4, we have $d = 2$ and $p = 5$, so we have ten timeslots (i.e., blocks) during which all sections of all courses occur.

Our Integer Linear Program (ILP) will generate the Master Timetable by assigning exactly one timeslot to each event. Using two other binary decision variables, we will determine the set of events assigned to each teacher, as well as the set of events assigned to each student.

The notation used in our Model is presented in Table 1.

Table 1: Notation used for the model

Symbol	Definition
Sets	
$t \in T$	set of teachers
$i \in I$	set of individual students
$c \in C$	set of courses
$r \in R$	set of rooms
$d \in D$	set of days
$p \in P$	set of periods in a day
$(s, c) \in E$	set of event tuples (s, c)
$(d, p) \in B$	set of block/timeslot tuples (d, p)
T_c	set of teachers qualified to teach course c
E_r	set of event tuples assigned to room r
Parameters	
$P_{i,c}$	an integer representing the preference coefficient of student $i \in I$ being enrolled in course $c \in C$.
$\#T_{s,c}$	the number of teachers that must be assigned to event (s, c), for each $(s, c) \in E$
$\#I_{s,c}$	the maximum number of students who can enroll in event (s, c), for each $(s, c) \in E$
Decision Variables	
$Z_{t,s,c,d,p}$	binary variable that indicates whether teacher t is assigned to event (s, c) in timeslot (d, p)
$Y_{i,s,c,d,p}$	binary variable that indicates whether student i is enrolled in event (s, c) in timeslot (d, p)
$X_{s,c,d,p}$	binary variable that indicates whether event (s, c) is scheduled in timeslot (d, p)

Our ILP has the following objective function, which we seek to maximize:

$$\sum_{i\in I}\sum_{(s,c)\in E}\sum_{(d,p)\in B} P_{i,c}\cdot Y_{i,s,c,d,p}$$

For each student $i \in I$, we denote by $P_{i,c}$ their preference for getting enrolled in a requested course c. This parameter will be some positive integer that is based on the student's grade (older students have more weight than younger students) as well as the course itself (required courses have more weight than elective courses). We assume that $P_{i,c}$ is a parameter that is independent of the section s, day d, and period p.

We now present our hard constraints. As we are employing a matheuristic approach that integrates Integer Linear Programming (ILP) with a Large Neighbourhood Search (LNS) heuristic, our constraints are structured into two stages, aligning with our two-stage solving strategy.

3.1 Stage I: Hard restrictions on course and teacher scheduling

Our first step is to find a feasible timetable that meets all of the hard constraints involving courses, teachers, and rooms, while ignoring student requests.

Each event (s, c) must be scheduled in exactly one timeslot (d, p).

$$\sum_{(d,p)\in B} X_{s,c,d,p} = 1 \qquad \forall\,(s,c)\in E \tag{1}$$

An event is scheduled in a timeslot if and only if at least one qualified teacher is assigned to this event in this timeslot.

$$X_{s,c,d,p} \le \sum_{t\in T_c} Z_{t,s,c,d,p} \qquad \forall\,(s,c)\in E,\ (d,p)\in B \tag{2}$$

$$X_{s,c,d,p} \ge Z_{t,s,c,d,p} \qquad \forall\ t\in T_c, (s,c)\in E,\ (d,p)\in B \tag{3}$$

A teacher cannot be assigned a course that they are unqualified to teach.

$$\sum_{(d,p)\in B} Z_{t,s,c,d,p} = 0 \qquad \forall\,(s,c)\in E, t\notin T_c \tag{4}$$

A teacher cannot be assigned to two different events in any timeslot.

$$\sum_{(s,c)\in E} Z_{t,s,c,d,p} \le 1 \qquad \forall\, t\in T,\ (d,p)\in B \tag{5}$$

A room cannot accommodate two different events in any timeslot.

$$\sum_{(s,c)\in E_r} X_{s,c,d,p} \le 1 \qquad \forall\, r\in R,\ (d,p)\in B \tag{6}$$

Each event must be assigned the required number of teachers.

$$\sum_{t\in T_c}\sum_{(d,p)\in B} Z_{t,s,c,d,p} = \#T_{s,c} \qquad \forall\,(s,c)\in E \tag{7}$$

The hard constraints (1)-(7) represent the fundamental requirements that must be satisfied in any valid timetable.

In addition to these, a school timetable has additional constraints. Following the structure of [14], we categorize these specific constraints into three families: restrictions on sets of events, restrictions on teacher assignments, and relationships between sets of events. A general formulation is provided for each family.

