

A Unified Model for Course Scheduling Under Sudden Scarcity: Applications to Pandemic Planning

Cynthia Barnhart

Civil and Environmental Engineering, Massachusetts Institute of Technology, cbarnhar@mit.edu

Dimitris Bertsimas

Sloan School of Management, Massachusetts Institute of Technology, dbertsim@mit.edu

Arthur Delarue

Operations Research Center, Massachusetts Institute of Technology, adelarue@mit.edu

Julia Yan

Operations Research Center, Massachusetts Institute of Technology, jyyan@mit.edu

Problem Definition: Physical distancing requirements during the COVID-19 pandemic have dramatically reduced the effective capacity of university campuses. In response, educational strategies have been reevaluated to make the most of newly-scarce resources. Under these conditions, we examine the related problems of curriculum planning and course timetabling.

Academic / Practical Relevance: We propose a unified model for university course scheduling problems under a two-stage framework, and draw parallels between component problems while showing how to accommodate individual specifics. During the pandemic, our models were critical to measuring the impact of several innovative proposals, including expanding the academic calendar, teaching across multiple rooms, and rotating student attendance through the week and school year.

Methodology: We use integer optimization combined with enrollment data from thousands of past students. Our models scale to thousands of individual students enrolled in hundreds of courses.

Results: We projected that if MIT moved from its usual two-semester calendar to a three-semester calendar, over 90% of students' courses could be satisfied on campus even with each student only on campus for two out of three semesters and without increasing faculty workloads. For the Sloan School of Management, we produced a new schedule which was implemented in Fall 2020. The schedule adhered to safety guidelines while allowing for half of Sloan courses to include a physically-distanced in-person component, and affording over two thirds of Sloan students the opportunity for safe in-person learning in at least half their classes, despite a fourfold reduction in classroom capacity.

Managerial Implications: Integer optimization can enable decision-making at a large scale in a domain that is usually managed manually by university administrators. Our models, although inspired by the pandemic, are generic and could apply to any scheduling problem under severe capacity constraints.

Key words: scheduling, optimization, education, policy modeling

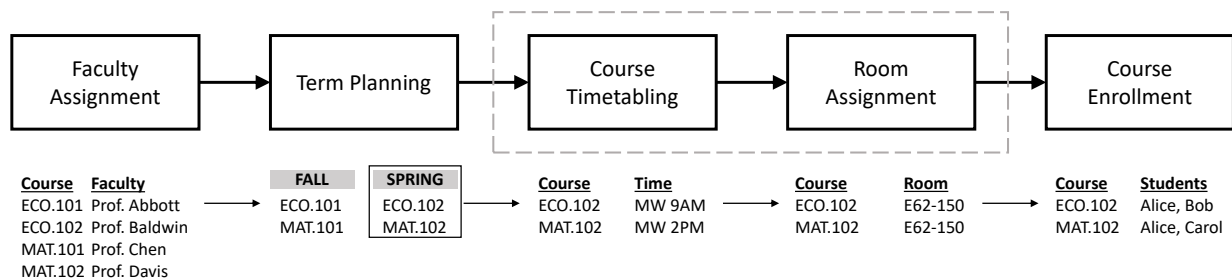
1. Introduction

In December 2019, a new coronavirus named SARS-CoV-2 was identified in Wuhan, China, and quickly spread across the globe. By March 11, 2020, the World Health Organization declared the outbreak a pandemic. Governments took drastic steps to slow the spread of the virus. Having quickly determined that dense campuses were fertile ground for an outbreak, universities moved spring courses online and sent students home. Physical distancing, the practice of keeping a safe amount of physical space with others to reduce the risk of infection, became widely adopted.

While the decision to close campuses was comparatively quick, setting policy around when and how to open campuses has been challenging. There is a fundamental tension between universities' mission of providing high-quality education, and the civic imperative to ensure safety on campus. Universities cannot operate at the same levels of campus density during a pandemic, as physical distancing rules dramatically reduce the effective capacity of classrooms, dorms, and labs. This reduced-capacity environment poses new operational challenges, with the essential question being whether universities can safely and responsibly provide in-person teaching.

Many universities explored creative ways of offering hybrid in-person and online modes of education, while also reducing campus density. Brown University (2020), Stanford University (2020) and the Massachusetts Institute of Technology (2020) all considered increasing the number of terms in the academic year, allowing students to stagger their residencies while preserving their on-campus experience. Stanford would invite each student on campus for two out of four quarters, and Brown and MIT for two out of three semesters. This policy is not without pre-pandemic precedent. For instance, when Dartmouth University first began admitting female undergraduates, the number of male admissions was not correspondingly decreased. In order to reduce strain on campus capacity, a summer term was added and students spent time both on- and off-campus (Farber 1971).

In developing universities' pandemic response, optimization can ensure that newly-scarce resources are used efficiently, particularly when scheduling courses, which includes both deciding in which terms each course is offered (which we call *term planning*) and in which times and rooms it will take place (which we call *course timetabling*). In cases where universities only allow a fraction of undergraduate students on campus at a time, offering courses in different terms may be necessary to ensure that students' time on campus coincides with the offering of key in-person courses such as labs or performance courses. And when students are on campus, safety considerations compel a complete re-working of teaching pedagogy and course scheduling; new hybrid teaching modes must be offered for students or faculty who cannot attend or choose not to attend class in person, courses with in-person instruction must be held in spaces that allow for physical distancing, and cohorts of students must be separated to whatever extent possible.

Figure 1 A simplified hierarchy of university course scheduling problems.

Note. Course timetabling and room assignment are often performed jointly (dashed box). This flowchart is a simplification and does not capture all the nuances of each university's procedures, which are often more complicated in practice. Feedback from stakeholders at each stage can influence later stages or prompt changes at earlier stages.

At MIT, these considerations compelled the authors and many community members to model and analyze creative policy solutions to the problem of operating the university safely and effectively. The disruptive nature of the pandemic meant that much of MIT's existing operations had to be re-built from scratch, in contrast to the incremental changes that would be made in a typical year. Without the use of optimization models, this re-building process would have been impossible due to its scale and complexity. Our paper discusses the methodology, results, and takeaways from pandemic planning at MIT, and will focus on term planning at the institute level as well as course timetabling at MIT's Sloan School of Management. Although this paper was motivated by COVID-19 planning at MIT, the methodology is generic, and could be applied to scheduling in any resource-constrained setting.

1.1. Related work

Scheduling courses in a university setting requires several operational decisions. The set of courses to be offered does not change much from year to year, but the university must still decide who will teach each course, and in which term (term planning). Then, given the course listings for a particular term are, the university must choose when and where to offer each course (timetabling and room assignment). Once a schedule is finalized, students may enroll in classes. A good schedule should enable students to attend all the courses they need or want to take, without conflicts. We summarize these related problems in Figure 1. Course scheduling problems are well-studied in the literature due to their scale, complexity, and variety of application-specific details. Johnes (2015) provides a comprehensive overview of operations research applications in educational settings.

Course timetabling and room assignment. A majority of the course scheduling literature focuses on course timetabling and room assignment, the two problems with the largest decision space and thus with the most to gain from an automated approach. Traditionally, timetabling problems are divided by application area, with the recent review by Pillay (2014) differentiating between school

and university timetabling, and further between regular events like courses and one-off events like exams (Burke and Bykov 2016, García-Sánchez et al. 2019). In university course timetabling, approaches typically triangulate between a *curriculum-based* view, in which groups of courses (curricula) that must be taken together cannot be offered in conflicting times, and a *post-enrollment* view, in which individual students’ courses of interest are taken into account (Bettinelli et al. 2015).

Modeling individual students is computationally intensive and requires a data management system to keep track of each student. As a result, early work on course timetabling and room assignment, including Mulvey (1982) and Tripathy (1984), focuses on developing optimization-based algorithms to the more tractable curriculum-based approach. Defining sets of courses that should not conflict scales well and can model a variety of problem specifics, and as a result has been used in a variety of real-world applications (Hinkin and Thompson 2002, Daskalaki et al. 2004, Strichman 2017). Many methods have been developed to enhance tractability, from stronger integer programming formulation via cuts (Da Fonseca et al. 2017) to heuristics such as simulated annealing (Abramson 1991) and local search (Da Fonseca et al. 2016).

However, the modeling power of the curriculum-based approach can be limited, for instance in handling parallel sections of one course, where scheduling conflicts remain acceptable as long as students have the opportunity to attend at least one section. The curriculum-based approach is ill-suited here, though Kannan et al. (2012) propose a clever sampling approach to tackle it.

A more general approach is post-enrollment timetabling, in which individual students’ enrollment is modeled explicitly. Ceschia et al. (2012) note that this approach requires more pre- and post-processing of the solution, and that exact optimization approaches typically do not scale (motivating the use of metaheuristics). In this spirit, Méndez-Díaz et al. (2016) suggest an integer optimization formulation that is heuristically decomposed to produce high-quality solutions. Recently, Gonzalez et al. (2018) explain that advances in integer programming solvers such as CPLEX have made exact approaches possible, especially with a good warm start. We note that some timetabling approaches forego modeling students or curricula entirely: Dinkel et al. (1989) model time-of-day preferences so that classes which should not conflict have different preferences.

Finally, some works consider timetabling independently from room assignment, either because rooms are pre-assigned (Birbas et al. 2009, Dorneles et al. 2017) or because they are not scarce (Boland et al. 2008). Focusing on timetabling alone can lead to stronger formulations, as in (Santos et al. 2012), or enable methods to take advantage of additional structure: Dorneles et al. (2017) exploit an underlying multi-commodity flow problem, while Boland et al. (2008) first cluster courses into *blocks* that can be co-scheduled, then assign all courses in a block to the same time. Conversely, Carter and Tovey (1992) and Phillips et al. (2015) exclusively focus on room assignment once the timetable has been fixed, capturing further nuances of the problem such as teacher room

preferences. Even in studies which consider both timetabling and room assignment, decomposing the problem is often a useful technique to enhance tractability (Sorensen and Dahms 2014).

Beyond timetabling. Some attention has been devoted in recent years to non-timetabling course scheduling problems. Under the term planning umbrella in Figure 1, Chiarandini et al. (2012) and Ceschia et al. (2014) investigate the Balanced Academic Curriculum Problem, which aims to assign courses to terms such that students can complete their degrees on time without overloading any terms. The question of assigning students to courses, especially capacity-constrained ones, has sparked interesting work in mechanism design (Budish et al. 2017, Atef Yekta and Day 2020). Finally, both Lindahl et al. (2018) and Garcia (2019) highlight the impact of automated scheduling algorithms on strategic decision-making, including capacity expansion in both time and space.

1.2. Contributions

This paper presents novel methodology to address real problems faced by universities experiencing a sudden scarcity of campus capacity, e.g., due to physical distancing needs during the COVID-19 pandemic. Our methodological contributions are as follows:

- **A unifying framework.** We provide a unifying framework that encompasses term planning, timetabling, and room assignment, and show that these problems share a single two-stage structure. To our knowledge, this paper is the first to highlight similarities between the seemingly disparate problems of timetabling and term planning, and to propose a single formulation that unites them under the umbrella of *course scheduling problems*.
- **A flexible model.** We formulate the term planning and timetabling problems using mixed-integer optimization. This choice of methodology enables us to capture a host of both problem specifics both usual (for example, parallel sections, contiguity in time and space, pre- and co-requisites) and novel (for example, limits on campus density, hybrid teaching modes). We are able to address the vast majority of constraints explored in the literature for both curriculum-based and post-enrollment timetabling problems.
- **A scalable approach.** Our optimization approach scales to thousands of individual students and hundreds of courses without resorting to metaheuristic approaches. We leverage significant improvements in modern optimization solvers to solve the problem efficiently. Fast solve times are a necessity in policy making, as they enable multiple rounds of community feedback.

