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. Under these conditions, we examine how to make the most of newly-scarce resources in 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, with each student attending two semesters in person, over 90% of student course demand could be satisfied on campus without increasing faculty workloads. For the Sloan School of Management, we produced a new schedule which was implemented in Fall 2020. The schedule allowed half of Sloan courses to include an in-person component while adhering to safety guidelines. Despite a fourfold reduction in classroom capacity, it afforded two thirds of Sloan students the opportunity for in-person learning in at least half their courses.
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 from others to reduce risk of infection, became widely adopted.

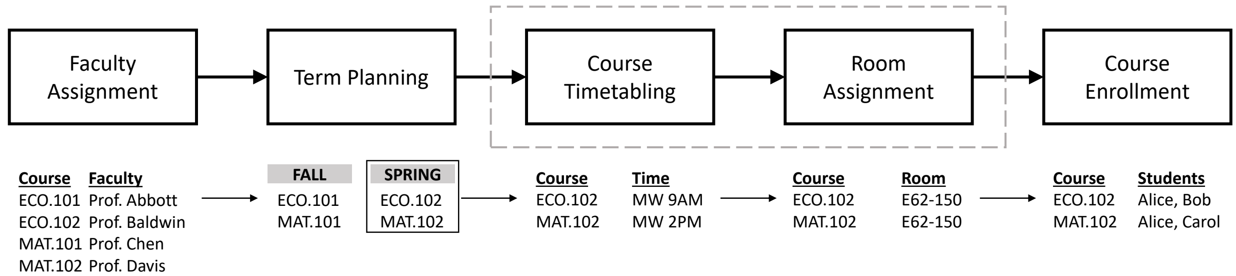
While the decision to close campuses was made quickly, setting policy around when and how to open schools has been challenging because of the fundamental tension between universities' mission of providing high-quality education, and the civic imperative to ensure community safety. Universities cannot operate normally when physical distancing rules dramatically reduce the effective capacity of classrooms, dorms, and labs. This reduced-capacity environment poses new operational challenges, particularly around whether and how to 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, class size increased dramatically, leading to the introduction of a summer term (Farber 1971).

In developing universities' pandemic response, optimization can ensure that newly-scarce resources are used efficiently, particularly when scheduling courses. This 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*). When 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. For such courses, safety considerations compel a complete re-working of both pedagogy and scheduling. New hybrid teaching modes must be offered for students or faculty who cannot or choose not to attend lessons in person; courses with in-person instruction must allow for physical distancing; and cohorts of students must be separated to the extent possible.

These considerations compelled us to analyze creative policy solutions to operate MIT 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 typically incremental year-over-year changes. Our paper discusses the methodology, results, and takeaways from pandemic planning at MIT, and

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; individual universities' procedures are often more complicated in practice, and the order may vary slightly by institution. Feedback at each stage can also influence later stages or prompt changes at earlier stages.

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 course listings for a particular term, the university must choose when and where to offer each course (timetabling and room assignment). Once a schedule is finalized, students may enroll in courses. 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 focuses on developing optimization-based algorithms to the more tractable curriculum-based approach (Mulvey 1982, Tripathy 1984). Defining sets of courses that should not conflict scales well and can model a variety of problem specifics, and has thus 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 formulations (Da Fonseca et al. 2017) to heuristics such as simulated annealing (Abramson 1991) and local search (Da Fonseca et al. 2016). However, modeling power in 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.

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 courses 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 teachers' 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. In comparison with timetabling, relatively less attention has been devoted in recent years to other course scheduling problems. Under the term planning umbrella in Figure 1, Hnich et al. (2002) propose 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. Monette et al. (2007) solve this problem using constraint programming, while Chiarandini et al. (2012) and Ceschia et al. (2014) investigate metaheuristic approaches. Recently, Mühlenthaler and Wanka (2016) consider additional concerns such as fairness in developing multi-term curricula.

In addition, 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 can be used for term planning as well as 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 binary optimization. This choice of methodology enables us to capture a host of both problem specifics both usual (parallel sections, contiguity in time and space, pre- and co-requisites) and novel (limits on campus density, hybrid teaching modes). We chiefly adopt a post-enrollment approach, with some curriculum-based constraints for new courses lacking enrollment data.
- **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 explored a proposal to expand MIT's undergraduate academic calendar from two to three semesters, and constructed a hybrid online and in-person schedule which was implemented at the MIT Sloan School of Management in the fall of 2020.

