Optimizing Student Course Preferences in School Timetabling

Richard Hoshino and Irene Fabris

Quest University Canada, Squamish, British Columbia, Canada
{richard.hoshino, irene.fabris}@questu.ca

Abstract. School timetabling is a complex problem in combinatorial optimization, requiring the best possible assignment of course sections to teachers, timeslots, and classrooms. There exist standard techniques for generating a school timetable, especially in cohort-based programs where students take the same set of required courses, along with several electives. However, in small interdisciplinary institutions where there are only one or two sections of each course, and there is much diversity in course preferences among individual students, it is very difficult to create an optimal timetable that enables each student to take their desired set of courses while satisfying all of the required constraints.

In this paper, we present a two-part school timetabling algorithm that was applied to generate the optimal Master Timetable for a Canadian all-girls high school, enrolling students in 100% of their core courses and 94% of their most desired electives. We conclude the paper by explaining how this algorithm, combining graph coloring with integer linear programming, can benefit other institutions that need to consider student course preferences in their timetabling.

Keywords: School Timetabling · Post Enrollment Course Timetabling Problem · Integer Programming · Graph Coloring · Optimization.

1 Introduction

Every educational institution needs to produce a Master Timetable, listing the complete set of offered courses, along with the timeslot and classroom for each section of that course. This timetable allows teachers to know what courses they are teaching, and enables students to enroll in a subset of these courses.

As many school administrators know, creating a timetable is incredibly difficult, requiring the careful balance of numerous *requirements* (hard constraints) and *preferences* (soft constraints). When timetables are constructed by hand, the process is often 10% mathematics and 90% politics [4], leading to errors, inefficiencies, and resentment among teachers and students.

To address these concerns, scholars in Operations Research have analyzed the School Timetabling Problem (STP) ever since the 1960s [10]. Various heuristics have been applied to create timetables for schools in Argentina, Brazil, Denmark, Germany, Greece, Italy, Netherlands, South Africa, and Vietnam [23].

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 [7].

Real-life timetabling problems involve additional constraints that must be satisfied [24], further increasing the complexity of the STP. These variations include *event* constraints (e.g. Course X must be scheduled before Course Y), and *resource* constraints (e.g. scheduling only one lab-based course in any timeslot). At large universities, there are additional constraints that must be considered, such as taking into account the time students need to walk from one end of the campus to the other.

Over the past five decades, numerous algorithms have been applied to generate optimal (or nearly-optimal) timetables for STP benchmark instances. These techniques include constraint programming [6], evolutionary algorithms [26], simulated annealing [18], and tabu search [20]. The complete list of methods appears in a comprehensive survey paper published earlier this decade [23].

Given how hard the STP is, a common practice is to focus only on teacher requirements and preferences, ignoring the wishes of the students (i.e., the individuals most affected by the timetable). This assumption is made because many school programs are *cohort-based* [15], where students are divided into fixed groups and take the same sequence of courses to complete their education. At many high schools and universities, the timetabling is done via *homogeneous sectioning* [4], where students are grouped according to their interests or majors: for example, students in the Arts stream versus students in the Sciences stream.

There are obvious deficiencies to this practice, most notably in small interdisciplinary institutions where cohorts do not exist, and each student takes a unique set of courses from all departments. Many such institutions are private schools, where their revenue comes exclusively from student tuition. If students cannot enroll in their desired courses, they (or their parents) will go to a different school that will accommodate their preferences. Thus, these schools are under tremendous pressure to create a timetable that satisfies teachers *and* students. This is the motivation for the *Post-Enrollment Course Timetabling Problem (PECTP)*, an active area of research in the field of automated timetabling.

This paper proceeds as follows. In Sections 2 and 3, we define the PECTP and provide a brief literature review on related work that incorporates student course preferences in timetabling. In Sections 4 and 5, we describe our solution to the PECTP, which is a two-part algorithm that generates an optimal coloring of a weighted conflict graph for single-section courses, after which an integer linear program is solved to generate the final timetable. In Section 6, we generate the optimal Master Timetable for an interdisciplinary all-girls high school in Canada, and demonstrate the speed and quality of our two-part algorithm. And in Sections 7 and 8, we explore the strengths and limitations of our timetabling algorithm to large universities, and conclude the paper with some ideas and directions for future research.

2 Problem Definition

The standard School Timetabling Problem (STP) is an example of a constraint satisfaction problem, which asks whether there exists a feasible assignment of course sections to teachers and timeslots. To avoid confusion, we will rename timeslots as *blocks*, so that T will denote the set of teachers and B will denote the set of blocks during which the courses will take place.

The more general version of the STP is a combinatorial optimization problem, which asks for the best assignment satisfying all of the hard constraints while maximizing the preferences of the teachers being assigned their desired courses in specific blocks.

Both versions of the STP can be set up as a 0-1 integer linear program (ILP), in which each unknown variable $X_{t,c,b}$ represents whether teacher t is assigned to a section of course c in block b. The total number of variables is $n = |T||C||B|$, where $|T|$ is the number of teachers, $|C|$ is the number of offered courses, and $|B|$ is the number of blocks.

Let $D_{t,c,b}$ be the desirability of teacher t assigned to course c in block b. This coefficient will be a function of teacher t's ability and willingness to teach course c, combined with their preference for teaching that course in block b.