Our models were used to support decision-making at MIT in the midst of the COVID-19 pandemic, and address a variety of the challenges of operating a reduced-density campus with hybrid teaching modes. In particular, we evaluated the following new policies:

- **From two to three semesters.** We showed that, counterintuitively, existing redundancy in MIT’s schedule could be leveraged to stagger students’ presence on campus across an expanded

academic calendar, without impeding students’ degree progress or increasing faculty workload. Although the three-semester academic calendar compared favorably to the status quo, the community ultimately opted for a less ambitious path forward given the significant disruption students and faculty had already experienced.

- **Partitioning of class years.** In keeping with public health guidelines, separating student cohorts from different years and programs is desirable because it reduces unnecessary contact between students. At the Sloan School of Management, we demonstrated that a moderate increase in teaching staff would allow first- and second-year Master of Business Administration (MBA) students to attend classes on-campus on different days of the week. Following our recommendation, the school increased the number of first-year MBA cohorts from six to nine.
- **Hybrid teaching modes.** After courses moved fully online at MIT Sloan in the spring of 2019, students and faculty felt that a virtual classroom did not provide the same educational value as a traditional one. In response to this feedback, we constructed a schedule which was implemented at the Sloan School of Management in the Fall 2020 semester, incorporating hybrid in-person and online instruction while adhering to safety standards. Despite a fourfold reduction in classroom capacity, the schedule allows for half of Sloan courses to include a physically-distanced in-person component, and grants two out of every three students the opportunity for in-person learning in at least half of their courses.

Section 2 describes our modeling approach for course scheduling. Section 3 then applies our models to a case study at MIT and discusses the policy recommendations introduced above.

2. Methods

In this section, we present our optimization models for course scheduling, beginning with an example to highlight how we model the problem’s main structure, then describing detailed formulations for both term planning and course timetabling.

2.1. Motivating example

Although the two problems studied in this work, term planning and course timetabling, differ in many ways, they do share a fundamental two-stage structure. First, courses are scheduled, then students decide which to take. An ideal course schedule (first-stage decision) should allow students to take the courses they want to take, without conflicts (second-stage decision).

More formally, consider a simplified version of the term planning/timetabling problem, in which each course c in a set of courses \mathcal{C} needs to be assigned a time $t \in \mathcal{T}$, where \mathcal{T} designates the discrete set of all possible times the course can be scheduled. Then, the key first-stage decision variables can be written as $x_{c,t} \in \{0, 1\}$, where $x_{c,t} = 1$ if course c is held in slot t , and 0, otherwise. Meanwhile, we assume that for each student $s \in \mathcal{S}$, we know the set of courses $\mathcal{C}(s) \subseteq \mathcal{C}$ that they

want to take — we refer to a course $c \in \mathcal{C}(s)$ as a *required course* for student s . It is then natural to define the binary second-stage variables $z_{s,c,t}$, where $z_{s,c,t} = 1$ if student s takes course c at time t , and 0, otherwise. The core problem we seek to solve can thus be written generally as:

$$\max \quad f(\mathbf{x}, \mathbf{z}) \tag{1a}$$

$$\text{s.t.} \quad z_{s,c,t} \leq x_{c,t} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), t \in \mathcal{T}. \tag{1b}$$

$$\mathbf{x} \in \mathcal{X} \tag{1c}$$

$$\mathbf{z} \in \mathcal{Z} \tag{1d}$$

The function $f(\cdot)$ quantifies the benefit of a particular schedule. For instance $f(\mathbf{x}, \mathbf{z}) = \sum_{s \in \mathcal{S}, c \in \mathcal{C}(s), t \in \mathcal{T}} z_{s,c,t}$ scores the schedule by granting one point for each course that a student both wants and is able to take. The set \mathcal{X} represents all course-related constraints: for example, enforcing that two courses with the same instructor are not held concurrently, restricting certain courses to certain times, etc. Meanwhile, the set \mathcal{Z} represents all student-related constraints: enforcing pre- or co-requisites, classroom capacities, schedule conflicts, etc. Note that both \mathcal{X} and \mathcal{Z} may be defined using auxiliary variables if convenient. The constraints (1b) relate the first and second stages, ensuring that a student can only take a course when it is offered.

The two-stage structure presented here encompasses much of the course scheduling literature. Curriculum-based timetabling models do not include the \mathbf{z} variables or related constraints, while post-enrollment models may include no or few constraints of type (1c). In practice, it may be necessary to adopt a hybrid approach, modeling scheduling constraints using both historical student enrollments and curriculum information. More interestingly, formulation (1) can be used for timetabling problems, but also for other course scheduling problems such as term planning. To the best of our knowledge, this work is the first to connect these seemingly disparate tasks.

The major advantage of our general model is its flexibility: indeed, we will see it is well-suited to tackle both the term planning and the timetabling problems, and accommodate a wide range of special-purpose constraints. Its main disadvantage is of course that it includes decision variables for each student, required course and possible time, which can lead to scaling issues. However, we will discuss exact methods and heuristics that significantly alleviate this problem. Note also that the two-stage structure assumes that (a) students' preferences do not change, and (b) students are perfectly rational in selecting their schedule (i.e., if a course they want to take does not produce schedule conflicts, they will take it). While these assumptions can be questioned for an individual student, they are reasonable in aggregate. Indeed, given data on the courses students have historically chosen to take, it can be reasonable to circumvent students' thought process on choosing courses and focus on its outcome. We discuss the data we use in more detail in Section 3.1.

Perhaps the biggest limitation is the assumption that students do not change their required courses based on when they are offered: at the term level, by not simultaneously taking two time-consuming lab courses, and at the weekly level, by prioritizing sleeping in over early-morning or late-evening lectures. Presumably this assumption is more valid for requirements, without which students cannot progress towards their degree, as compared to elective courses.

Modeling course scheduling at a student level may seem like an unnecessary profusion of decision variables. However, it allows us to capture important aspects of the problem: for instance, (a) some courses are offered in multiple *sections*, with each student only attending lecture with one section, (b) some students may need to take certain pre- or co-requisites for a particular course, and (c) in a COVID-19 limited-capacity setting, it is of primary interest to know the total number of students on campus at any time, as total building occupancies may be limited by local authorities.

Having presented the basic structure of our optimization model, we discuss in Section 2.2 how it applies to the term planning problem. We then discuss the timetabling problem in Section 2.3. The parameters for both problems are listed in Table 1 for reference.

2.2. Term planning to reduce student density on campus

The term planning problem is motivated by academic calendar and capacity disruptions resulting from the COVID-19 pandemic, which led universities to explore the idea of increasing the number of terms in which courses would be taught during the academic year. For students, required courseloads would remain as before, while the number of terms spent on campus would be kept constant or reduced. The overall intention of this policy was to spread students out over the course of the academic year, thereby reducing campus occupancy and mitigating the spread of disease.

Typically, universities already possess a feasible course schedule, with the number of terms dictated by the existing academic calendar; for example, MIT teaches on a two-semester schedule, while Stanford teaches on a three-quarter schedule. The COVID-19 pandemic led both universities to consider an expansion of the academic calendar. MIT explored the idea of adding a winter semester in addition to the usual (but rescheduled) fall and spring semesters, with each student on campus for two of the three semesters. Similarly, Stanford announced that they would move from a three-quarter year to a four-quarter year, with each student on campus for two of the four quarters. It is highly unusual for a university to revisit the number of terms during which courses are offered, and the complexity and scale of the considered disruption makes a scheduling algorithm a necessity. In this section, the formulations generalize to any number of terms, but our discussion will focus on the case that we explored with MIT, which involved going from two to three semesters.

In the term planning problem, we are given a set of courses \mathcal{C} , and a set of terms \mathcal{T} (e.g. semesters, trimesters, quarters, etc.) and we must decide the term(s) in which each course will be taught. In

Table 1 Summary of the notation.

Symbol	Description
General school operations	
\mathcal{C}	The set of all courses offered by the university
\mathcal{S}	The set of all students enrolled at the university
$\mathcal{C}(s)$	The subset of courses that student $s \in \mathcal{S}$ needs to take
$\mathcal{S}(c)$	The subset of students for whom c is a required course.
Term planning	
\mathcal{T}	The set of terms that the university will be open; for example, {Fall, Spring, Summer}.
$\text{pre}(c)$	The set of prerequisites of course $c \in \mathcal{C}$
K	The maximum number of courses a student should take in one term
B_c	The number of terms that course c is offered in a typical year (e.g. 1, 2, 3)
Y_t	The maximum number of students to be on campus in term $t \in \mathcal{T}$
T_s^{\min}	The minimum number of terms student s is allowed on campus
T_s^{\max}	The maximum number of terms student s is allowed on campus
Timetabling	
<i>Course structure</i>	
$\mathcal{J}(c)$	The set of sections that enrollment for course $c \in \mathcal{C}$ is divided into
\mathcal{J}	The set of all sections, i.e., $\mathcal{J} := \cup_{c \in \mathcal{C}} \mathcal{J}(c)$
$\mathcal{L}(j)$	The set of lessons that students in section $j \in \mathcal{J}(c)$ must attend for course $c \in \mathcal{C}$; e.g., {Lecture, Recitation}
$\mathcal{L}(s)$	The set of lessons that correspond to courses that student s is enrolled in, i.e., $\mathcal{L}(s) := \cup_{c \in \mathcal{C}(s)} \cup_{j \in \mathcal{J}(c)} \mathcal{L}(j)$
\mathcal{L}	The set of all lessons, i.e., $\mathcal{L} := \cup_{j \in \mathcal{J}} \cup_{\ell \in \mathcal{L}(j)} \mathcal{L}(j)$
\mathcal{F}	The set of all faculty
$\mathcal{F}(\ell)$	The subset of faculty that teach lesson $\ell \in \mathcal{L}$
<i>Time resources</i>	
\mathcal{D}	The set of days that courses can be taught; for example, {M, T, W, R, F}
\mathcal{P}	The set of all periods when courses can be taught; for example, {8am-9:30am, 9:30am-11am, ..., 8pm-9:30pm}
$\mathcal{P}(d)$	The set of periods when courses can be taught on day $d \in \mathcal{D}$
M_ℓ	The <i>multiplicity</i> , or number of days lesson $\ell \in \mathcal{L}$ repeats each week
N_ℓ	The <i>length</i> in periods that lesson $\ell \in \mathcal{L}$ consumes
<i>Space resources</i>	
\mathcal{R}	The set of all rooms
\mathcal{B}	The set of all room blocks, where each block consists of one or more rooms
$\mathcal{B}(\ell)$	The set of room blocks that lesson $\ell \in \mathcal{L}$ can be held in
$\mathcal{R}(b)$	The set of rooms that are in room block b
$\mathcal{B}(r)$	The set of blocks that use room r
Q_r	The number of seats in room r
Y_d	The maximum number of students to be on campus on day $d \in \mathcal{D}$

the second stage, we assume that students can only be on campus for a subset of all terms (e.g. two out of three terms) in order to reduce overall campus occupancy. Students can take courses whether or not they are on campus. However, since many courses include in-person components, team projects, or other learning experiences that benefit from the campus environment, it is preferable for students to be on campus while their required courses are offered.