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 a structural overview, followed by 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 share a two-stage structure: first, courses are scheduled, then students decide which to take. An ideal course schedule (first stage) should allow students to take the courses they want to take, without conflicts (second stage).

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 $c \in \mathcal{C}(s)$ at time t, and 0, otherwise. The problem can thus be written generally as:

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

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

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

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

The function $f(\cdot)$ quantifies the benefit of a particular schedule. For instance $f(\boldsymbol{x}, \boldsymbol{z}) = \sum_{s \in \mathcal{S},\ c \in \mathcal{C}(s),\ t \in \mathcal{T}} z_{s,c,t}$ 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 $\boldsymbol{z}$ 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.

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 that it includes decision variables for each student, required course and possible time, which can lead to scaling issues. We later discuss exact methods and heuristics to alleviate this problem.

Note also that the two-stage structure assumes that students do not change their required courses based on when they are offered (for example, not taking two time-consuming lab courses in the same term, or avoiding early-morning lectures to sleep in). This assumption is likely more reasonable for required courses as compared to elective courses. Furthermore, these effects may be less prominent in aggregate due to students' highly heterogeneous preferences, which we discuss in Section 3.1.

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 consider 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 MIT to consider 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.

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, $\{\text{Fall}, \text{Spring}, \text{Summer}\}$.
$\texttt{pre}(c)$	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	
Course structure	
$\mathcal{J}(c)$	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., $\{\text{Lecture}, \text{Recitation}\}$
$\mathcal{L}(s)$	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 $\ell \in \mathcal{L}$
Time resources	
$\mathcal{D}$	The set of days that courses can be taught; for example, $\{\text{M}, \text{T}, \text{W}, \text{R}, \text{F}\}$
$\mathcal{P}$	The set of all periods when courses can be taught; for example, $\{\text{8am-9:30am}, \text{9:30am-11am}, \ldots, \text{8pm-9:30pm}\}$
$\mathcal{P}(d)$	The set of periods when courses can be taught on day $d \in \mathcal{D}$
M_ℓ	The *multiplicity*, or number of days lesson $\ell \in \mathcal{L}$ repeats each week
N_ℓ	The *length* in periods that lesson $\ell \in \mathcal{L}$ consumes
Space resources	
$\mathcal{R}$	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 $d \in \mathcal{D}$

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 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:

$$x_{c,t} = 1 \text{ if course } c \in \mathcal{C} \text{ is offered in term } t \in \mathcal{T},\ 0 \text{ otherwise;} \tag{2a}$$

$$z^{\text{on}}_{s,c,t} = 1 \text{ if student } s \in \mathcal{S} \text{ can take course } c \in \mathcal{C}(s) \text{ on campus in term } t \in \mathcal{T},\ 0 \text{ otherwise;} \tag{2b}$$

$$z^{\text{any}}_{s,c,t} = 1 \text{ if student } s \in \mathcal{S} \text{ can take course } c \in \mathcal{C}(s) \text{ in any mode in term } t \in \mathcal{T},\ 0 \text{ otherwise;} \tag{2c}$$

$$y_{s,t} = 1 \text{ if student } s \in \mathcal{S} \text{ is on campus in term } t \in \mathcal{T},\ 0 \text{ otherwise.} \tag{2d}$$

Note that the $\boldsymbol{z}$ variables are duplicated into $\boldsymbol{z}^{\text{on}}$ and $\boldsymbol{z}^{\text{any}}$, 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 $\boldsymbol{y}$ 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)):

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

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

$$\sum_{t \in \mathcal{T}} x_{c,t} = B_c \qquad \forall c \in \mathcal{C}, \tag{2f}$$

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, and cannot take too many courses per term:

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

$$\sum_{c \in \mathcal{C}(s)} z^{\text{any}}_{s,c,t} \leq K \qquad \forall s \in \mathcal{S}, t \in \mathcal{T}. \tag{2h}$$

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

$$z^{\text{any}}_{s,c,t} \leq \sum_{t' \in \mathcal{T}: t' < t} z^{\text{any}}_{s,c',t'} \qquad \forall s \in \mathcal{S}, c \in \mathcal{C}(s), c' \in \texttt{pre}(c), t \in \mathcal{T}. \tag{2i}$$