Family I: Restrictions on sets of events For each restriction, let E^* denote the set of events affected by the restriction and B^* denote the set of timeslots affected by the restriction. We have

$$\sum_{(s,c)\in E^*}\sum_{(d,p)\in B^*} X_{s,c,d,p} \;\; \{=,\leq,\geq\} \;\; N^* \tag{8}$$

For each of these constraints, we choose the appropriate sign from $\{=,\leq,\geq\}$ and we choose N^* to be a non-negative integer. Let us provide several examples to illustrate the versatility of this family of constraints.

(i) "All sections of AP Physics must be scheduled on Day 1" means E^* is the set of events with c equal to AP Physics, and B^* is the set of timeslots with $d \neq 1$. Our sign is $=$, and $N^* = 0$.
(ii) "Each timeslot can have at most four Math classes" means we create a separate constraint for each of the timeslots $(d,p) \in B$. For each timeslot B^*, we let E^* be the set of events for which c is a Math course, our sign is $\leq$, and $N^* = 4$.
(iii) "The art room must be used at least twice every day" means we create a separate constraint for each day d. For each day d, we list all the timeslots B^* for that day, we let E^* be the set of events that take place in the art room, our sign is $\geq$, and $N^* = 2$.

As we can see from the examples above, constraints involving sets of events can be expressed in the form of equation (8).

Family II: Restrictions on teacher assignments For each restriction, let E^* denote the set of events affected by the restriction and B^* denote the set of timeslots affected by the restriction. We have

$$\sum_{(s,c)\in E^*}\sum_{(d,p)\in B^*} Z_{t,s,c,d,p} \;\; \{=,\leq,\geq\} \;\; N^* \tag{9}$$

For each of these constraints, we choose the appropriate sign from $\{=,\leq,\geq\}$ and we choose N^* to be a non-negative integer. Let us provide several examples to illustrate the versatility of this family of constraints.

(iv) "Teacher Smith has a teaching load of five courses" means E^* is the set of events for which c is a course that Teacher Smith is qualified to teach, B^* is the set of all timeslots, our sign is $=$, and $N^* = 5$.
(v) "Teacher Smith must have at least one non-teaching period on Day 1" means E^* is the set of events for which c is a course that Teacher Smith is qualified to teach, B^* is the set of all timeslots with $d = 1$, our sign is $\leq$, and $N^* = |P| - 1$.
(vi) "Teacher Smith must teach at least one class every afternoon" means we create a separate constraint for each day d. For each day d, we list all the timeslots B^* that occur in the afternoon, we let E^* be the set of events for which c is a course that Teacher Smith is qualified to teach, our sign is $\geq$, and $N^* = 1$.

As we can see from the examples above, constraints involving teacher assignments can be expressed in the form of equation (9).

Family III: Relationships between sets of events For some integer $v \geq 2$, let $E_1, E_2, \ldots, E_v$ be a set of v events. Using a linear equation or linear inequality, we can model three additional timetabling constraints that relate these v events.

All v events must occur in the same timeslot.

$$X_{s_i,c_i,d,p} = X_{s_{i+1},c_{i+1},d,p} \quad \forall (d,p) \in B, i \in [1, v-1] \tag{10}$$

All v events must occur on the same day.

$$\sum_{p \in P} X_{s_i,c_i,d,p} = \sum_{p \in P} X_{s_{i+1},c_{i+1},d,p} \quad \forall d \in D, i \in [1, v-1] \tag{11}$$

The v events must occur on v different days.

$$\sum_{i \in [1,v]} \sum_{p \in P} X_{s_i,c_i,d,p} \leq 1 \quad \forall d \in D \tag{12}$$

This versatile and flexible framework enables us to model constraints that relate almost any set of events to each other. For example, we can ensure that certain courses are not scheduled on the same day, that two Grade 9 French classes occur in the exact same timeslot, and that a part-time teacher only needs to be at the school on just one of the $|D|$ days.

3.2 Stage II: Hard Constraints on Student Enrollment

Using equations (1)-(12), we find a feasible timetable by solving the ILP, where the variables $X_{s,c,d,p}$ tell us the timeslot for each event, and the variables $Z_{t,s,c,d,p}$ tell us the teaching assignment for each teacher. This is Stage I.

In Stage II, we incorporate the constraints involving the student decision variables $Y_{i,s,c,d,p}$.