Decision variables. Similar to the general formulation presented in Section 2.1, the key decision variables for the term planning problem include:

$$x_{c,t} = \begin{cases} 1, & \text{if course } c \in \mathcal{C} \text{ is offered in term } t \in \mathcal{T}, \\ 0, & \text{otherwise;} \end{cases} \quad (2a)$$

$$z_{s,c,t}^{\text{on}} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ takes course } c \in \mathcal{C}(s) \text{ on campus in term } t \in \mathcal{T} \\ 0, & \text{otherwise;} \end{cases} \quad (2b)$$

$$z_{s,c,t}^{\text{any}} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ takes course } c \in \mathcal{C}(s) \text{ in any mode in term } t \in \mathcal{T} \\ 0, & \text{otherwise;} \end{cases} \quad (2c)$$

$$y_{s,t} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ is on-campus in term } t \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases} \quad (2d)$$

Note that the z variables are duplicated into z^{on} and z^{any} , since we want the objective to differentiate courses taken while on campus (which can include an in-person component) to courses taken while off campus (which cannot include an in-person component). In addition, we define the auxiliary variables y to capture key campus occupancy constraints.

As with example (1), the constraints of the term planning problem can be split into three categories: relational constraints (analogous to constraint (1b)), course-related constraints (analogous to constraint (1c)), and student-related constraints (analogous to constraint (1d)). We first detail relational and course-related constraints, then discuss student constraints.

Relational and course constraints. The relational and course constraints are as follows.

- A student can only take a course in a particular term if it is offered in that term (analogous to relational constraint (1b)):

$$z_{s,c,t}^{\text{any}} \leq x_{c,t} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), t \in \mathcal{T}. \quad (2e)$$

- Each course is offered as many times as it would be offered in a typical year:

$$\sum_{t \in \mathcal{T}} x_{c,t} = B_c \quad \forall c \in \mathcal{C}, \quad (2f)$$

where B_c designates the number of offerings of course c in a normal school year. This constraint ensures that faculty teaching loads are shifted, not increased.

Student constraints. We now describe the constraints modeling student effects.

- A student can take a course at most once in the coming year:

$$\sum_{t \in \mathcal{T}} z_{s,c,t}^{\text{any}} \leq 1 \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s). \quad (2g)$$

- A student can only take a course if they have previously fulfilled all pre-requisites:

$$z_{s,c,t}^{\text{any}} \leq \sum_{t' \in \mathcal{T}: t' < t} z_{s,c',t'}^{\text{any}} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), c' \in \text{pre}(c), t \in \mathcal{T}. \quad (2h)$$

- A student cannot take too many courses in each term:

$$\sum_{c \in \mathcal{C}(s)} z_{s,c,t}^{\text{any}} \leq K \quad \forall s \in \mathcal{S}, t \in \mathcal{T}. \quad (2i)$$

- A student can only take a course on campus if taking it in any mode is feasible:

$$z_{s,c,t}^{\text{on}} \leq z_{s,c,t}^{\text{any}} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), t \in \mathcal{T}. \quad (2j)$$

- A student can only take a course on campus in a particular term if they are on campus:

$$z_{s,c,t}^{\text{on}} \leq y_{s,t} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), t \in \mathcal{T}. \quad (2k)$$

- The total campus density is capped each term:

$$\sum_{s \in \mathcal{S}} y_{s,t} \leq Y_t \quad \forall t \in \mathcal{T}. \quad (2l)$$

- Each student has an appropriate number of terms on campus:

$$T_s^{\min} \leq \sum_{t \in \mathcal{T}} y_{s,t} \leq T_s^{\max} \quad \forall s \in \mathcal{S}. \quad (2m)$$

We assume students are rational and knowledgeable of pre-requisites, i.e., $c' \in \mathcal{C}(s)$ if $c' \in \text{pre}(c)$, $c \in \mathcal{C}(s)$, and student s has not taken course c' in a previous year.

Given the constraints described above, the objective (2n) is to maximize the number of courses that students are able to take, with a preference for on-campus experiences:

$$\max_{\mathbf{x}, \mathbf{y}, \mathbf{z}^{\text{on}}, \mathbf{z}^{\text{any}}} \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}(s)} \sum_{t \in \mathcal{T}} (z_{s,c,t}^{\text{on}} + \lambda z_{s,c,t}^{\text{any}}). \quad (2n)$$

The non-negative parameter λ is used to prioritize on-campus experiences, with on-campus experiences weighted by $1 + \lambda$ and off-campus experiences weighted by λ .

Formulation (2) summarizes the basic components of the term planning problem, but can accommodate a variety of additional constraints and variations. Some examples include:

- **Closeness to previous schedule.** Increasing the number of terms in the academic calendar is a major disruption, and it may be desirable to mitigate this disruption. For example, if a fall-spring calendar is transitioned to a fall-winter-spring calendar, it may be desirable for fall-only courses to stay in either fall or winter, and similarly for spring-only courses. For any such course $c \in \mathcal{C}$, we can simply add the constraint $x_{c,t} = 0$, where t designates the spring term for fall-only courses, and the fall term for spring-only courses.
- **Faculty teaching preferences.** If a faculty member typically teaches a fall course $c \in \mathcal{C}$ and a spring course $c' \in \mathcal{C}, c' \neq c$, it may not be desirable for both c and c' to be moved to the winter term. To prevent this case, we can add the constraint $x_{c,\text{WINTER}} + x_{c',\text{WINTER}} \leq 1$.
- **Increased teaching loads.** Constraint (2f) is written as a hard constraint. However, in practice, university administrators might want to explore whether increasing teaching loads might improve the student experience, particularly if it is hard to satisfy constraint (2m) regarding the minimum and maximum number of terms on campus allowed. In this case, we could change constraint (2f) to $\sum_{t \in \mathcal{T}} x_{c,t} \geq B_c$, and add the expression $\sum_{c \in \mathcal{C}} (B_c - \sum_{t \in \mathcal{T}} x_{c,t})$

to the objective with an appropriate penalty. Recalling that Formulation (2) is a maximization problem, this change would have the effect of penalizing any additional offerings of courses beyond what is offered in a normal year.

- **Co-requisites.** Constraint (2h) enforces that a student cannot take a course unless they have satisfied all prerequisites. Some courses have *co-requisites*, i.e., other courses that need to be taken either before or concurrently with the course in question. We can modify the constraint to hold for any co-requisite c' by summing over $t' \leq t$ instead of $t' < t$.

2.3. Timetabling to manage on-campus space

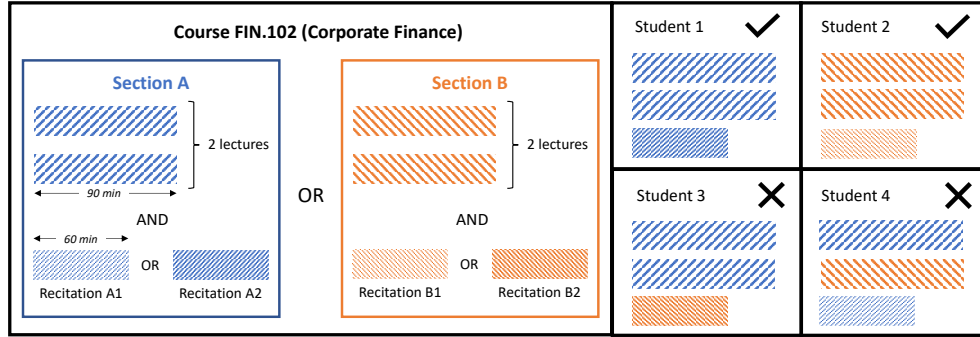
We now discuss the timetabling problem. As in the term planning problem, the goal is to schedule courses $c \in \mathcal{C}$ so that each student s can take as many required courses in $\mathcal{C}(s)$ as possible without conflicts. However, the timetabling problem adds several layers of complexity.

First and foremost, each course involves multiple in-person and/or online components: for instance, a typical MIT Sloan course involves two 90-minute lectures taught by faculty, and one 60-minute recitation led by a Teaching Assistant (TA). In order to develop a general model, we define a *lesson* to be a (possibly repeating) uninterrupted period of time devoted to teaching a particular course. A lesson ℓ is associated with a *length* N_ℓ , corresponding to the integer number of 30-minute time periods it encompasses, and a *multiplicity* M_ℓ , corresponding to the number of times the lesson is repeated weekly. Our typical MIT Sloan course would thus encompass two lessons ℓ_1 and ℓ_2 , with lengths $N_{\ell_1} = 3$ and $N_{\ell_2} = 2$ and multiplicities $M_{\ell_1} = 2$ and $M_{\ell_2} = 1$. We informally refer to a lesson ℓ with multiplicity $M_\ell \geq 2$ as a *repeating lesson*.

The notion of lesson multiplicity may seem arbitrary in this example. Why not simply decompose the course into three lessons, two of length 3 (90 minutes) and one of length 2 (60 minutes)? First, there is a greater conceptual difference between the lecture and the recitation, which differ by length and instructor as well as time of offering, than between the two lectures, which only differ in time of offering. In addition, this framework allows us to impose structure between lessons in our formulation; for example, it is preferable for lectures taught by the same faculty to occur at the same time and in the same room, while the same does not hold true for recitation.

An additional complication is the introduction of *sections*. Some popular courses are taught in multiple parallel tracks, that might or might not be scheduled concurrently or taught by the same faculty. We define a *section* as a set of lessons that together fulfill all teaching components of a particular course c . One can conceptualize a course with two sections as two parallel, identical courses. Students are indifferent between sections, but once they select one, they can only attend the lessons for that section. We offer further details on the course-section-lesson hierarchy in Fig. 2.

As mentioned before, one effect of physical distancing guidelines was the dramatic reduction of many classroom capacities. Many rooms which would normally seat over 80 students would now fit

Figure 2 Example diagram of the course-section-lesson hierarchy for a fictitious course.

Note. FIN.102 (Corporate Finance) is offered in two sections, labeled A and B, each comprising two 90-minute lectures, and two 60-minute recitations (of which students attend at most one). Example student schedules (right panel) show that students should only attend lessons from the same section (like students 1 and 2) rather than “mix and match” (like students 3 and 4). In our nomenclature, each section comprises three *lessons*: one for the 90-minute lectures (with a multiplicity of 2) and one for each recitation.

no more than 30 — plenty of space for courses with small enrollments, but not enough for medium to large enrollments. In response, MIT Sloan decided to experiment with simultaneous teaching in two neighboring classrooms, with the lecturer either present in one room and projected in the other or addressing both rooms remotely, effectively doubling the maximum capacity allowed for many lectures. From a modeling perspective, we therefore introduce the notion of a room *block*.