Then, subject to all of the hard constraints, we want to maximize

$$\sum_{t\in T}\sum_{c\in C}\sum_{b\in B} D_{t,c,b}\cdot X_{t,c,b}.$$

The Post-Enrollment Course Timetabling Problem (PECTP) was introduced just over a decade ago [17], as part of the second International Timetabling Competition. In the PECTP, points are awarded for enrolling students in any section of a desired course. For example, if there are ten different sections of Calculus 101, a student wishing to take Calculus 101 needs to be assigned to exactly one of these ten sections.

In addition to all of the constraints in the STP (e.g. no teacher can be assigned to two courses in overlapping blocks), the PECTP involves additional student-related hard constraints, such as ensuring that no student is enrolled in multiple sections of the same course.

Let $Y_{s,c,b}$ be the binary decision variable representing whether student s is assigned to a section of course c in block b, and let $P_{s,c,b}$ be the preference of student s being enrolled in course c in block b.

Then, subject to all of the hard constraints, we want to maximize

$$\sum_{t\in T}\sum_{c\in C}\sum_{b\in B} D_{t,c,b}\cdot X_{t,c,b} + \sum_{s\in S}\sum_{c\in C}\sum_{b\in B} P_{s,c,b}\cdot Y_{s,c,b}.$$

This is the most basic formulation of the STP and PECTP. There are extensions that we will not consider in this paper, such as adding a penalty function whenever student s has a class in the last block of the day or has a class in three consecutive blocks. For a full discussion and treatment of these PECTP extensions, we refer the reader to [19].

3 Related Work

The PECTP was introduced in 2007 as one of the tracks of the International Timetabling Competition (ITC). Over the past twelve years, different teams of Operations Research scholars have developed algorithms to tackle hard instances of the PECTP. The majority of these approaches rely on multi-stage heuristics.

Fonseca et al. [8] propose a three-stage hybrid solver involving graph algorithms, metaheuristics, and “matheuristics”. Nothegger et al. [22] present an iterative three-step ant colony optimization algorithm. One of the finalists [5] for ITC 2007 employs a multiphased heuristic solver based on a stochastic local search, whereas the winning team [3] applies a two-stage local search approach combining tabu search and simulated annealing.

Heuristics are advantageous for they easily compute within the strict time limit imposed by the ITC, yet they cannot guarantee the optimality of the output solution. The latter is a particular weakness of local search approaches, which lack the flexibility of moving in the space of feasible solutions and get stuck in local minima despite the large size of the search neighborhood [3].

Recent papers have made much progress. Cambazard et al. [3] find provably-optimal solutions to three of the PECTP benchmark instances, by augmenting simulated annealing with a large neighborhood search. Kristiansen et al. [15] embed exact repair methods within an adaptive large neighborhood search (ALNS) to find a feasible solution first, which they optimize by solving a mixed integer program (MIP). Their ALNS finds timetables within 1% of the optimal solution, outperforming Gurobi, a state-of-the-art MIP solver, on large instances.

Parallel to the noteworthy progress in heuristic approaches, the last decade was marked by a significant advance in general-purpose MIP solvers. An obvious advantage of Integer Programming is its ability to issue certificates of optimality [14]. Since it is NP-complete to solve a 0-1 Integer Program [13], it has become common practice to decompose IP models into smaller sub-problems [27]. Van Den Broek et al. [29] use the lexicographical optimization of four ILP sub-problems to solve a real-world instance of PECTP at a Dutch university, and Kristiansen et al. [14] devise a two-step MIP algorithm for Danish high schools.

The problem considered in this paper is most similar to the formulation of the generalized PECTP presented by Mendez-Diaz et al. [19] and Carter [4], which were inspired by university timetabling problems in Argentina and Canada, respectively. In these two papers, the researchers first assign course sections to blocks, and then assign students to course sections. On the next page, we present our mathematical model that explains how we can perform both assignments simultaneously.

We make two contributions in this paper. First, we present a complete algorithm that guarantees fast optimal PECTP solutions for small educational institutions. Secondly, we provide a graph-theoretic framework to demonstrate how courses can be “bundled” together and treated as a single super-course, which significantly reduces the time required to generate a nearly-optimal timetable. This makes our algorithm scalable for larger schools and universities.

4 Mathematical Model

Let T be the set of teachers, S be the set of students, C be the set of courses, and B be the set of blocks.

For each $t \in T, c \in C, b \in B$, let $X_{t,c,b}$ be the binary variable that equals 1 if teacher t is assigned to course c in block b, and is 0 otherwise. Similarly, for each $s \in S, c \in C, b \in B$, let $Y_{s,c,b}$ be the binary variable that equals 1 if student s is enrolled in course c in block b, and is 0 otherwise.

Earlier we defined the desirability coefficient $D_{t,c,b}$ and the preference coefficient $P_{s,c,b}$. Our Integer Linear Program (ILP) has the following objective function:

$$\sum_{t \in T} \sum_{c \in C} \sum_{b \in B} D_{t,c,b} \cdot X_{t,c,b} + \sum_{s \in S} \sum_{c \in C} \sum_{b \in B} P_{s,c,b} \cdot Y_{s,c,b}.$$

We now present our hard constraints.