- A student can only take a course on campus if they can take it in any mode, and are also on campus:

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

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

- Total campus density is capped each term, and each student spends enough time on campus:

$$\sum_{s\in\mathcal{S}} y_{s,t} \le Y_t \qquad \forall t\in\mathcal{T}. \tag{2l}$$

$$T_s^{\min} \le \sum_{t\in\mathcal{T}} y_{s,t} \le T_s^{\max} \qquad \forall s\in\mathcal{S}. \tag{2m}$$

We assume students are rational and knowledgeable of pre-requisites, i.e., $c' \in \mathcal{C}(s)$ if $c' \in \texttt{pre}(c)$, $c \in \mathcal{C}(s)$, 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:

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

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. Our experiments use the value $\lambda = 0.1$, which heavily prioritizes on-campus experiences; under this setting, 11 off-campus student experiences are equivalent to one on-campus student experience.

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 course offerings.** It may be desirable to mitigate the disruption of increasing the number of terms in the academic calendar. For example, if a fall-spring calendar moves to a fall-winter-spring calendar, it may be desirable for fall-only courses to stay in either fall or winter. For any such course $c \in \mathcal{C}$, we can add the constraint $x_{\text{SPRING},t} = 0$. Analogous constraints can be added for spring-only courses.
- **Faculty teaching preferences.** If a faculty member typically teaches a fall course $c \in \mathcal{C}$ and a spring course $c' \in \mathcal{C}, c' \ne c$, 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 $x_{c,\text{WINTER}} + x_{c',\text{WINTER}} \le 1$.
- **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) bounding the number of terms students are allowed on campus. In this case, we could change constraint (2f) to $\sum_{t\in\mathcal{T}} x_{c,t} \ge B_c$, and add the expression $\sum_{c\in\mathcal{C}} \left(B_c - \sum_{t\in\mathcal{T}} x_{c,t}\right)$ to the objective with a suitable penalty. As Formulation (2) is a maximization problem, this change would penalize any additional offerings of courses beyond those of a normal year.
- **Co-requisites.** Constraint (2i) 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 $t' \le t$ 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*.

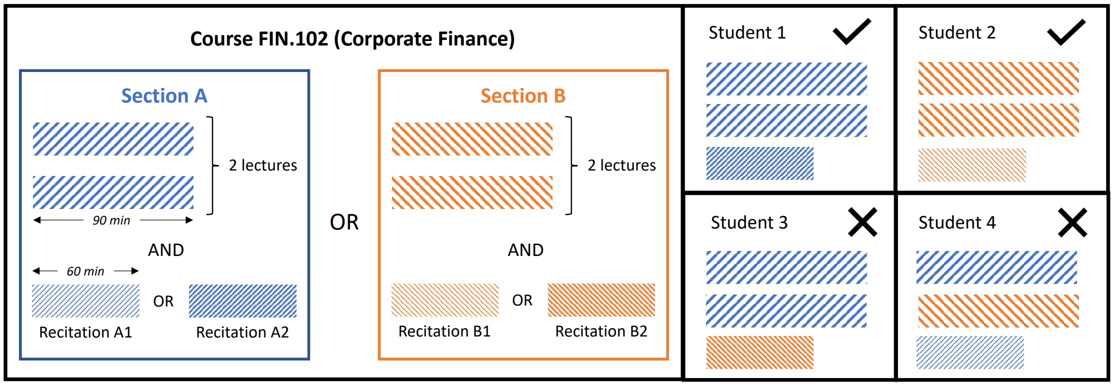
The concept of lesson multiplicity may seem overwrought. Why not simply decompose the course into three lessons, two of length 3 (90 minutes) and one of length 2 (60 minutes)? First, there is a greater conceptual difference between lecture and recitation, which differ by length and instructor as well 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 (for example, scheduling lectures at the same time in the same room, without constraining recitation).

Some popular courses are taught in multiple parallel tracks, possibly taught by different faculty. We define a *section* as a set of lessons that together fulfill all teaching components of a particular course c. We can think of 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 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

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.

r, and write $\mathcal{B}(\ell)$ as the set of blocks in which lesson ℓ can be held. To model the possibility of a course being taught online, we also add to the set of blocks $\mathcal{B}$ a fictitious "online" block with zero in-person capacity. The fictitious online block will always be included in the set $\mathcal{B}(\ell)$ for any lesson ℓ, thereby allowing all lessons to be scheduled even if classrooms are scarce.