Formally, in addition to the set of all rooms \mathcal{R} , we introduce a set of blocks \mathcal{B} , where each block b corresponds to a set of one or two rooms $\mathcal{R}(b) \subseteq \mathcal{R}$. We note that a classroom can be part of multiple blocks, so we need to take care that we do not simultaneously use two blocks that share a classroom. In a slight abuse of notation, we write $\mathcal{B}(r)$ as the set of blocks that include classroom r , and write $\mathcal{B}(\ell)$ as the set of blocks in which lesson ℓ can be held.

Our model also introduces a new hybrid teaching system to reduce campus capacity, called *rotation*. In a rotating course, each student attends only some of lessons on campus, and attends the rest online. For instance, a course that meets two days a week might be attended by half of its students each day. Although it is possible to have students rotate at arbitrary frequencies spanning multiple weeks, it was felt that students attending lessons any less than once a week would be unlikely to consider the course an adequate in-person experience, and might opt to attend purely online. As such, we limited the set of blocks $\mathcal{B}(\ell)$ for each lesson $\ell \in \mathcal{L}$ to exclude any blocks that were smaller than $1/M_\ell$ times the course’s enrollment per section.

Finally, we note that available times are now indexed not by the term t , but by (d, p) , where $d \in \mathcal{D}$ denotes a day of the week, and $p \in \mathcal{P}(d)$ denotes a particular 30-minute *period*. We choose 30 minutes as a discrete-time unit because all MIT Sloan lessons occur in multiples of 30 minutes, with a large majority of 90-minute lessons along with some 60-minute and 120-minute lessons.

Decision variables. We conserve the basic two-stage structure in which the variables \mathbf{x} determine course schedules, and the variables \mathbf{z} correspondingly determine student schedules, taking into account the refinements mentioned above:

$$x_{\ell,b,d,p} = \begin{cases} 1, & \text{if lesson } \ell \in \mathcal{L} \text{ starts on day } d \in \mathcal{D}, \text{ period } p \in \mathcal{P}(d), \text{ in room block } b \in \mathcal{B}(\ell), \\ 0, & \text{otherwise;} \end{cases} \quad (4a)$$

$$z_{s,\ell,d,p}^{\text{on}} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ starts lesson } \ell \in \mathcal{L}(s) \text{ on campus on day } d \in \mathcal{D}, \text{ period } p \in \mathcal{P}(d), \\ 0, & \text{otherwise;} \end{cases} \quad (4b)$$

$$z_{s,\ell,d,p}^{\text{any}} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ starts lesson } \ell \in \mathcal{L}(s) \text{ in any mode on day } d \in \mathcal{D}, \text{ period } p \in \mathcal{P}(d), \\ 0, & \text{otherwise;} \end{cases} \quad (4c)$$

$$y_{s,d} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ is allowed on campus on day } d \in \mathcal{D}, \\ 0, & \text{otherwise.} \end{cases} \quad (4d)$$

We further define the following auxiliary variables:

$$\rho_{\ell,b} = \begin{cases} 1, & \text{if lesson } \ell \in \mathcal{L} \text{ is held in room block } b \in \mathcal{B}(\ell), \\ 0, & \text{otherwise;} \end{cases} \quad (4e)$$

$$\phi_{\ell,p} = \begin{cases} 1, & \text{if lesson } \ell \in \mathcal{L} \text{ starts in period } p \in \mathcal{P} \text{ on any day,} \\ 0, & \text{otherwise;} \end{cases} \quad (4f)$$

$$\sigma_{s,c,j} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ is assigned to section } j \in \mathcal{J}(c) \text{ for course } c \in \mathcal{C}(s), \\ 0, & \text{otherwise.} \end{cases} \quad (4g)$$

Following Da Fonseca et al. (2017), our decision variables indicate the period in which a lesson starts, not all periods in which it is scheduled. This modeling choice has the advantage of precluding the need for imposing temporal contiguity constraints, at the small cost of slightly complicating temporal conflict-avoiding constraints.

Relational constraints. We now detail the main constraints necessary to model the timetabling problem. We begin with constraints of type (1b), relating student and course variables.

- Each student can only take a lesson during a day and period it is offered (analogous to (2e)):

$$z_{s,\ell,d,p}^{\text{any}} \leq x_{\ell,b,d,p} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}(s), b \in \mathcal{B}(\ell), d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4h)$$

- Each room block b has a limited capacity for lecture ℓ :

$$\sum_{s \in \mathcal{S}: \ell \in \mathcal{L}(s)} z_{s,\ell,d,p}^{\text{on}} \leq \hat{Q}_{b,\ell} x_{\ell,b,d,p} \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}(\ell), d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4i)$$

Above, we have defined the capacity of a block as $\hat{Q}_{b,\ell}$, depending not only on the block b but also on the lesson ℓ . This may seem overwrought, as the physical capacity of a classroom is fixed. But

we introduce this notion to model rotations, which, as previously discussed, divide the enrollment into groups of students who each attend one on-campus lesson a week.

The calculation of capacity $\hat{Q}_{b,\ell}$ takes into account the following cases. If all of the students for lesson ℓ can fit in block b , then there is no need to rotate. But if the block cannot fit all the students, then students must attend lesson ℓ in rotation, with each student attending one out of every M_ℓ weekly lessons (recalling that the set of blocks $\mathcal{B}(\ell)$ only includes blocks large enough for such a rotation). Then, only a fraction $1/M_\ell$ of the students would attend lessons in-person on any given day, and the capacity $\hat{Q}_{b,\ell}$ must be adjusted accordingly. In addition, we know that for each lesson ℓ for a course c , the maximum number of students that can attend will be given by $S_\ell := |\mathcal{S}(c)| (1/|\mathcal{J}(c)| + \epsilon)$, where ϵ is a pre-specified tolerance level that enforces balanced enrollment across sections up to this tolerance. We can then set the capacity as:

$$\hat{Q}_{b,\ell} = \begin{cases} S_\ell, & \text{if } S_\ell \leq \sum_{r \in \mathcal{R}(b)} Q_r, \\ S_\ell/M_\ell, & \text{if } S_\ell > \sum_{r \in \mathcal{R}(b)} Q_r, \end{cases}$$

where the first case represents that the block is large enough not to require rotation for the lesson, and the second case represents that $1/M_\ell$ will attend each lesson in rotation in smaller room blocks.

Course constraints. We now describe constraints of type (1c) which model course- and faculty-related scheduling constraints.

- Each lesson is assigned to the same block of rooms over the course of the week:

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} x_{\ell,b,d,p} = M_\ell \rho_{\ell,b} \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}(\ell), \quad (4j)$$

$$\sum_{b \in \mathcal{B}(\ell)} \rho_{\ell,b} = 1 \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}(\ell). \quad (4k)$$

- Each lesson is scheduled at the same time over the course of the week:

$$\sum_{b \in \mathcal{B}(\ell)} \sum_{d \in \mathcal{D}: p \in \mathcal{P}(d)} x_{\ell,b,d,p} = M_\ell \phi_{\ell,p} \quad \forall \ell \in \mathcal{L}, p \in \mathcal{P}, \quad (4l)$$

$$\sum_{p \in \mathcal{P}} \phi_{\ell,p} = 1 \quad \forall \ell \in \mathcal{L}, p \in \mathcal{P}. \quad (4m)$$

- Multiple lessons cannot consume room or faculty resources concurrently:

$$\sum_{b \in \mathcal{B}: r \in \mathcal{R}(b)} \sum_{\ell \in \mathcal{L}: b \in \mathcal{B}(\ell)} \sum_{p'=p-N_\ell+1}^p x_{\ell,b,d,p'} \leq 1 \quad \forall r \in \mathcal{R}, d \in \mathcal{D}, p \in \mathcal{P}(d), \quad (4n)$$

$$\sum_{\ell \in \mathcal{L}: f \in \mathcal{F}(\ell)} \sum_{b \in \mathcal{B}(\ell)} \sum_{p'=p-N_\ell+1}^P x_{\ell,b,d,p'} \leq 1 \quad \forall f \in \mathcal{F}, d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4o)$$

- Lessons cannot continue into forbidden time periods (e.g. lunch or end of day):

$$\sum_{p \in \mathcal{P}(d): \sum_{p'=p}^{p+N_\ell-1} \mathbf{1}_{p' \notin \mathcal{P}(d)} > 0} x_{\ell,b,d,p} = 0 \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}(\ell), d \in \mathcal{D}. \quad (4p)$$

Student constraints. We finally describe constraints of type (1d), modeling student enrollment and density effects.

- Each student must attend all mandatory lessons for their assigned section:

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}} z_{s,\ell,d,p}^{\text{any}} = M_\ell \sigma_{s,c,j} \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), j \in \mathcal{J}(c), \ell \in \mathcal{L}(j). \quad (4q)$$

We note that if a student only needs to attend one of the lessons ℓ_1, \dots, ℓ_k with the same multiplicity (for instance, one of two recitations in Fig. 2), we can easily modify constraint (4q), summing the left-hand side over the lessons ℓ_1, \dots, ℓ_k .

- Each student is assigned to at most one section per course:

$$\sum_{j \in \mathcal{J}(c)} \sigma_{s,c,j} \leq 1 \quad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), j \in \mathcal{J}(c). \quad (4r)$$

- No student can attend multiple lessons in the same day and period (analogous to (2i)):

$$\sum_{\ell \in \mathcal{L}(s)} \sum_{p'=p-N_\ell+1}^p z_{s,\ell,d,p'}^{\text{any}} \leq 1 \quad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4s)$$

- Each student can only attend a lesson on campus if they meet the requirements to attend it in any format (analogous to (2j)):

$$z_{s,\ell,d,p}^{\text{on}} \leq z_{s,\ell,d,p}^{\text{any}} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}(s), d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4t)$$

- Each student can only attend a lesson on campus if on campus that day (analogous to (2k)):

$$z_{s,\ell,d,p}^{\text{on}} \leq y_{s,d} \quad \forall s \in \mathcal{S}, \ell \in \mathcal{L}(s), d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (4u)$$

- The total number of students on campus each day is capped (analogous to (2l)):

$$\sum_{s \in \mathcal{S}} y_{s,d} \leq Y_d \quad \forall d \in \mathcal{D}. \quad (4v)$$

- Total student enrollment is balanced across sections of the same course, up to a tolerance ϵ :

$$\sum_{s \in \mathcal{S}(c)} \sigma_{s,c,j} \leq \left(\frac{1}{|\mathcal{J}(c)|} + \epsilon \right) \sum_{s \in \mathcal{S}(c)} \sum_{j' \in \mathcal{J}(c)} \sigma_{s,c,j'} \quad \forall c \in \mathcal{C}, j \in \mathcal{J}(c). \quad (4w)$$

Objective. The objective (4x) is to maximize the number of courses that students are able to take, with a preference for on-campus experiences:

$$\max_{\mathbf{x}, \mathbf{y}, \mathbf{z}^{\text{on}}, \mathbf{z}^{\text{any}}, \boldsymbol{\rho}, \boldsymbol{\phi}, \boldsymbol{\sigma}}, \sum_{s \in S} \sum_{\ell \in \mathcal{L}(s)} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} (z_{s,\ell,d,p}^{\text{on}} + \lambda z_{s,\ell,d,p}^{\text{any}}). \quad (4x)$$

The non-negative parameter λ is used to prioritize on-campus experiences, with on-campus experiences weighted by $1 + \lambda$ and off-campus experiences weighted by λ .