No teacher can be assigned to two different classes in the same block, and at most one section of any course is offered in any given block.

$$\sum_{c \in C} X_{t,c,b} \leq 1 \qquad \forall\, t \in T,\ b \in B \tag{1}$$

$$\sum_{t \in T} X_{t,c,b} \leq 1 \qquad \forall\, c \in C,\ b \in B \tag{2}$$

Define O_c to be the number of offered sections of course c.

$$\sum_{b \in B} \sum_{t \in T} X_{t,c,b} = O_c \qquad \forall\, c \in C \tag{3}$$

No student can be enrolled in more than one course in the same block, nor can any student be enrolled in two sections of the same course.

$$\sum_{c \in C} Y_{s,c,b} \leq 1 \qquad \forall\, s \in S,\ b \in B \tag{4}$$

$$\sum_{b \in B} Y_{s,c,b} \leq 1 \qquad \forall\, s \in S,\ c \in C \tag{5}$$

No student can be enrolled in a course during a block in which that course is not offered by any teacher.

$$Y_{s,c,b} \leq \sum_{t \in T} X_{t,c,b} \qquad \forall\, s \in S,\ c \in C,\ b \in B \tag{6}$$

Let R be the number of available rooms in the school.

$$\sum_{t \in T} \sum_{c \in C} X_{t,c,b} \leq R \qquad \forall\, b \in B \tag{7}$$

Let M be the maximum size of a class.

$$\sum_{s \in S} Y_{s,c,b} \leq M \qquad \forall\, c \in C,\ b \in B \tag{8}$$

Our ILP maximizes the objective function subject to these eight constraints.

This model has a total of $(|T| + |S|) \cdot |C||B|$ binary decision variables. In practice, the large majority of these variables $X_{t,c,b}$ and $Y_{s,c,b}$ will be pre-set to 0, since teachers are qualified to only teach a small subset of the offered courses, and likewise, students will only want to be enrolled in a small subset of these courses. By fixing these zero variables, we can solve the PECTP using the above ILP, guaranteeing an optimal timetable whenever $|T|$, $|S|$, and $|C|$ are of reasonable size. But when these values are large, like at most universities, simplifications are required to ensure tractability.

There are two natural ways to simplify the problem: assume there are *cliques of teachers* or assume there are *cohorts of students*. In the former, the Master Timetable is generated one clique at a time: first assign course sections to the math teachers and fix those assignments, then do the same with the science teachers, and so on. In the latter, the students are pre-divided into fixed groups, and each group is assigned to the same set of course sections.

Unfortunately, these two approaches fail when there are many teachers who teach different subjects (e.g. Ms. X teaches Grade 12 Math and Grade 7 French), and when cohorts do not exist and students wish to take a unique combination of courses from two different faculties (e.g. an undergraduate attempting a double-major in Chemistry and Sociology).

Our approach is not to bundle teachers or bundle students, but rather to *bundle courses*. We now present a graph-theoretic approach that efficiently partitions one-section courses into discrete bundles that enable us to significantly reduce the running time of the ILP.

5 Bundling One-Section Courses

C is the set of courses. Some of these courses will be sought by many students, and so multiple sections of the course must be offered in the timetable. The rest are specialized courses that will attract only a small number of students, and so only a single section is required. Let $C = C_M \cup C_O$, where C_M is the set of multiple-section courses and C_O is the set of one-section courses.

While there is much flexibility to timetabling courses in C_M, courses in C_O can only be assigned to a single block, and so we must ensure that the courses in C_O avoid any type of scheduling conflict: by teacher, by room, or by student.

Define G to be the *weighted conflict graph*, where C_O is the set of vertices. For each pair $x, y \in C_O$, we calculate the edge weight $w(x, y)$ as follows:

(a) Add a weight of w_t if the same teacher is required to teach both x and y.
(b) Add a weight of w_r if the same room must be used for both x and y.
(c) Add a weight of w_s for *each* student who wishes to take both x and y.

The weights w_t, w_r, w_s can vary, though in practice it is most logical to set high values of w_t and w_r and low values for w_s (e.g. $w_t = 100, w_r = 100, w_s = 1$).

For each integer $i \geq 0$, define G_i to be the graph with vertex set C_O whose edge set only consists of edges with weight greater than i. By definition, there exists a sufficiently large integer i for which G_i is an empty graph with no edges.

For each G_i, the *chromatic number* $\chi(G_i)$ is the fewest number of colors needed to color the vertices of G_i so that no two vertices joined by an edge share the same color.

If $\chi(G_0)$ is at most $|B|$, the number of blocks in the timetable, then all of the one-section courses assigned the same color can be "bundled" together in the same block. This guarantees that every student will be able to take all of their desired one-section courses, since no pair will be offered at the same time. These bundles can be thought of as the "supernodes" of the conflict graph [2].

If $\chi(G_0) > |B|$, then by definition, it is impossible to create a timetable that enables every student to get into all of their desired courses. In this case, we find the smallest index t for which $\chi(G_t) \leq |B|$, and once again, the color classes correspond to our bundles for the one-section courses.

Although it is NP-complete to determine the chromatic number of a general graph [13], for many large graphs we can compute $\chi(G)$ using state-of-the-art algorithms based on local search [12]. We can also compute $\chi(G)$ by solving the corresponding 0-1 ILP, and adding the constraint that no color class can contain more than R courses, where R is the number of rooms available for teaching.