Each student can be enrolled in at most one event per timeslot.

$$\sum_{(s,c)\in E} Y_{i,s,c,d,p} \le 1 \qquad \forall\, i \in I,\ (d,p) \in B \tag{13}$$

For any course, each student can be enrolled in at most one section of that course.

$$\sum_{(d,p)\in B} \sum_{s:(s,c)\in E} Y_{i,s,c,d,p} \le 1 \qquad \forall\, i \in I,\ c \in C \tag{14}$$

No student can be enrolled in an event during a timeslot in which that event is not scheduled.

$$Y_{i,s,c,d,p} \le X_{s,c,d,p} \qquad \forall\, i \in I,\ (s,c) \in E,\ (d,p) \in B \tag{15}$$

Each event has a maximum number of students who can be enrolled.

$$\sum_{i\in I} Y_{i,s,c,d,p} \le \#I_{s,c} \cdot X_{s,c,d,p} \qquad \forall\, (s,c) \in E,\ (d,p) \in B \tag{16}$$

Finally, no student can be enrolled in a course that they did not request. Specifically, if student i did not request course c, then for all events $(s,c) \in E$,

$$\sum_{(d,p)\in B} Y_{i,s,c,d,p} = 0 \tag{17}$$

This is our model for the PECTP with flexible teacher assignments.

Our optimal timetable is found by maximizing the objective function of this ILP subject to these seventeen constraints. Our solution consists of three families of binary variables: the variables $X_{s,c,d,p}$ tell us the timeslot for each event, the variables $Z_{t,s,c,d,p}$ tell us each teacher's set of assigned courses, and the variables $Y_{i,s,c,d,p}$ tell us each student's set of enrolled courses.

While our ILP is guaranteed to output an optimal timetable, the computing time grows exponentially as the problem size increases. Thus, for a large school with hundreds of students and course offerings, we might not be able to solve the ILP.

To find a close-to-optimal timetable in a reasonable amount of time, we use Large Neighbourhood Search (LNS), a well-known technique [22] to iteratively improve our solution. Given any solution to the PECTP (e.g. the initial timetable found in Stage I), we lock in all but h of the variables $X_{s,c,d,p}$, set them as hard constraints, and then re-calculate the PECTP to generate a new solution where some of these h events may be reassigned to other timeslots, and the teacher assignments $Z_{t,s,c,d,p}$ can change as well.

Like all local search algorithms, the LNS may get stuck in a local minimum, especially if the value of h is small. However, when h is sufficiently large, the results of the LNS get better at each step, and we stop when the algorithm appears to have converged.

We now apply our work on a real-world instance and we describe that in the following section.

4 Application

West Point Grey Academy (WPGA) is one of Canada's leading independent schools. Founded in 1996, WPGA is located in Vancouver, the most populous city in the Canadian province of British Columbia. Its competitive debate team is one of the most decorated teams in Canada, with five national titles since 2008, including victory at the World Schools Debate Championship in 2010.

WPGA has an enrollment of 940 students from Junior Kindergarten to Grade 12, with just over 400 students and teachers in the Senior School (Grades 8-12). The lead author has worked closely with the Senior School's academic head, building the WPGA Master Timetable for each of the past five years.

Unlike students in the lower grades who take mostly required courses in fixed cohorts, the Senior School students have numerous course options, and each student wants to enroll in a different combination of courses in the 9 timeslots.

For the 2024-2025 timetable, the $|S| = 403$ students requested a total of 3565 courses, which was fewer than the theoretical maximum of $403 \times 9 = 3627$. This occurred because the oldest (Grade 12) students could take a "self-study period", i.e., a spare block, as could some of the younger students.

Based on the student requests, the school decided to offer $|C| = 124$ different courses, of which 68 were single-section courses. Of the 56 multi-section courses, many had two or three sections, though several required courses had up to five sections. In all, there were 245 total course sections, i.e., 245 events.

Each school day consists of five periods, alternating between Day 1 (A, B, C, D, X) and Day 2 (E, F, G, H, I), repeating this pattern for the entire academic year. The final timeslot on Day 1 is a special period where no courses are scheduled, and instead students go through an advisory program that emphasizes mental well-being, social awareness and personal relationships.

Thus, we can think of the WPGA timetable as having $|D| = 2$ days, $|P| = 5$ periods, and $|D||P| - 1 = 9$ timeslots, since no event can be scheduled on Day 1 Period 5.

For each of the $|C| = 124$ courses, the academic head of the WPGA Senior School listed the set of teachers who were qualified to teach that course. For 40 of these courses, the academic head listed more than one teacher who could teach that course, thus creating the flexible structure that motivated this paper. (For the remaining 84 courses, the academic head decided that only one teacher could teach that course, and so these courses had pre-assigned teachers).

Furthermore, for each of the $|T| = 54$ teachers, the academic head listed the teaching load for that teacher, i.e., the number of events they needed to be assigned, as well as any special constraints. For example, several of the teachers were part-time and so all of their classes needed to be on either Day 1 or on Day 2. Several key teachers, known as Department Heads, needed to have a common non-teaching block in which to meet.