We can also refine the formulation to take into account desirability of teaching times. For days and periods where students or faculty are completely unavailable, the relevant \mathbf{x} variables can be set to zero. For soft preferences, weights can be added to the objective function, with higher weights assigned to desirable hours such as weekdays between 10am and 2pm, and lower weights assigned to undesirable hours such as 8am to 10am.

Specialized constraints. Formulation (4) summarizes the basic components of the timetabling problem, but can accommodate a variety of additional considerations, including repeating lesson patterns, students in international time zones, and room changes. The formulation of these extensions is discussed in the online supplement.

2.4. Practical considerations

Trading off multiple objectives. Readers will note that in refining the model, we often introduce new objectives of interest, e.g. minimizing schedule gaps or room changes. Our model must therefore be able to trade off these different priorities. One possible approach is to consider a weighted sum of all objectives of interest. While this approach can yield insights on the efficiency frontiers of different objectives, tuning the weights takes time. We therefore use a simpler technique, in which we first rank the objectives in order of importance. We then solve the problem with only the first objective, then with the second objective, and so on, each time constraining the previous objectives not to deviate (possibly within some tolerance) from their optimal values.

For instance, when we discuss scheduling core courses for first-year MBAs in the following section, we will minimize three objectives: schedule conflicts, schedule gaps, and days spent on campus. We first try to minimize schedule conflicts; then minimize schedule gaps, subject to the constraint of not introducing any additional conflicts; and finally minimize the number of days on campus, subject to the constraint of not introducing any additional conflicts or gaps.

Tractability. One cost of our model's expressiveness is a profusion of decision variables, which can make the problem difficult to solve. We develop several techniques to improve tractability. The first involves simple dimensionality reduction: instead of allowing lessons to start in any period, we designate a subset of periods as *starting periods*, and only allow lessons to start at these designated times. For instance, since most lessons are 90 minutes long, we only allow starts every 90 minutes,

thus reducing the number of variables by up to a factor of three. This simplification is common in the usual manual timetabling process.

A second complexity of our model is its joint consideration of room and time selection. In order to improve tractability, we can instead adopt a two-step approach, in which we first schedule courses, then assign them rooms. Because courses can always be offered online, this procedure is guaranteed to find a feasible solution, though it may be suboptimal. Indeed, this type of decomposition is widely used in the timetabling literature (Sorensen and Dahms 2014).

Finally, we note that a group of students $\bar{\mathcal{S}} \subseteq \mathcal{S}$ may share the same set of required courses, introducing symmetry which can be broken in various ways, such as by replacing the binary variables $\{z_{s,\ell,d,p}\}_{s \in \bar{\mathcal{S}}}$ with a single integer variable indexed by $\bar{\mathcal{S}}$, and bounded above by $|\bar{\mathcal{S}}|$ (Boland et al. 2008). This approach can complicate student conflict modeling; we instead adopt a simpler one. If a subset of students $\bar{\mathcal{S}} \subseteq \mathcal{S}$ have the same required classes, we impose that these students always attend the same lessons — functionally acting as a single “super-student” of size $|\bar{\mathcal{S}}|$. We then introduce a binary variable $z_{\bar{\mathcal{S}},\ell,d,p}$, which equals 1 if all students in $\bar{\mathcal{S}}$ attend lesson ℓ on day d and period p , and 0 if none of them do. This all-or-nothing approach should be used with care, as it can significantly reduce the feasible solution space. However, we will describe how one of the applications presented in the following section provides an ideal use case for this approach.

3. Case Study: Scheduling at MIT

In this section, we describe how our methods are implemented in practice. We first detail data collection and preprocessing, then explain how our models informed MIT’s exploration of a three-semester academic calendar in the Spring of 2020, before describing how we constructed a schedule for the Sloan School of Management, which was implemented for the Fall 2020 semester.

3.1. Data preparation

When we began the scheduling process at MIT, students had not yet pre-registered for courses. Rather than ask students to go through pre-registration while campus policy (including the schedule) was in flux due to COVID-19, we decided to use enrollment data from the last academic year as a proxy for the upcoming year. A limitation of this approach is that students’ required courses may themselves change according to the schedule, but lacking a model of how such changes might manifest themselves, we opted to simply use last year’s data.

For the term planning problem, the MIT Registrar provided us with the following data: (1) the previous year’s course schedule (2019-2020); (2) course enrollment history for all undergraduates from the graduating classes of 2016 and onward, including current students; (3) pre- and co-requisites for all courses. From this data, we needed to identify the required courses $\mathcal{C}(s)$ for all

students $s \in \mathcal{S}$. For example, a physics major’s required courses could include physics courses along with relevant mathematics courses.

To this end, we associated each major at MIT with a list of “common major courses,” which were defined as courses that at least 10% of historical students in that major had enrolled in at some point. The historical students were taken from the graduating class years of 2019 and 2020, so that newer courses would be included. Upon inspection, we found that the common major courses included both major requirements and popular electives. The 10% threshold was quite conservative: for example, many science and engineering course lists included a few music courses, as a nontrivial fraction of MIT students minor in music. We felt it was better to include a few less relevant courses than to miss a critical elective, particularly for students in small majors or major specializations. Each student was then associated with the courses that they had actually taken in the 2019-20 academic year, filtered to the common major courses for that student’s major(s).

To be conservative in respecting prerequisite relationships, we also added any unfulfilled prerequisites to each students’ required courses, meaning that we actually attempt to schedule more courses than strictly necessary. We made this addition because it was unknown whether students had skipped prerequisites because of a desire to move ahead quickly in their coursework or if there had been a scheduling issue preventing them from taking a prerequisite; given this uncertainty, we opted to be conservative. This meant that even MIT’s original two-semester schedule would not fulfill 100% of student-courses in our computational results.

Our term planning data set then consisted of 4,287 undergraduates enrolled across 620 courses, with an average of 5.26 required courses per year. First-years had the highest number of required courses, while seniors had the lowest (6.57 and 3.40, respectively). These numbers reflect the natural advancement of undergraduate coursework at MIT. First-years, regardless of major, must complete core “General Institute Requirements” in calculus, biology, chemistry, and physics. As students move forward and fulfill their requirements, they become more flexible in their course choices, and the number of required courses goes down accordingly. We capped the number of required courses each student could take per semester at three. Students enrolled in more than six required courses over the year were allowed to take up to half of their required courses in one term. Our added prerequisites did not count towards a student’s cap.

For timetabling, the Sloan School of Management provided us with the following data: (1) courses to be scheduled and associated faculty for the upcoming semester (Fall 2020); (2) course enrollments from Fall 2019; (3) available classrooms, with updated capacities to account for physical distancing; (4) faculty availability and teaching mode preferences.

For a few new courses with no past enrollment data, we took the list of degree programs for which the courses were targeted (e.g., Master of Business Administration), then sampled students

uniformly from these programs and added them to the new courses. Our final timetabling data set consisted of 1,455 students enrolled across 114 courses, with an average of 5.33 courses per student; as well as 115 faculty members and 26 classrooms with a mean capacity of 19.46 students.

All code was implemented using the `Julia` language (Bezanson et al. 2017) and the optimization package `JuMP` (Dunning et al. 2017) using the Gurobi solver (Gurobi Optimization, Inc. 2016). Computational experiments for term planning were run on a single multi-core machine on a computing server, requesting eight cores for Gurobi. Computational experiments for timetabling were run on the authors’ laptop computers (Early 2016 MacBook Pro and 2019 Dell XPS 13).

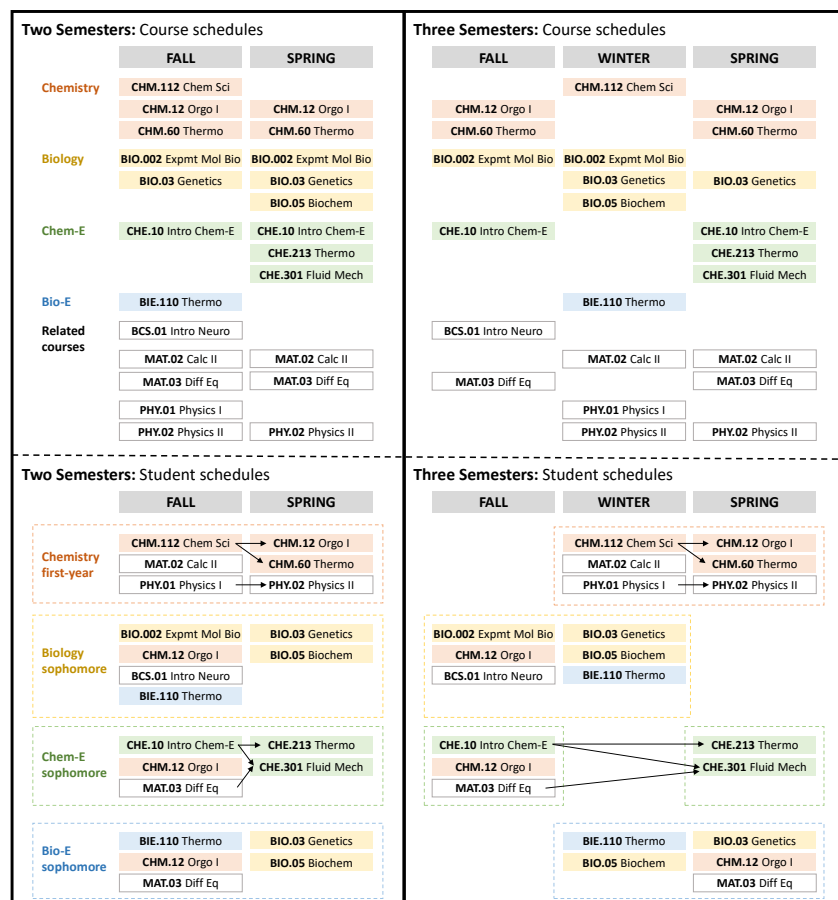
3.2. Three-semester planning at MIT

3.2.1. Illustrative schedules. When MIT first started to consider altering its academic calendar by increasing the number of semesters, both students and faculty expressed uncertainty as to the viability of this path. Students were concerned that important courses might be offered during their time off-campus, limiting their ability to learn from faculty and each other. Faculty were concerned that extra courses would need to be offered in order to ensure that students could take courses on-campus, dramatically increasing teaching loads.

In order to convey the intuition behind the three-semester model, we presented course schedules and student schedules for a small set of related departments (Figure 3). The Chemistry (CHM), Biology (BIO), Chemical Engineering (CHE), and Biological Engineering (BIE) departments share many required courses; for example, many students in these departments take CHM.12 (Organic Chemistry I). Furthermore, many required courses include a laboratory component best completed on campus. The combination of shared requirements and laboratory needs made these four departments a motivating use case for our model. We note that for the sake of clarity, we label departments using three-letter labels (e.g. CHM, BIO) instead of the standard MIT practice, which identifies most departments using numbers (e.g. 5 for Chemistry, 7 for Biology).

The top left panel of Figure 3 shows the existing two-semester course schedule for a variety of courses across the four departments of interest, as well as some relevant courses in related departments. A course appears under FALL heading if it is offered in the fall semester, and similarly for the spring; some courses, such as CHM.12 (Organic Chemistry I), are offered in both semesters. The bottom left panel of Figure 3 shows some typical student schedules. Once again, courses appear in the columns corresponding to the terms in which they are taken, during which the course must also be offered as per the course schedule.