This motivates our solution to the Post-Enrollment Course Timetabling Problem, where we use graph coloring to reduce the number of variables in our ILP.

(i) Construct the weighted conflict graph G, where the vertex set is C_O.
(ii) Starting with $i = 0$, calculate $\chi(G_i)$. If $\chi(G_i) \leq |B|$, then stop. Otherwise increment i by 1 until we find some index $i = t$ for which $\chi(G_i) \leq |B|$.
(iii) Find a $|B|$-coloring of $\chi(G_i)$ where the number of one-section courses in each color class is at most the number of available rooms. Let X_j be the set of courses in C_O assigned to color j.
(iv) Redefine C to equal $C_M \cup X_1 \cup X_2 \cup \ldots \cup X_{|B|}$, where there are $|C_M|$ courses that have multiple sections, and $|B|$ bundles, each of which is a one-section "super-course" with multiple teachers that can be assigned to any number of students. We then solve the previously-defined ILP, using this new set C.

For example, suppose that there are $|C| = 120$ courses to be timetabled into $|B| = 10$ blocks, where $|C_M| = 20$ and $|C_O| = 100$. The above algorithm bundles the 100 one-section courses into $|B| = 10$ bundles. Thus, instead of considering $|C| = 120$ courses in our ILP, we now only need to consider $|C_M| + |B| = 30$ courses. By reducing the number of variables by a factor of four, we create a massive reduction in the total running time while only sacrificing a small percentage in quality, as measured by the value of our objective function.

Our approach is particularly useful in small interdisciplinary institutions that offer numerous one-section courses desired by different sets of students. We now provide an example of such an educational institution, and apply our algorithm to create the optimal Master Timetable for this all-girls independent school.

6 Application

St. Margaret's School (SMS) is located in Victoria, the capital city of the Canadian province of British Columbia. Since 1908, educators at SMS have dedicated themselves to inspiring girls who want to change the world and helping them become women who do change the world. The school has an enrollment of approximately 375 students, starting from Junior Kindergarten (age 3 and 4).

As mentioned in their online handbook [28], SMS prides itself on their small-scale learning environment, which provides teachers with the flexibility to personalize learning for each student and challenge each girl to realize her own potential. In order to achieve this goal, the school spends several months each year constructing the Master Timetable, by hand, with several dozen iterations.

The biggest challenge is timetabling the courses for the Grade 11 and 12 students, i.e., the juniors and seniors at the high school. Unlike students in the lower grades who take mostly required (core) courses, there are numerous elective courses in the final two years, and each student wants to enroll in a different combination of courses from the *eighty* offerings that are available, including Advanced Placement Calculus, Law Studies, Studio Art, and Creative Writing.

The $|S| = 58$ students going into Grade 11 and 12 completed a survey indicating their course choices for the following year. The administrators used these responses to decide to offer $|C| = 39$ of the 80 possible courses. To ensure a maximum class size of 18, the administrators assigned two sections to $|C_M| = 9$ courses requested by more than 18 students, and one section for the remaining $|C_O| = 30$ courses that were requested by at most 18 students.

There are five periods in each day, and a total of $|B| = 9$ blocks. The nine blocks are fixed in the schedule, as follows:

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Block 1	Block 2	Assembly	Block 1	Block 2
Block 7	Block 8	Block 8	Block 9	Block 3
Block 4	Block 3	Block 6	Block 5	Block 7
Block 5	Block 9	Block 4	Block 6	Block 8
Block 6	Block 7	Block 5	Block 4	Block 9

Of the $|C| = 39$ courses, 18 are "short" courses offered in blocks 1/2/3 with two weekly classes, and the other 21 are "long" courses in blocks 4/5/6/7/8/9 with three weekly classes. Each student is required to take a set of core courses, with the rest being freely-chosen electives. Most (but not all) of the core courses are long, and most (but not all) of the elective courses are short.

Each student s updated their survey with their most desired courses for 2019-2020, listing up to 3 short courses and up to 6 long courses. This selection included the core courses of English and Career/Leadership, as well as an additional English Language Learner course for non-native English speakers. Finally, Grade 11 students were required to take a Physical Education course. Thus, all Grade 11s had at most 4 core courses, while Grade 12s students had at most 3.

The $|S| = 58$ students requested a total of 447 courses, which is fewer than the maximum total of $|S| \times |B| = 58 \times 9 = 522$. This occurred because some students had no preference for certain electives (e.g. they viewed every Social Science course as interchangeable), and also because many of the students qualified to take eight courses with their ninth one being a "self-study period".

For each block b, we set the preference coefficient $P_{s,c,b}$ as follows:

(i) 10 points if c is a core course
(ii) 3 points if c is an elective course and s is a Grade 12 student
(iii) 1 point if c is an elective course and s is a Grade 11 student

The SMS leadership team pre-assigned each of the $9 + 9 + 30 = 48$ course sections to one of the $|T| = 19$ teachers in the Senior School, based on extensive consultations with each teacher. With teacher preferences pre-assigned, this reduces our ILP's objective function to maximizing student preferences. Mathematically this is equivalent to setting the desirability coefficient $D_{t,c,b}$ to equal 0 if teacher t could be assigned to course c in some block b, and -1000 otherwise.