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, students are unlikely to consider attending lessons any less than once a week an adequate in-person experience. 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 $\boldsymbol{x}$ determine course schedules, and the variables $\boldsymbol{z}$ correspondingly determine student schedules, taking into account the refinements mentioned above:

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

$$z^{\text{on}}_{s,\ell,d,p} = 1 \text{ if student } s \in \mathcal{S} \text{ starts lesson } \ell \in \mathcal{L}(s) \text{ on campus on day } d \in \mathcal{D}, \text{ period } p \in \mathcal{P}(d),$$

$$0 \text{ otherwise;} \tag{4b}$$

$$z^{\text{any}}_{s,\ell,d,p} = 1 \text{ if student } s \in \mathcal{S} \text{ starts lesson } \ell \in \mathcal{L}(s) \text{ in any mode on day } d \in \mathcal{D}, \text{ period } p \in \mathcal{P}(d),$$

$$0 \text{ otherwise;} \tag{4c}$$

$$y_{s,d} = 1 \text{ if student } s \in \mathcal{S} \text{ is allowed on campus on day } d \in \mathcal{D},\ 0 \text{ otherwise.} \tag{4d}$$

We further define the following auxiliary variables:

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

$$\phi_{\ell,p} = 1 \text{ if lesson } \ell \in \mathcal{L} \text{ starts in period } p \in \mathcal{P} \text{ on any day},\ 0 \text{ otherwise;} \tag{4f}$$

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

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 precludes the need for 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)):

$$z^{\text{any}}_{s,\ell,d,p} \le \sum_{b \in \mathcal{B}(\ell)} x_{\ell,b,d,p} \qquad \forall s \in \mathcal{S}, \ell \in \mathcal{L}(s), d \in \mathcal{D}, p \in \mathcal{P}(d). \tag{4h}$$

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

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

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.

We calculate $\hat{Q}_{b,\ell}$ as follows. For the fictitious online block, in-person capacity is zero. For a physical block b, there is no need to rotate if it can fit all students for lesson ℓ. Otherwise, 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 can attend lessons in-person on any given day. For each lesson ℓ for a course c, the maximum number of students that can attend is given by $S_\ell := |\mathcal{S}(c)|/|\mathcal{J}(c)|(1+$

ϵ), 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:

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

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:

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

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

  Recall that the sets $\mathcal{B}(\ell)$ include the fictitious online block.
- Each lesson is scheduled at the same time over the course of the week:

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

$$\sum_{p\in\mathcal{P}} \phi_{\ell,p} = 1 \qquad \forall \ell\in\mathcal{L}, p\in\mathcal{P}. \tag{4m}$$

- Multiple lessons cannot consume room or faculty resources concurrently:

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

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

  Note that because the fictitious online block is not associated with any real classrooms, constraint (4n) still allows multiple lessons to be taught simultaneously online.
- Lessons cannot continue into forbidden time periods (e.g. lunch or end of day):

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

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:

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

  If a student only needs to attend one of the lessons $\ell_1,\ldots,\ell_k$ (for instance, one of two recitations in Fig. 2), we can modify constraint (4q), summing the left-hand side over the lessons $\ell_1,\ldots,\ell_k$.

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

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

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

$$\sum_{\ell\in\mathcal{L}(s)} \sum_{p'=p-N_\ell+1}^{p} z^{\text{any}}_{s,\ell,d,p'} \le 1 \qquad \forall s\in\mathcal{S}, d\in\mathcal{D}, p\in\mathcal{P}(d). \tag{4s}$$

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

$$z^{\text{on}}_{s,\ell,d,p} \le z^{\text{any}}_{s,\ell,d,p} \qquad \forall s\in\mathcal{S}, \ell\in\mathcal{L}(s), d\in\mathcal{D}, p\in\mathcal{P}(d). \tag{4t}$$

$$z^{\text{on}}_{s,\ell,d,p} \le y_{s,d} \qquad \forall s\in\mathcal{S}, \ell\in\mathcal{L}(s), d\in\mathcal{D}, p\in\mathcal{P}(d). \tag{4u}$$

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

$$\sum_{s\in\mathcal{S}} y_{s,d} \le Y_d \qquad \forall d\in\mathcal{D}. \tag{4v}$$

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

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

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

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

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 take into account desirability of teaching times. For days and periods where students or faculty are completely unavailable, the relevant 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 10am to 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 maximize satisfied course demand without conflicts; then minimize schedule gaps, subject to the constraint of not introducing additional conflicts; and finally minimize the number of days on campus, subject to the constraint of not introducing any additional conflicts or gaps.