The top right panel of Figure 3 shows the proposed three-semester course schedule, which only involves shifting select fall and spring courses to the new winter semester. Constraint (21) was set to limit on-campus density to 50% in the fall semester, and 75% in each of the two subsequent

Figure 3 Course and student schedules for a subset of departments under two and three semesters.

Note. Sample courses from the interconnected departments of Chemistry (CHM, red), biology (BIO, yellow), Chemical Engineering (CHE, green), and Biological Engineering (BIE, blue) are displayed along with related courses (white) from Brain and Cognitive Sciences (BCS), Mathematics (MAT), and Physics (PHY). The left panels show the course schedules under the existing two-semester policy (top), as well as some typical example student schedules (bottom). The right panels show how by shifting select fall and spring courses to a new winter semester, students can be spread out on campus for two out of three semesters. Prerequisite relationships (black arrows) are respected in both the two-semester and three-semester student schedules.

semesters. The bottom right panel shows the student schedules induced by the three-semester schedule. Notably, each student's coursework is confined to only two of the three semesters. Much of the students' schedules are simply translated across semesters; for example, the chemistry first-year and the chemical engineering sophomore's schedules occur in exactly the same order as they did under the two-semester policy. This is thanks to the enforcement of prerequisite relationships in constraint (2h), examples of which are displayed with arrows in Figure 3. For subjects without prerequisite relationships, schedules could be more freely rearranged.

A key reason why these courses can be so neatly rearranged in three semesters without impeding students' progress is that the existing MIT schedule contains substantial redundancy for a number

of courses. For example, CHM.12 (Organic Chemistry I) is offered both in the fall and the spring. In the fall, it is taken by sophomores who completed their general chemistry prerequisites during their freshman year, typically outside of the chemistry department. But in the spring, it is taken by first-year chemistry majors looking for a jump start on important major-related coursework. The fall and spring offerings of CHM.12 could conceivably be condensed into a single spring offering without running afoul of prerequisites, if the non-chemistry sophomores deferred taking CHM.12 from the fall to the spring semester. However, all of the students in Figure 3 would then need to be on-campus during the spring semester, violating the campus density constraint (21). As such, the second offering of CHM.12 is critical for spreading students across the academic year.

For the relatively small number of courses and students in Figure 3, it is possible to manually find and evaluate a solution. However, with larger numbers of courses and students at the scale of an entire university, optimization is critical for computing a three-semester schedule. In the next set of experiments, we will demonstrate the tractability of our model, and then discuss the policies considered by MIT in the term planning process.

3.2.2. Tractability on large datasets To test the tractability of our model, we created subsets of the term planning data in increments of about 20% of the courses in the full dataset. We began with students in the chemistry department (115 courses and 31 students), then added students from the related departments of Figure 3 (246 courses, 385 students). At each step, we added students from the departments related most closely to the given subset of data, as measured by the number of students enrolled in courses from that department. The sizes of the various data instances are given in Table 2, along with the corresponding model objective values and the time taken to finish solving to an optimality gap of 0.5% or less (capped at 720 minutes). We imposed on-campus density of 50% in the fall semester, and 75% in the winter and spring.

Our model solves within a few minutes smaller instances, but instances with hundreds of courses and thousands of students require multiple hours. These running times are acceptable for term planning, a strategic decision that is only infrequently reevaluated. Moreover, although the full problem did not reach the desired gap in 720 minutes, optimality gaps of 10% and 5% were achieved in 120 and 252 minutes, respectively, indicating that should time be scarce, early termination of the solver would still leave policymakers with actionable solutions.

3.2.3. Policy evaluation and decision-making We now evaluate the impact of moving to a three-semester schedule on the MIT undergraduate experience. Our models were run on a variety of different policies with either two or three semesters. The two-semester policies were run as a baseline, and enforced that each student was on campus for exactly one of two semesters, with campus capacity not exceeding 50%. Under the three-semester policy, 50% of students would be

Table 2 Objective values and running times for term-planning problems of varying size.

Courses	Students	Objective	Time (min)
115	31	247	0
246	385	2,579	3
372	787	4,699	11
479	1,682	10,271	133
620	4,287	26,854	720

Solver time limit of 720 minutes, gap tolerance of 0.5%; the 620-course example solved to a gap of 2.4%.

present in the fall, and 75% of students would be present in each of the two subsequent semesters, with each student on campus for exactly two semesters, and off campus for exactly one semester.

The policies also included various methods of partitioning students across the different semesters. The first was a fully flexible partition, allowing the model to decide which students to invite on campus each semester (subject to density limits). The second method enforced that students in the same major and year would be assigned to be on campus in the same terms. The final method partitioned students by class year, and further enforced which semesters would be attended on campus by each class year. Under a two-semester model, first-years and sophomores would be present in the fall, and juniors and seniors would be present in the spring. Under a three-semester model, first-years and sophomores would be present in the fall, sophomores, juniors, and seniors would be present in the winter, and first-years, juniors, and seniors would be present in the spring. The major-year and year partitions were considered more actionable than a fully flexible policy.

Table 3 Model performance and running time for various term-planning policies.

Terms	Student Partition	Percentage of Students' Courses			Time (min)
		On-Campus	Off-Campus	Unsatisfied	
2	Flexible	59%	35%	6%	42
2	Major-Year	57%	37%	6%	69
2	Year	51%	43%	6%	0
3	Flexible	96%	3%	1%	720
3	Major-Year	92%	6%	2%	720
3	Year	90%	8%	2%	2

Solver time limit of 720 minutes; the three-semester flexible and major-year policies solved to gaps of 2.4% and 5.1%, respectively.

Table 3 reports the performance of each policy in terms of the percentage of courses satisfied during the average student's on-campus semesters, off-campus semesters, or left unsatisfied due to either prerequisite constraints or the cap on courses taken per semester (2i). Unsurprisingly, a two-semester schedule with students on campus for exactly one semester would result in students taking only half of their required courses while on campus. This metric could be improved in a

fully flexible policy, by assigning students to the semester in which they had the most required courses, but only to 59%. About 6% of students' courses were left unsatisfied, due to the additional prerequisites added to each students' required courses.

By contrast, moving to a three-semester calendar would substantially improve the student experience while also satisfying campus density restrictions. The increased time on campus would allow students to take 90% or more of their required courses while on campus, up from half in the two-semester calendar. The fully flexible partition naturally showed the best performance, with 96% of students' courses satisfied while on-campus. An additional benefit of the three-semester calendar was that the increased flexibility helped satisfy a greater number of students' courses; only 1-2% of students' courses were left unsatisfied, as compared to 6% in the two-semester calendar.

These metrics were reported to MIT community members in a series of town halls and focus groups. Although our model's performance was convincing, concerns were raised that such an approach would be disruptive in what had already been a year of unprecedented disruption to the routines of students, faculty, and staff. As such, MIT decided to move forward with the normal two-semester calendar, only committing to inviting seniors to campus in the fall, and holding all undergraduate courses online (Song 2020), with some courses also having in-person components.

3.3. Physically-distanced timetabling at MIT Sloan

Once MIT decided to move forward with a reduced-density campus on the normal two-semester calendar, departments within the university had to develop teaching plans for the 2020-2021 academic year, finalizing the weekly course timetable for a hybrid in-person and online student experience.

Each department faced unique challenges. For instance, the Sloan School of Management was relatively unaffected by rules impacting undergraduate students due to its small undergraduate enrollment. However, Sloan needed to accommodate the diverse needs of a graduate body of more than 1,000 students across several different degree programs and, because of the pandemic, located across the globe. Table 4 lists a selection of Sloan degree programs under consideration, as well as the approximate number of students in each program.

Most Sloan programs are non-thesis degrees, and therefore depend almost entirely on coursework. For students in these programs, a dynamic classroom experience is a key attraction, as they interact with faculty and their classmates in discussion-based classes. Therefore, it was feared that a fully online experience could fall short of students' expectations. This issue was felt particularly acutely for first-year students, who had for the most part never met in person. Indeed, through the spring and summer of 2020, many incoming students voiced a strong interest in maintaining an in-person educational component to Sloan leadership.

The challenge faced by the Sloan School of Management in the early summer of 2020 was thus to develop a course timetable that would enable students to necessarily fulfill their degree

Table 4 Approximate enrollment numbers for select Sloan degree programs.

Program	Students
MBA First-Year	490
MBA Second-Year	360
MFin	170
Sloan Fellow	100
Undergraduate	100
Other	200

obligations with an appropriate amount of physically distanced in-person learning, while keeping Sloan community members safe according to public health guidelines.

Public health also motivated our particular approach to the timetabling problem. In the fall semester, first-year MBA students take a series of five required *core* courses. Enrollment for these core courses is closed to other programs, which allows for a natural decomposition of the problem between first-year MBA students and everyone else. Such a decomposition is desirable not just for model tractability, but also because separation between different degree programs complements existing physical distancing measures. We first discuss timetabling for the MBA first-year core classes in Section 3.3.1, before discussing the remainder of the problem in Section 3.3.2.

3.3.1. The MBA first-year core First-year MBA students at Sloan are typically divided into cohorts (also referred to as “oceans”) of approximately 60 students. Each individual cohort attends all core courses together, and has its own dedicated section for each course.

The COVID-19 pandemic compelled a re-working of the usual timetabling procedure. First, classroom capacities were dramatically reduced to facilitate physical distancing between students. Second, although this posed concerns for student, faculty, and staff workload, the weekly calendar was lengthened to include Saturdays, and extend from 8am to 9:30pm on Monday through Thursday, and from 8am to 7pm on Fridays and Saturdays (excluding a daily lunch break from 12:30 to 2pm). While student and faculty preferences had previously led to many courses being scheduled on weekdays between 10am and 2pm, a pandemic-era schedule would need to spread out use over the entire available week. However, faculty were allowed to specify that they were unavailable to teach either in the early mornings, late evenings, or Saturdays.

For safety reasons, it was desirable to schedule MBA first-years and all other students on-campus on alternating days. There were also certain large classrooms that, while potentially convenient for accommodating larger courses, were not directly managed by Sloan, and usage of these classrooms needed to be reserved for other MIT students. We refer to these classrooms as “expanded capacity,” as compared to the “base capacity” corresponding to classrooms directly under Sloan’s purview.

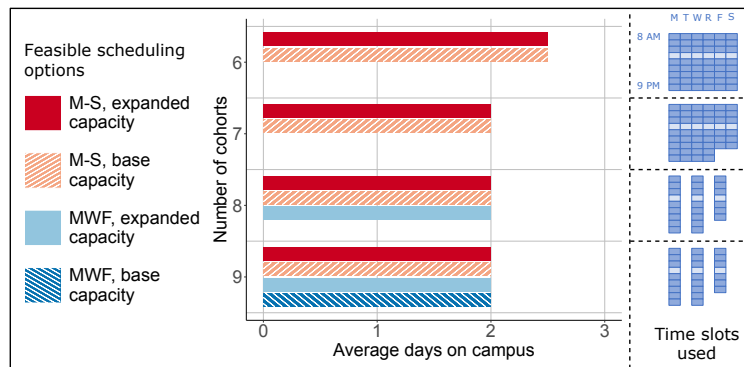
A further challenge was that instead of the usual MBA class size of about 360 students, MIT Sloan admitted substantially more students for the fall of 2020. Though the final enrollment was not known (due to deferrals, withdrawals, etc.), in the early summer of 2020, MIT Sloan was planning for a class size of 490 students. A key question was therefore to determine not just when to schedule courses for each cohort, but indeed determining how many cohorts to create. A few large cohorts might not have many room blocks available, while many small cohorts could create strain on classroom and teaching resources. Our optimization model was used to make the key decision of selecting the number of cohorts in which to split the incoming MBA first-years.