Finally, every teacher with a load of 4 courses had to have an even 2-2 split between the two Days, every teacher with a load of 6 courses had to have an even 3-3 split, and every teacher with a load of 5 courses had to have a 3-2 split or a 2-3 split.

The academic head set the weight for each preference coefficient $P_{i,c}$, which was a function of the student's grade as well as the type of course. Older students were given higher weights, and required courses were given more points than elective courses.

Our optimization program, written in Python, inputted an Excel sheet consisting of all the course/teacher/room data as well as the course requests from each student. To solve the ILP, we used the solver from IBM's ILOG CPLEX Optimization Studio with the Google OR-Tools wrapper [12].

Following the process presented in the previous section, we first found an initial timetable that considered all of the course/teacher/room constraints, i.e., a feasible assignment of 245 events to 9 timeslots. This process took less than half a second on our computer, an 8GB Lenovo laptop running Windows 10 with a 2.1 GHz processor.

Once we found our initial timetable, we incorporated the student course requests, following our two-stage heuristic. We applied our LNS with a threshold of $h = 60$ so that our ILP locked in $245 - h = 185$ events in its timeslot, while allowing the remaining $h = 60$ events to be re-assigned to other timeslots.

We found that randomly reshuffling $h = 60$ events was ideal for our problem size, so that each step of the search computed in an average time of 90 seconds. Our local search algorithm converged to the same solution within one hour, in every single trial we ran.

Here were our final results, with just eight missed course requests.

```
3557 out of 3565 total course requests satisfied: 99.78 percent
706 out of 711 Grade 12 course requests satisfied: 99.3 percent
695 out of 697 Grade 11 course requests satisfied: 99.71 percent
708 out of 709 Grade 10 course requests satisfied: 99.86 percent
724 out of 724 Grade 9 course requests satisfied: 100.0 percent
724 out of 724 Grade 8 course requests satisfied: 100.0 percent
```

Of the eight students who were not enrolled in a course they requested, all of them missed a low-priority elective course. Thus, it was a relatively easy fix for the school, since each of these students could select a different elective course.

We naturally wanted to know whether 8 misses was the best possible result. To verify this, we first changed all of the preference coefficients $P_{i,c}$ to equal 1 if student i selected course c, and set $P_{i,c}$ equal to 0 otherwise. We ran the ILP on just the 79 Grade 10 students, and our CPLEX solver quickly confirmed that 708 points was the optimal solution, i.e., the best we could do was one miss for the Grade 10 students.

We then ran our optimization model on the 160 students in Grades 11 and 12. Running our Python program for over twenty-four hours without any LNS threshold, the best our solver could find was a solution with 1402 out of 1408 course requests satisfied. We are extremely confident that this is the best possible result for the Grade 11 and 12 students, which means that the lower bound on misses is $6 + 1 = 7$.

Our solution, generated in less than one hour using a simple LNS local search heuristic, found a nearly-perfect timetable with only 8 misses.

To illustrate the effectiveness of flexible teacher assignments, we ran five trials where we randomly pre-assigned teachers to courses to satisfy constraints (1) to (7) of our model, fixed those assignments so they could not be changed, and then ran our Large Neighbourhood Search model to solve for $X_{s,c,d,p}$ and $Y_{i,s,c,d,p}$. Our best solution was 19 misses and our worst solution was 26 misses. While all of these trials resulted in a success rate over 99%, pre-assigning teachers to courses led to far more misses than the 8 misses in our solution.

5 Conclusion

In this paper, we presented our solution to solving the Post-Enrollment Course Timetabling Problem (PECTP) with flexible teacher assignments. We then applied our ILP model on a real-world instance at a Canadian high school, successfully enrolling the 403 students in 99.8% of their requested courses.

We recognize that our research on the PECTP is in its beginning stages, as further research will need to be conducted to assess how our model scales to larger data sets with thousands of students. For these larger data sets, what is the best value of h? And how does our approach compare to pure heuristics? These are questions for further exploration.

In conclusion, we have shown that flexible teacher assignments do make a difference, and will result in better Master Timetables that satisfy 100% of the course/teacher/room constraints while increasing the percentage of students who can get enrolled in their desired courses.