In addition to the eight constraints we mentioned previously, we also included extra constraints requested by the school. For example, some of the teachers work part-time, and are only available to teach on Mondays, Wednesdays, and Thursdays. This forced all of their teaching blocks to be 1, 4, 5, or 6. Two courses are "two-block combination courses" (e.g. Pre-Calculus 11 and Pre-Calculus 12), and these double courses must be offered in Block 1 and Block 8.

Our optimization program, written in Python, requires two Excel sheets as input: one called "Student Data" and one called "Course Data". These two documents encapsulate all of the information described above. For the actual optimization, we use COIN-OR Branch and Cut (CBC), an open-source MIP solver, with the Google OR-Tools linear solver wrapper [9]. The final model has a total of $(|T| + |S|) \cdot |C||B| = 77 \times 39 \times 9 = 27027$ binary decision variables.

Our ILP generates the optimal timetable in 201.6 seconds on a stand-alone laptop, specifically a 8GB Lenovo running Windows 10 with a 2.1 Ghz processor. The student author has created a repository containing all of the Python code used in this paper, as well as the input files of the student course choices. The repository can be found at https://github.com/ifabrisarabellapark/Timetabling.

Here are the summary statistics.

	Grade 12	Grade 11
Total Students	25	33
Requested Core Courses	64	103
Enrolled Core Courses	64	103
Requested Elective Courses	129	151
Enrolled Elective Courses	122	141

Every teacher is assigned to their desired set of courses, with at most 18 students in any class. Our timetable enrolls students in 167 out of 167 core courses (100%) and 263 out of 280 elective courses (94%), corresponding to an objective value of $167 \times 10 + 122 \times 3 + 141 \times 1 = 2177$.

Of the $|S| = 58$ students, 41 receive all of their desired courses, while the remaining 17 receive all courses except for one elective. This provably-optimal Master Timetable, presented below, was accepted by St. Margaret's School for the 2019-2020 academic year.

Optimal Timetable for St. Margaret's School

Block 1	Block 2	Block 3
Culinary Arts 11A (DR)	Culinary Arts 11B (DR)	Comp. Prog. 11 (WF)
Core French Intro 11 (AS)	Life Education 11A (DH)	Life Education 11B (DH)
Philosophy 12 (JP)	Entrepreneurship 11 (CJ)	Spoken Language 11 (MC)
Economics 12 (SW)	Drama 11/12 (NC)	Japanese Intro 11 (MH)
Life Connections 12A (KD)	Life Connections 12B (KD)	Composition 12 (NP)
EarthSci 11/12 Combo (CJ)	Law Studies 12 (SW)	Spanish 12 (BP)
Pre-Calc 11/12 Combo (CT)	Social Justice 12 (JP)	Comp. Cultures 12 (SW)

Block 4	Block 5	Block 6
Chemistry 11A (SB)	Art Studio 11 (LH)	Physics 11 (CT)
Active Living 11 (JS)	Fitness 11 (JS)	Life Sciences 11 (DR)
AP Studio Art 12 (LH)	Japanese 12 (MH)	Pre-Calculus 11A (WF)
English Studies 12A (NP)	Pre-Calculus 12A (CT)	English Studies 12B (NP)
AP Calculus 12 (CT)	Chemistry 12A (SB)	

Block 7	Block 8	Block 9
New Media 11 (NP)	Pre-Calculus 11B (WF)	Chemistry 11B (SB)
Creative Writing 11A (CN)	Creative Writing 11B (NP)	Core French 12 (AS)
Anatomy/Physio 12 (SB)	Chemistry 12B (SB)	Human Geog 12 (LZ)
Pre-Calculus 12B (CT)	EarthSci 11/12 Combo (CJ)	Univ/Grad Prep 12 (KD)
Physics 12 (WF)	Pre-Calc 11/12 Combo (CT)	

In the above timetable, the teacher's initials appear in parentheses, and multiple-section courses are indicated - e.g. Chemistry 11A and Chemistry 11B.

Our optimal Master Timetable allocates seven Grade 11 and 12 courses in blocks 1/2/3, and four or five Grade 11 and 12 courses in blocks 4/5/6/7/8/9, resulting in a symmetric well-balanced schedule. Once these course assignments were confirmed, it was easy to manually schedule the Grade 9 and 10 courses since these students take the same set of required courses, taught by the same set of teachers.

The final Senior School timetable was delivered in May 2019, months before the start of the 2019-2020 academic year. The early deliverable enabled the school guidance counsellor to have one-on-one meetings with each of the students before they left for the summer. The seventeen Grade 11 and 12 students who were not given one of their most-desired elective courses worked with the guidance counsellor to select an alternative course in the same subject area.

Given that our ILP solved to optimality in just over three minutes, there was no need to apply the time-reducing "course-bundling algorithm" we developed prior to receiving the final data sets from the school. However, we now provide this information to illustrate the effectiveness of this approach, especially when we have more than $|T| = 19$ teachers, $|S| = 58$ students, and $|C| = 39$ courses.

We create our conflict graph G on our $|C_O| = 30$ one-section courses. For each pair of courses c_1 and c_2, we assign a weight of 100 if these two courses must be taught by the same teacher or if they must take place in the same classroom. For each student s desiring both c_1 and c_2, we assign a weight of $\min(P_{s,c_1,b}, P_{s,c_2,b})$. This conflict graph G has 30 vertices and 94 edges.