For this problem, we first minimized schedule conflicts, then gaps in student schedules, then the number of days each student spent on campus, as outlined in Section 2.4. We also imposed that students should not occupy more than one room block per day, in order to avoid crowding among students switching between classes. Finally, we relaxed the constraint that each instance of a repeating lesson must occur in the same room block.

We first showed that hosting the usual six cohorts would only be possible at the cost of first-year MBA students occupying the entirety of the Monday-through-Saturday calendar, extended to include Friday and Saturday evenings until 9:30pm. The further expansion of the week was necessary because only one block of classrooms was large enough to hold 82 students, which meant that no core courses could be held simultaneously. Furthermore, although most cohorts would be able to limit their commutes to campus to two days a week, some cohorts would have to commute to campus for three days a week, which would unfairly increase their public health risk.

We then showed that adding cohorts would achieve a number of desirable safety objectives. Adding a seventh cohort allowed us to avoid scheduling courses during the unpopular late evening hours on weekends, and also allowed all students to limit their commutes to campus to two days a week. Adding an eighth cohort allowed all first-year courses to be scheduled on Monday, Wednesday, and Friday, rather than take up all six days in the academic week. This would allow for separation between first-year MBAs and students in other programs, who could then have their classes scheduled only on Tuesday, Thursday, and Saturday. However, Sloan's expanded capacity had to be used in order to fit all lectures in three days. Finally, adding a ninth cohort allowed for three-day scheduling without using expanded-capacity rooms, which could then be reserved exclusively for students in other programs at MIT. We summarize our results in Figure 4. From this analysis, Sloan decided to use nine cohorts, achieving the goals of allowing for physical distancing and also avoiding usage of classrooms that needed to be reserved for other MIT students, while minimizing the additional faculty teaching load.

Figure 5 shows the lessons taking place in an illustrative room block comprising two classrooms with combined capacity sufficient for a 54-person cohort. The first two cohorts have all eight lectures

Figure 4 Scheduling implications of different numbers of first-year MBA cohorts.

Note. On the left, we show the average number of days per week students spend on campus for different numbers of cohorts and scheduling considerations. Missing bars indicate infeasibilities. On the right, we show the minimum set of time slots necessary to make the problem feasible, illustrating that more cohorts allows for scheduling on three out of six days, leaving the remaining three days for students in other programs.

Figure 5 First-year MBA room usage for an example block comprising classrooms E62-262 and E62-276 .

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
08:00 AM	MGT.311 Cohort 1		MGT.060 Cohort 3		MGT.280 Cohort 1	
09:30 AM	MGT.060 Cohort 1		MGT.280 Cohort 3		MGT.060 Cohort 1	
11:00 AM	MGT.515 Cohort 1		MGT.010 Cohort 3		MGT.515 Cohort 1	
12:30 PM						
02:00 PM	MGT.010 Cohort 1		MGT.515 Cohort 3		MGT.010 Cohort 1	
03:30 PM	MGT.515 Cohort 2		MGT.515 Cohort 3			
05:00 PM	MGT.311 Cohort 2		MGT.280 Cohort 2			
06:30 PM	MGT.010 Cohort 2		MGT.010 Cohort 2			
08:00 PM	MGT.060 Cohort 2		MGT.060 Cohort 2			
09:30 PM						

Note. Scheduling options were nine cohorts on Mon/Wed/Fri, using only the base classroom capacity. Cohorts 1 (red) and 2 (blue) stay entirely in classrooms E62-262 and E62-276. Half of the lectures for Cohort 3 (yellow) are not shown; on Friday, this cohort moves to classrooms E62-164 and E62-176 so as not to conflict with Cohort 1.

scheduled in the same classrooms, while the third cohort only has four lectures scheduled in the same classroom. On Friday, the third cohort must move to two other classrooms of comparable size (not shown) so as not to conflict with the first cohort. Crucially, all cohorts are able to stay in the same room over the course of each day, and the lessons are completely contiguous in time (excluding the Sloan-wide 12:30-2pm lunch break), facilitating physical distancing on campus.

3.3.2. Other MIT Sloan courses We now discuss the remaining courses outside of the first-year MBA core required courses, which we will call “non-core” courses for brevity. Each course

was allowed to specify that either lecture or recitation should be in-person, that both lecture and recitation should be online, or that any mode of teaching was allowable. By default, since teaching assistant (TA) preferences were unknown, we set recitations to be online unless requested otherwise. The online courses included MBA electives, which have substantial enrollment from both first- and second-year students, and thus were set online to keep the class years separated. Courses with substantial undergraduate enrollments were also set online, due to the institute-wide decision for undergraduate courses to be online. Finally, various faculty indicated in a survey that they would not find teaching in a physically-distanced classroom to be effective, and would consequently prefer to teach online. Overall, 47 out of 109 non-core courses were set to be purely online.

We limited non-online lectures to be held only on Tuesday, Thursday, or Saturday, since Monday, Wednesday, and Friday on-campus resources were reserved for core courses. As mentioned in Section 2.4, the scale of the problem necessitated that we decompose the problem by solving first for time periods, then rooms. We could not guarantee that non-online lectures would have a classroom allocated, especially in cases of large enrollments. However, we increased the weights on z^{any} for important program-related courses, as well as for courses with faculty who had made a strong commitment to teaching in-person for the semester. We largely limited online lectures to be held on Monday, Wednesday, and Friday, excepting MBA electives, to preclude first-year MBA conflicts.

The full procedure is summarized as follows. (1) For Tuesday/Thursday/Saturday lectures: (a) solve for time periods assuming all are online, (b) fix lesson time periods and solve for classrooms. (2) For Monday/Wednesday/Friday lectures: solve for time periods assuming all are online. We set a time limit of 30 minutes for each step, chosen to accommodate iterative feedback from community members under tight time frames, while still achieving optimality gaps of at most 2%. Solving for classrooms terminated in seconds, while solving for time periods consumed the full time limit.

Our end result was a schedule where 97% of students' non-core lectures could be scheduled without conflicts. Moreover, despite substantial resource constraints, 96% of graduate students would take at least one course with some in-person component, and 68% of graduate students would take at least half of their courses with any in-person component. These metrics are likely underestimates, since they rely on last year's data, and students presented with a schedule can choose courses taking into account the availability of an in-person component.

In accordance with public health guidelines, our schedule also kept campus density to acceptably low levels. Mondays, Wednesdays, and Fridays were associated with the lowest number of students, ranging from 220 to 385 per day, since they were reserved for the smaller first-year MBA cohorts. On Tuesdays, Thursdays, and Saturdays, the number of students ranged from 374 to 557, in accordance with a goal to stay below 600 people on campus including faculty and staff.

Table 5 Teaching modes for all MIT Sloan courses.

Student Attendance	Classrooms	Courses	Students per Section
In-Person	One classroom	19	16.2
In-Person	Multiple classrooms	19	44.4
Rotating Capacity	One classroom	5	29.4
Rotating Capacity	Multiple classrooms	13	55.3
Fully Online	–	58	40.6
Total		114	49.0

The first column indicates whether all students in the courses have full access to the in-person lessons, whether they must rotate between days of the week, or whether the course is fully online. The second column indicates whether courses are assigned one or multiple classrooms. The last two columns show the number of courses and average number of students per section under each of the policies.

A summary of the various teaching modes for all of the courses in our schedule is presented in Table 5. We can see that courses with larger capacity needs tend to be given multiple classrooms rather than one classroom, or assigned to rotate rather than give students full access to in-person lessons. About half of the courses are offered with some in-person component, no mean feat given that campus capacity had been decreased to about a quarter of its pre-COVID-19 capacity.

Attending online lectures while on-campus. Student schedules on Tuesday, Thursday and Saturday might include online lessons, because we could not guarantee that all lectures scheduled on those days would receive a classroom assignment. This issue raised an immediate question: where would students go if they had to take both on-campus and online classes in a single day? Fortunately, this secondary problem is easier to solve. First, classrooms would not need to be allocated for *all* students enrolled in the online course, but only those who also had to attend some on-campus lecture that day. We found that only 138 students on Tuesday, 86 students on Thursday, and 26 students on Saturday needed to have a classroom at any time of the day, and not all simultaneously. Second, students could be split up into as many classrooms as necessary, only needing a place to sit while watching the lecture online. We implemented postprocessing routines to house small numbers of students in available classrooms over short blocks of time.

3.3.3. The Sloan schedule in practice

Incorporating faculty feedback. In addition to the models described here, implementing the final schedule relied on an iterative process of community feedback. A draft schedule was initially made available for open comments. The flexibility of our two-stage optimization model allowed us to directly incorporate new constraints and preferences, and partially re-optimize the schedule, adjusting only courses for which concerns were raised. After the final schedule was produced by the model, a few one-off scheduling changes were requested, but they were handled manually, by enumerating the possible moves and choosing the one with the smallest objective degradation.

Modeling limitations. The feedback process was particularly important, because it allowed the final schedule to overcome the limitations of our modeling assumptions. We used historical data to create a schedule that would minimize conflicts for the previous year’s students. Of course, student interests can vary from year to year, based on new course offerings, faculty schedules, and even the days and times courses are offered. The impact of these limitations is mitigated in part by Sloan’s course enrollment caps (preventing a 60-person course from suddenly becoming a 200-person course). Furthermore, though our model assigns student schedules (including rotations), it is likely that courses will adjust rotation models based on their teaching plans. Our model is also conservative in assuming all students will seek out in-person components. In practice, it is always possible for students who prefer to do so to attend all their courses remotely—but this simply frees up more on-campus capacity for students who desire an on-campus component.

Finally, careful readers of our model will no doubt have noticed that it does not include a fairness component (ensuring that every student in a rotating lesson has access to campus at least once). The model may discriminate against students who are enrolled in a small number of courses. It turns out that this number of students is relatively small, and this effect can be mitigated through more creative multi-week rotation schedules, or other course-by-course adjustments.

4. Conclusions

In this paper, we introduce a two-stage view of the course scheduling problem, encompassing several existing problems in the operations research literature. Our formulation has significant modeling power and can accommodate a variety of problem-specific constraints. We demonstrate the efficacy of our approach, as well as its limitations, on two case studies at the Massachusetts Institute of Technology, showing that optimization can be an indispensable tool to evaluate policy proposals and create new schedules when emergency conditions force a reevaluation of the status quo.

Acknowledgments

We would like to thank Ian Waitz, Mary Callahan, and Ri Romano for providing data from the MIT Registrar. We are also especially grateful to the MIT Sloan Scheduling Working Group, including Bill Garrett, Dan Gormley, Hasmik Kouchadjian, Retsef Levi, Lisa Monaco, Tara Walor, and Ezra Zuckerman Sivan for spearheading the construction of the Sloan Fall 2020 schedule, including defining objectives, operationalizing constraints, and organizing the feedback process. Computations for the MIT term planning problem and the non-core course timetabling problem were realized on MIT’s Supercloud computing system.