Feasibility. The timetabling problem has so many constraints that practitioners may reasonably worry about infeasibility. In our application, we are guaranteed to obtain a feasible schedule because (i) any course can be scheduled online, and (ii) scheduling conflicts for individual students manifest in objective deterioration rather than infeasibility. In the absence of the online option, a feasible schedule may not exist. In this case, a reasonable approach is to follow Stallaert (1997) and identify constraints that are not absolutely necessary, but perhaps merely nice to have. For example, when we created a schedule for the MBA core courses, which we describe in Section 3.3.1, classroom space was so scarce that we had to allow some lessons to take place in different rooms across the week by relaxing constraints (4k). Otherwise, it would not have been possible to host all of the core courses in-person. By contrast, the capacity constraints (4i) and (4v) were prioritized as hard constraints, in order to ensure community safety and adherence to regulation.

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 is dimensionality reduction: instead of allowing lessons to start in any period, we only allow them to start at a few designated *starting periods.* More precisely, 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 also produces more legible schedules, so lessons only rarely deviate from this pattern in the usual manual timetabling process at MIT Sloan.

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 courses, 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 approach can significantly reduce the feasible solution space; as such, we use it sparingly and model students individually unless otherwise noted.

We note that though we consider term planning and course timetabling as separate but related problems, the formulations presented here could be adapted to solve the two problems concurrently, by adding a term index to the timetabling variables. However, such treatment would lead to a profusion of decision variables, and is beyond the scope of this paper.

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-term 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 pre-register while campus policy, including the schedule, remained in flux, we used enrollment data from the last academic year as a proxy for the upcoming year. A limitation is that students' required courses may themselves change according to the schedule, but lacking a model of such changes, we opted to simply use last year's data.

For the term planning problem, the MIT Registrar provided us with: (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. We then needed to identify the required courses $\mathcal{C}(s)$ for each student $s \in \mathcal{S}$. For a physics major, for example, these might include physics along with relevant mathematics courses.

We associated each major at MIT with a list of "common major courses," i.e., courses taken by at least 10% of historical students in that major. Historical students were taken from the graduating class years of 2019 and 2020, so newer courses would be included. Our 10% threshold was conservative, as we preferred including some less-relevant courses to missing important electives: for example, many of our science and engineering course lists included music courses in addition to expected requirements, reflecting the prevalence of music minors. 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).

We also added any unfulfilled prerequisites to each students' required courses, meaning that we actually attempted to schedule more courses than strictly necessary. We made this addition because we did not know whether students had skipped prerequisites voluntarily or due to past scheduling issues; we therefore opted to be conservative, understanding that then even MIT's original two-semester schedule would not fulfill 100% of student-courses in our computational results.

Our term planning data set consisted of 4,287 undergraduates enrolled across 620 courses and showed heterogeneous preferences, with these students enrolled in 4,054 unique sets of courses. First-years had the highest number of required courses (6.57), while seniors had the lowest (3.40), reflecting the natural advancement of undergraduate coursework. We capped the number of required courses each student could take per semester at either three courses or half of their per-year total, whichever was greater. Added prerequisites did not count towards this cap.

For timetabling, Sloan leadership provided us with: (1) courses and associated faculty for the upcoming semester (Fall 2020); (2) course enrollments from Fall 2019; (3) available classrooms, with capacities accounting 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, then sampled students uniformly from these programs and added them to the new courses. Our final timetabling data set consisted of 1,455 students across 114 courses, with 5.33 courses per student on average; as well as 115 faculty members and 26 classrooms with a mean capacity of 19.46 students. Student preferences were heterogeneous; among the 1,455 students, there were 763 unique sets of required courses. Thus, any conflict between a pair of courses previously taken together, even by a single student, would be penalized in the objective.

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 proposed altering its academic calendar by increasing the number of semesters, both students and faculty expressed concerns. Students worried that important courses might be offered during their time off-campus, limiting their ability to learn from faculty and each other. Faculty wondered whether additional course offerings would be needed, increasing teaching loads.