G is a two-component graph, since the set of "short" courses that must be scheduled in blocks 1/2/3 is disjoint from the set of "long" courses that must be scheduled in blocks 4/5/6/7/8/9. We determine that $\chi(G) = 5 + 7 = 12$ using GrinPy, an open source program that quickly calculates graph invariants [11].

Let G_1 be the same graph as G, except we only include edges with weight more than 1. Then G_1 becomes a graph with 30 vertices and 69 edges, with $\chi(G_1) = 3 + 6 = 9$. Following the algorithm described in Section 5, we find a 3-coloring of the short courses and a 6-coloring of the long courses.

We then take each pair of one-section courses in the same bundle (e.g. c_1 and c_2), determine the assigned teachers for those two courses (e.g. t_1 and t_2) and add the following constraint to our ILP:

$$X_{t_1,c_1,b} = X_{t_2,c_2,b} \qquad \forall\, b \in B \tag{9}$$

This constraint ensures that courses c_1 and c_2 are assigned to the same block, i.e., bundled together. Our modified ILP has only $|C_M| + 3 + 6 = 18$ courses instead of 39, since have cleverly combined one-section courses to virtually eliminate student conflicts.

This ILP is rapidly solved, requiring only 5.3 seconds of computation time. The generated timetable enrolls students in 167 out of 167 core courses (100%) and 246 out of 280 elective courses (88%), corresponding to an objective value of $167 \times 10 + 118 \times 3 + 128 \times 1 = 2152$.

Of course, the quality of the final timetable is dependent on the initial 3-coloring of the short courses and 6-coloring of the long courses. Thus, we run a simulation of 1000 trials, using the NetworkX package for graph coloring [21]. We use the built-in *greedy_color* method to randomly order the vertices and assign the first available color to each vertex. (If the greedy coloring of G_1 requires more than $\chi(G_1)$ colors, then we re-run the method until a valid coloring is attained.)

The average objective value of our 1000 trials is 2155.03, with a minimum of 2141 and a maximum of 2170. The average running time is 4.17 seconds, with a minimum of 1.4 seconds and a maximum of 8.9 seconds. Thus, for this data set, our graph-theoretic bundling algorithm reduces the total running time by 98% (from 201.6 seconds to an average of 4.17 seconds) while reducing the solution quality by just 1% (from an objective value of 2177 to an average of 2155.03).

In all 1000 trials, the generated timetable enrolls students in 167 out of 167 core courses (100%), and on average, in 249 out of 280 elective courses (89%).

The latter result is lower than our rate of 94% in our optimal ILP, yet it is only moderately lower given the substantial time improvement

Our bundling technique reduced the total running time by 98% at virtually no cost to the objective function. However, as we will see in the following section, our graph-theoretic bundling technique does not guarantee nearly-optimal solutions in all cases. We now explore the general applicability of our approach, and discuss the strengths and limitations of our methods for much more complex instances of School Timetabling.

7 Strengths and Limitations

At many large university campuses, each faculty is its own distinct entity, with their own buildings, classrooms, courses, professors, and registered students. Because there is little to no overlap between faculties (e.g. between the Business School and the Medical School), we can think of the university-wide timetabling process as solving hundreds of discrete timetabling problems, and combining these non-overlapping solutions to produce a single Master Timetable.

For example, the largest university in our province has over 50,000 undergraduate students, with nearly 20% in the Faculty of Science. Within the Faculty of Science are twelve different Departments, including botany, mathematics, and zoology. Each department chair is responsible for their own timetabling, and no effort is made to coordinate timetabling to serve the small minority of students who have multiple specializations, e.g. a mathematics and zoology double major.

This is why our two-step timetabling approach holds much promise, even for large institutions. Each department chair is responsible for timetabling a number of third-year and fourth-year courses, many of which will be single-section courses appealing only to the students pursuing a major offered by that department. If the department has thousands of registered students, generating an optimal timetable might be computationally intractable, but a nearly-optimal timetable could be generated using our graph-theoretic course bundling approach.

At St. Margaret's School, most of the teachers have their own dedicated classroom, which meant that we could ignore constraints such as "there are only 3 science labs available, and so we cannot schedule 4 lab-based courses in the same block". Fortunately, there exist ways we can ensure feasible room assignments in situations like this, by analyzing a specific partial transversal polytope [16], to complement our two-step timetabling algorithm.

We can also consider the effect of relaxing or eliminating the pre-allocation of teachers to courses. If each teacher has a ranked preference list of courses they wish to teach, and each teacher is qualified to teach dozens of courses, then this will significantly increase the running time of our algorithm. For St. Margaret's School, all courses were pre-assigned to a teacher. For a high school whose timetable we are creating now, only student preferences are to be considered in the optimization. Thus, all teacher assignments are made *after* the courses are assigned to blocks, and we will optimally assign teachers to courses to ensure everyone has a feasible schedule maximizing the total "preference score".