References

Abramson D (1991) Constructing school timetables using simulated annealing. Sequential and parallel algorithms. *Management Science* 37(1):98–113.

- Atef Yekta H, Day R (2020) Optimization-based Mechanisms for the Course Allocation Problem. *INFORMS Journal on Computing* (August).
- Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization* (Athena Scientific Belmont, MA).
- Bettinelli A, Cacchiani V, Roberti R, Toth P (2015) *An overview of curriculum-based course timetabling*, volume 23 (Springer Berlin Heidelberg), ISBN 1175001503641.
- Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. *SIAM review* 59(1):65–98.
- Birbas T, Daskalaki S, Housos E (2009) School timetabling for quality student and teacher schedules. *Journal of Scheduling* 12(2):177–197.
- Boland N, Hughes BD, Merlot LT, Stuckey PJ (2008) New integer linear programming approaches for course timetabling. *Computers and Operations Research* 35(7):2209–2233.
- Brown University (2020) Plan for a healthy and safe 2020-21. URL <https://www.brown.edu/sites/g/files/dprerj316/files/healthy-brown-2020-21-plan.pdf>, accessed 2020-09-14.
- Budish E, Cachon GP, Kessler JB, Othman A (2017) Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research* 65(2):314–336.
- Burke EK, Bykov Y (2016) An adaptive flex-deluge approach to university exam timetabling. *INFORMS Journal on Computing* 28(4):781–794.
- Carter MW, Tovey CA (1992) When is the classroom assignment problem hard? *Operations Research* 40(1):S28–S39.
- Ceschia S, Di Gaspero L, Schaerf A (2012) Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers and Operations Research* 39(7):1615–1624.
- Ceschia S, DiGaspero L, Schaerf A (2014) The generalized balanced academic curriculum problem with heterogeneous classes. *Annals of Operations Research* 218(1):147–163.
- Chiarandini M, Di Gaspero L, Gualandi S, Schaerf A (2012) The balanced academic curriculum problem revisited. *Journal of Heuristics* 18(1):119–148.
- Da Fonseca GHG, Santos HG, Carrano EG, Stidsen TJ (2017) Integer programming techniques for educational timetabling. *European Journal of Operational Research* 262(1):28–39.
- Da Fonseca GHG, Santos HG, Toffolo TÂM, Brito SS, Souza MJF (2016) GOAL solver: a hybrid local search based solver for high school timetabling. *Annals of Operations Research* 239(1):77–97.
- Daskalaki S, Birbas T, Housos E (2004) An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research* 153(1):117–135.

- Dinkel JJ, Mote J, Venkataramanan MA (1989) OR Practice - An Efficient decision support system for academic course scheduling. *Operations Research* 37(6):853–864.
- Dorneles ÁP, de Araújo OC, Buriol LS (2017) A column generation approach to high school timetabling modeled as a multicommodity flow problem. *European Journal of Operational Research* 256(3):685–695.
- Dunning I, Huchette J, Lubin M (2017) Jump: A modeling language for mathematical optimization. *SIAM Review* 59(2):295–320, URL <http://dx.doi.org/10.1137/15M1020575>.
- Farber MA (1971) Dartmouth acts to admit co-eds. URL <https://www.nytimes.com/1971/10/27/archives/dartmouth-acts-to-admit-coeds-trustees-also-get-plan-for-yearround.html>, accessed 2020-08-03.
- Garcia C (2019) Practice summary: Managing capacity at the University of Mary Washington’s College of business. *Interfaces* 49(2):167–171.
- García-Sánchez Á, Hernández A, Caro E, Jiménez G (2019) Universidad politécnica de Madrid uses integer programming for scheduling weekly assessment activities.
- Gonzalez G, Richards C, Newman A (2018) Optimal course scheduling for United States Air Force Academy cadets. *Interfaces* 48(3):217–234.
- Gurobi Optimization, Inc (2016) Gurobi optimizer reference manual. URL <http://www.gurobi.com>.
- Hinkin TR, Thompson GM (2002) SchedulExpert: Scheduling courses in the Cornell University School of Hotel Administration.
- Johnes J (2015) Operational research in education. *EJOR* 243(3):683–696.
- Kannan A, Van Den Berg G, Kuo A (2012) ISchedule to personalize learning. *Interfaces* 42(5):437–448.
- Lindahl M, Mason AJ, Stidsen T, Sorensen M (2018) A strategic view of University timetabling. *European Journal of Operational Research* 266(1):35–45.
- Massachusetts Institute of Technology (2020) Different options for fall and beyond. URL <https://covid-team2020.mit.edu/options/>, accessed 2020-08-03.
- Méndez-Díaz I, Zabala P, Miranda-Bront JJ (2016) An ILP based heuristic for a generalization of the post-enrollment course timetabling problem. *Computers and Operations Research* 76:195–207.
- Mulvey JM (1982) A classroom/time assignment model. *EJOR* 9(1):64–70.
- Phillips AE, Waterer H, Ehrgott M, Ryan DM (2015) Integer programming methods for large-scale practical classroom assignment problems. *Computers and Operations Research* 53:42–53.
- Pillay N (2014) A survey of school timetabling research. *Annals of Operations Research* 218(1):261–293.
- Santos HG, Uchoa E, Ochi LS, Maculan N (2012) Strong bounds with cut and column generation for class-teacher timetabling. *Annals of Operations Research* 194(1):399–412.
- Song E (2020) Administrators hold town hall to discuss recent fall 2020 decisions. URL <https://thetech.com/2020/07/18/fall-2020-undergraduate-town-hall>, accessed 2020-08-03.

- Sorensen M, Dahms FH (2014) A Two-Stage Decomposition of High School Timetabling applied to cases in Denmark. *Computers and Operations Research* 43(1):36–49.
- Stanford University (2020) A message from President Marc Tessier-Lavigne and Provost Persis Drell on academic planning for the fall quarter and 2020-21 academic year. URL <https://healthalerts.stanford.edu/covid-19/2020/06/03/>, accessed 2020-08-03.
- Strichman O (2017) Near-Optimal Course Scheduling at the Technion. *Interfaces* 47(6):537–554.
- Tripathy A (1984) School Timetabling - a Case in Large Binary Integer Linear Programming. *Management Science* 30(12):1473–1489.

Appendix A: Specialized timetabling constraints

Formulation (4) summarizes the basic components of the timetabling problem, but can accommodate a variety of additional considerations. Some examples include:

- **Repeating lesson patterns.** It may be pedagogically desirable to spread the sessions of a repeating lesson across the week at a reasonable cadence. For example, having lectures on Monday and Wednesday allows students one day between lectures to review material if needed. In order to prevent the sessions of a repeating lesson ℓ from being spaced apart by k days, we can use the following constraint:

$$\sum_{b \in \mathcal{B}(\ell)} \sum_{p \in \mathcal{P}(d)} (x_{\ell,b,d,p} + x_{\ell,b,d+k,p}) \leq 1. \quad (5a)$$

Small k prevents repeating lessons from occurring too closely, while large k prevents repeating lessons from being spaced too far apart. Care must be taken to not add this constraint for high-multiplicity lessons that would make the constraint infeasible; for example, if a lesson is held four days per week, then two of the sessions must be held on consecutive days, and this constraint for $k = 1$ would make the problem infeasible.

- **International time zones.** The COVID-19 pandemic compelled many international students to leave the country to be with family, and travel restrictions prevented many international students who wanted to come to campus from leaving their home countries. For core or in-demand classes, it was therefore important to offer international students an opportunity to tune in remotely at a reasonable time. To ensure that a particular course $c \in \mathcal{C}$ starts during a period in a pre-specified set of periods that are acceptable for international students, given by the set $\mathcal{P}_{\text{INTL}}$, we can add the constraint

$$\sum_{p \in \mathcal{P}_{\text{INTL}}} \phi_{\ell,p} \geq 1, \quad (5b)$$

for each lesson ℓ associated with course c . If c has multiple sections, we can either include these constraints for all sections, or enforce that they must be satisfied by the lessons of at least one section. Depending on other constraints in the problem such as faculty availability, hard constraints on international time zones may not be feasible. The constraints can be softened by subtracting a slack variable from the right-hand side of the constraint, then penalizing an appropriately weighted sum of these slack variables in the objective.

- **Room changes.** Some groups of students may take several courses together (e.g. MBA core courses). In such cases, it may be desirable for students to attend all or most of their classes in the same room or block of rooms, so as to reduce unnecessary movement within the building. From a modeling perspective, we can introduce auxiliary binary variables η , such that

$$\eta_{s,d,r} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ uses room } r \in \mathcal{R} \text{ on day } d \in \mathcal{D}, \\ 0, & \text{otherwise,} \end{cases} \quad (5c)$$

and further impose that a student uses a room if they attend any lesson in that room:

$$\eta_{s,d,r} \geq \sum_{b \in \mathcal{B}(r)} x_{\ell,b,d,p} \quad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d), \ell \in \mathcal{L}(s). \quad (5d)$$

We can then cap the number of rooms used by a student each day by enforcing the constraint

$$\sum_{r \in \mathcal{R}} \eta_{s,d,r} \leq R^{\max} \quad \forall s \in \mathcal{S}, d \in \mathcal{D}, \quad (5e)$$

where the parameter R^{\max} designates the maximum number of rooms used by a student on any given day. Alternatively, we can penalize the left-hand side of constraint (5e) in the objective, summed over all students and days.

We impose this constraint on rooms instead of blocks to acknowledge the difference between two blocks that share a classroom and two blocks that do not. It may also be desirable to relax constraint (4k) to encourage lessons to occur in blocks of comparable size instead of the same block to avoid overly constraining the timetabling problem.

- **Gaps in student schedules.** Another priority when developing the schedule may be to reduce gaps in student schedules, i.e., idle periods between classes. While simply a convenience in normal times, avoiding schedule gaps is significant in a reduced-capacity pandemic setting, where the aim is to reduce unnecessary in-person interactions under a physical distancing mandate. We can incorporate an approach similar to Da Fonseca et al. (2017), introducing auxiliary variables β such that

$$\beta_{s,d,p} = \begin{cases} 1, & \text{if student } s \in \mathcal{S} \text{ is busy on day } d \in \mathcal{D}, \text{ at period } p \in \mathcal{P}(d), \\ 0, & \text{otherwise,} \end{cases} \quad (5f)$$

where “busy” means the student is either attending class or between two classes. The former is captured by the following constraint:

$$\sum_{\ell \in \mathcal{L}(s)} \sum_{p'=p-N_\ell+1}^p z_{s,\ell,d,p'}^{\text{any}} \leq \beta_{s,d,p} \quad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d), \quad (5g)$$

while the latter is modeled with another constraint:

$$\max_{p' < p} \beta_{s,d,p'} + \max_{p' > p} \beta_{s,d,p} - 1 \leq \beta_{s,d,p} \quad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d). \quad (5h)$$

The max operators on the left-hand side of constraint (5h) can be modeled using additional auxiliary variables to keep the formulation linear (see Bertsimas and Tsitsiklis (1997)). We can then de-incentivize gaps by penalizing the sum of the β variables in the objective.