References

1. Al-Yakoob, S.M., Sherali, H.D.: Mathematical models and algorithms for a high school timetabling problem. Computers & Operations Research **61**, 56–68 (2015)
2. Ceschia, S., Di Gaspero, L., Schaerf, A.: Educational timetabling: Problems, benchmarks, and state-of-the-art results. European Journal of Operational Research **308**(1), 1–18 (2023)
3. Dorneles, Á.P., de Araújo, O.C., Buriol, L.S.: A column generation approach to high school timetabling modeled as a multicommodity flow problem. European Journal of Operational Research **256**(3), 685–695 (2017)
4. Dorneles, Á.P., De Araújo, O.C., Buriol, L.S.: A fix-and-optimize heuristic for the high school timetabling problem. Computers & Operations Research **52**, 29–38 (2014)
5. Even, A.S.S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing **5(4)**, 691–703 (1976)
6. Fonseca, G.H., Santos, H.G.: Variable neighborhood search based algorithms for high school timetabling. Computers & Operations Research **52**, 203–208 (2014)
7. Fonseca, G.H., Santos, H.G., Carrano, E.G.: Integrating matheuristics and metaheuristics for timetabling. Computers & Operations Research **74**, 108–117 (2016)
8. Fonseca, G.H., Santos, H.G., Carrano, E.G., Stidsen, T.J.: Integer programming techniques for educational timetabling. European Journal of Operational Research **262**(1), 28–39 (2017)

9. Goh, S.L., Kendall, G., Sabar, N.R.: Improved local search approaches to solve the post enrolment course timetabling problem. European Journal of Operational Research **261**(1), 17–29 (2017)
10. Goh, S.L., Kendall, G., Sabar, N.R.: Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. Journal of the Operational Research Society **70**(6), 873–888 (2019)
11. Goh, S.L., Kendall, G., Sabar, N.R., Abdullah, S.: An effective hybrid local search approach for the post enrolment course timetabling problem. Opsearch **57**, 1131–1163 (2020)
12. Google OR-Tools: Fast and portable software for combinatorial optimization. `https://developers.google.com/optimization` (2025), accessed: 2025-02-14
13. Gotlieb, C.: The construction of class-teacher timetables. IFIP congress, Amsterdam **62**, 73–77 (1963)
14. Hoshino, R., Albers, J.: Cohort-based timetabling with integer linear programming. Proceedings of the 14th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2024) pp. 276–284 (2024)
15. Hoshino, R., Fabris, I.: Optimizing student course preferences in school timetabling. Proceedings of the 17th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2020) pp. 283–299 (2020)
16. Hoshino, R., Fabris, I.: Partitioning students into cohorts during COVID-19. Proceedings of the 18th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2021) pp. 89–105 (2021)
17. Kristiansen, S., Sørensen, M., Stidsen, T.R.: Integer programming for the generalized high school timetabling problem. Journal of Scheduling **18**, 377–392 (2015)
18. McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Gaspero, L.D., Qu, R., Burke, E.K.: Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS Journal on Computing **22**(1), 120–130 (2010)
19. Minh, K.N.T.T., Thanh, N.D.T., Trang, K.T., Hue, N.T.T.: Using tabu search for solving a high school timetabling problem. Advances in intelligent information and database systems pp. 305–313 (2010)
20. Nagata, Y.: Random partial neighborhood search for the post-enrollment course timetabling problem. Computers & Operations Research **90**, 84–96 (2018)
21. Pillay, N.: A survey of school timetabling research. Annals of Operations Research **218**, 261–293 (2014)
22. Pisinger, D., Ropke, S.: Large neighborhood search. In: Handbook of metaheuristics, pp. 99–127. Springer (2018)
23. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J.H., Kyngas, J., Nurmi, C., Ranson, D.: An xml format for benchmarks in high school timetabling. Annals of Operations Research **194**, 385–397 (2012)
24. Saviniec, L., Santos, M.O., Costa, A.M., dos Santos, L.M.: Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems. European Journal of Operational Research **280**(3), 1064–1081 (2020)
25. Sørensen, M., Dahms, F.H.: A two-stage decomposition of high school timetabling applied to cases in denmark. Computers & Operations Research **43**, 36–49 (2014)
26. Sørensen, M., Kristiansen, S., Stidsen, T.R.: International timetabling competition 2011: An adaptive large neighborhood search algorithm. In: 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012). pp. 489–492 (2012)

27. Tan, J.S., Goh, S.L., Kendall, G., Sabar, N.R.: A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Systems with Applications **165**, 113943 (2021)
28. Tassopoulos, I.X., Iliopoulou, C.A., Beligiannis, G.N.: Solving the greek school timetabling problem by a mixed integer programming model. Journal of the Operational Research Society **71**(1), 117–132 (2020)
29. Zhang, D., Liu, Y., M'Hallah, R., Leung, S.C.: A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. European journal of operational research **203**(3), 550–558 (2010)