Despite the promising results we have found thus far, there are limitations of course bundling. As an extreme example, consider the following scenario where we wish to schedule 5 one-section courses $\{a, b, c, d, e\}$ and 2 two-section courses $\{x, y\}$ into a 3-block timetable. Suppose that the students wish to take the following set of courses, in any order:

Student	Course 1	Course 2	Course 3
S_1	a	b	e
S_2	c	d	e
S_3	a	b	x
S_4	c	d	x
S_5	a	b	y
S_6	c	d	y
S_7	a	y	e
S_8	d	y	e
S_9	b	x	e
S_{10}	c	x	e

It is straightforward to show there is only one optimal timetable that enables all ten students to enroll in a desired course in each of the three blocks.

$$\text{Block 1} = \{a, d, x\}$$
$$\text{Block 2} = \{b, c, y\}$$
$$\text{Block 3} = \{x, y, e\}$$

For example, student S_{10} can register for c, x, e in Blocks 2, 1, 3, respectively.

Let $C_O = \{a, b, c, d, e\}$ be our single-section courses. We construct the conflict graph G by adding a weight of 1 to each pair of courses desired by a student. Below is the graph G with all edges of non-zero weight.

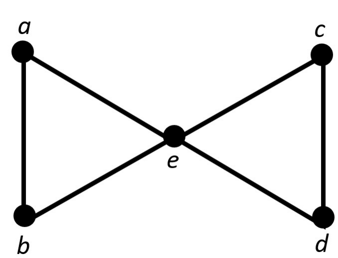

Fig. 1. The conflict graph of the single-section courses

Clearly, $\chi(G) = 3$, and there are two possible colorings up to isomorphism: the color classes are either $[\{a, d\}, \{b, c\}, \{e\}]$ or $[\{a, c\}, \{b, d\}, \{e\}]$. In the former, we bundle a and d into Block 1 and bundle b and c into Block 2. This immediately generates the optimal solution shown on the previous page.

In the latter, we bundle a and c into Block 1 and bundle b and d into Block 2. In this case, it is impossible to create a 3-block timetable that allows every student to get into all of their desired courses. The best timetable, shown below, necessitates a conflict for students S_8 and S_{10}.

$$\begin{aligned}\text{Block } 1 &= \{a, c, x\}\\ \text{Block } 2 &= \{b, d, y\}\\ \text{Block } 3 &= \{x, y, e\}\end{aligned}$$

Thus, our course bundling algorithm has a 50% chance of creating a timetable that reduces by 20% the number of students getting into all of their desired courses. While this is an extreme example, this result highlights the importance of not relying on a single coloring, since an unlucky assignment of colors to one-section courses may generate a sub-optimal Master Timetable.

However, if we run our course bundling algorithm n times on this instance, and accept the best result of all n iterations, then we have a probability of $1 - \frac{1}{2^n}$ of generating the optimal timetable.

In general, if there are $|B|$ blocks in the timetable, at least one $|B|$-coloring of the conflict graph must be identical to the optimal solution. By running our course bundling algorithm many times and selecting the best result, we increase the probability of producing the best possible timetable.

For St. Margaret's School, course bundling reduced the computation time by an average of 98% while having a minimal reduction on the quality of the generated timetable. We are optimistic that this technique can be applied to larger timetabling instances, to produce solutions to previously-intractable timetabling instances that are close to optimal.

8 Conclusion

In this paper, we presented an ILP-based model to optimally solve a real-life instance of the Post-Enrollment Course Timetabling Problem. Our Master Timetable for St. Margaret's School (SMS) enrolled Grade 11 and 12 students into 100% of their core courses and 94% of their most desired elective courses.

We also developed a "pre-processing phase" that partitioned one-section courses into discrete bundles using graph coloring. On the SMS data set, this approach decreased the total running time by 98%, with only a 1% reduction in the value of our objective function.

The collaboration with St. Margaret's School was a tremendous success, and our Master Timetable has been well-received by the school leadership, and more importantly, from their teachers and students.

During the past month, we have been approached by three different independent high schools in British Columbia, who have learned about the new "happiness-maximizing timetable" at St. Margaret's School and have requested our services to design their 2020-2021 Master Timetable. We are looking forward to these collaborations and learning ways to further improve the speed and quality of our timetabling algorithm on larger data sets.

There are many directions for future research. One natural direction is to test the performance of our two-step timetabling algorithm on benchmark instances maintained by scholars in Operations Research. Specifically, we propose testing our algorithm on XHSTT-2011, a High School Timetabling Archive containing 21 real-life instances from eight countries [25]. This archive includes an evaluator that checks syntax consistency and returns a cost value proportional to the number of violated constraints. Thus, the online evaluator allows us to compare the quality of our solution against multiple published algorithms.

To generate the best possible results, we will need to apply more sophisticated ways to color the nodes of the conflict graph beyond our approach of randomly ordering the vertices and assigning the first available color to each vertex. There exist systematic and deterministic approaches on how to color the nodes, including ordering the vertices by decreasing degree. The Python NetworkX package contains various algorithms, including one based on "degree saturation" [1], where the vertex order is determined dynamically, based on the number of colors that cannot be used because of conflicts with previously colored vertices.

Another direction is to commercialize a cloud-based application to allow school administrators to run our timetabling algorithm on their own machines. The current product requires the client to email us two Excel documents from which our Python program generates the optimal Master Timetable in the form of another Excel document. We are excited by the prospect of developing a cloud-based solution that can be deployed by the administration of high schools and universities from around the world, especially those institutions who wish to create timetables that optimize for student course preferences.

9 Acknowledgments

The authors thank the reviewers for their insightful comments that significantly improved the final version of this paper. We are grateful to Darlene DeMerchant and Megan Hedderick at St. Margaret's School for making this collaboration possible. Finally, we acknowledge that the student author's research was sponsored by a Student Project Grant awarded by the Research and Scholarly Works Committee at Quest University Canada.

References

1. Brélaz, D.: New methods to color the vertices of a graph. Communications of the ACM **22**(4), 251–6 (1979)
2. Burke, E. K., Mareček, J., Parkes, A., Rudová, H.: A supernodal formulation of vertex colouring with applications in course timetabling. Annals of Operations Research **179**(1), 105–130 (2010)
3. Cambazard, H., Hébrard, E., O'Sullivan, B., Papadopoulos, A.: Local search and constraint programming for the post enrolment-based course timetabling problem. Annals of Operations Research **194**(1), 111–135 (2012)

4. Carter, M. W.: A comprehensive course timetabling and student scheduling system at the University of Waterloo. In: Burke E., Erben W. (eds) PATAT 2000, LNCS, vol. 2079, pp. 64–82. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44629-X_5
5. Chiarandini, M., Fawcett, C., Hoos, H. H.: A modular multiphase heuristic solver for post enrolment course timetabling. In: PATAT (2008)
6. Demirović, E., Stuckey, P. J.: Constraint programming for high school timetabling: a scheduling-based model with hot starts. In: Van Hoeve. W. J. (eds) CPAIOR 2018, LNCS, vol. 10848, pp. 135–152. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93031-2_10
7. 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)
8. Fonseca, G. H., Santos, H. G., Carrano, E. G., Stidsen, T. J.: Modelling and solving university course timetabling problems through XHSTT. In: PATAT, vol. 16, pp. 127–138 (2016)
9. Google OR-Tools: Fast and portable software for combinatorial optimization, https://developers.google.com/optimization. Last accessed 17 Feb 2020
10. Gotlieb, C. C.: The construction of class-teacher timetables. In: IFIP congress, Amsterdam, vol. 62, pp. 73–77 (1963)
11. GrinPy: A NetworkX extension for calculating graph invariants, https://pypi.org/project/grinpy/. Last accessed 17 Feb 2020
12. Hébrard, E., Katsirelos, G.: A Hybrid Approach for Exact Coloring of Massive Graphs. In: Rousseau L. M., Kostas S. (eds) CPAIOR 2019, LNCS, vol. 11494, pp. 374–390, Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19212-9_25
13. Karp, R. M.: Reducibility among Combinatorial Problems. Complexity of Computer Computations, 85–103 (1972)
14. Kristiansen, S., Sørensen, M., Stidsen, T. R.: Integer programming for the generalized high school timetabling problem. Journal of Scheduling **18**(4), 377–392 (2015)
15. Kristiansen, S., Stidsen, T. R.: Elective course student sectioning at Danish high schools. Annals of Operations Research **239**(1), 99–117 (2016)
16. Lach, G., Lübbecke, M.: Optimal University Course Timetables and the Partial Transversal Polytope. In: Proceedings of the 7th international conference on Experimental Algorithms, pp. 235–248. Springer, Berlin, (2008)
17. Lewis, R., Paechter, R., McCollum, B.: Post Enrolment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition (2007)
18. Liu, Y., Zhang, D., Leung, S. C.: A simulated annealing algorithm with a new neighborhood structure for the timetabling problem. In: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation pp. 381–386 (2009)
19. Méndez-Díaz, I., Zabala, P., Miranda-Bront, J.J.: An ILP based heuristic for a generalization of the post-enrollment course timetabling problem. Computers & Operations Research **76**, 195–207 (2016)
20. 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. In: Nguyen N. T., Katarzyniak R., Chen S. M. (eds) Advances in Intelligent Information and Database Systems, vol. 283, pp. 305–313. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12090-9_26
21. NetworkX: Software for complex networks, https://networkx.github.io/. Last accessed 17 Feb 2020

22. Nothegger, C., Mayer, A., Chwatal, A., Raidl, G. R.: Solving the post enrolment course timetabling problem by ant colony optimization. Annals of Operations Research **194**(1), 325–339 (2012)
23. Pillay, N.: A Survey of School Timetabling Research. Annals of Operations Research **218**(1), 261–293 (2014)
24. 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**(1), 385–397 (2012)
25. Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., Schaerf, A.: XHSTT: an XML archive for high school timetabling problems in different countries. Annals of Operations Research **218**(1), 295–301 (2014)
26. Raghavjee, R., Pillay, N.: An application of genetic algorithms to the school timetabling problem. In: In Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology, pp. 193–199 (2008)
27. Sørensen, M., Dahms, F. H. W.: A Two-Stage Decomposition of High School Timetabling Applied to Cases in Denmark. Computers & Operations Research **1**(43), 36–49 (2014)
28. St. Margaret's Boarding School Viewbook, https://www.stmarg.ca/wp-content/uploads/SMS-Boarding-Viewbook.pdf. Last accessed 17 Feb 2020
29. Van Den Broek, J., Hurkens, C., Woeginger, G.: Timetabling problems at the TU Eindhoven. European Journal of Operational Research **196**(3), 877–885 (2009)