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 combination of shared requirements (e.g. Organic Chemistry) and laboratory needs made the departments of Chemistry (CHM), Biology (BIO), Chemical Engineering (CHE), and Biological Engineering (BIE) a motivating use case for our model. Contrary to standard MIT practice, and for clarity, we label departments using three-letter abbreviations (e.g. BIO) instead of numbers (e.g. 7 for Biology).

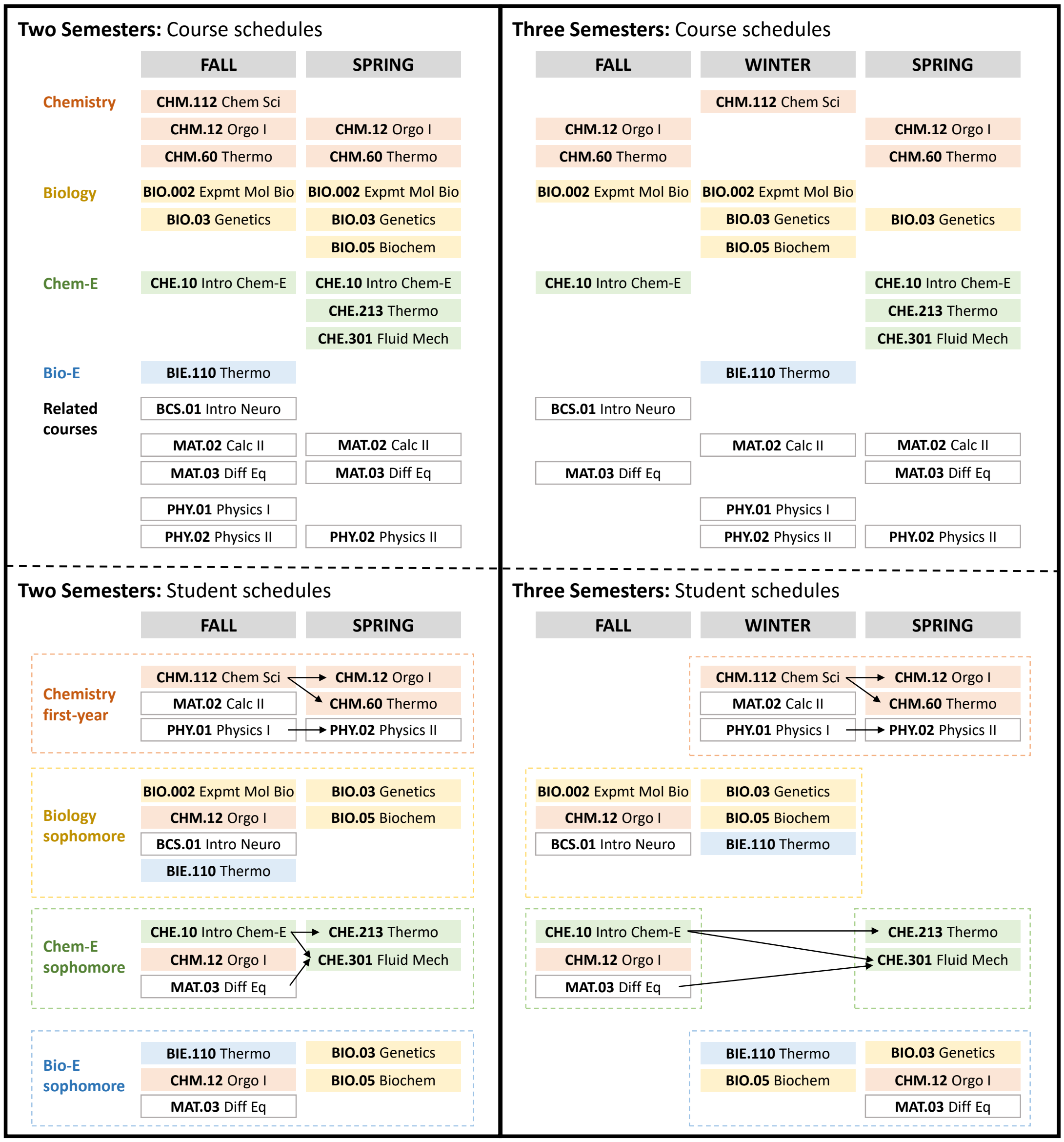
The top left panel of Figure 3 shows the existing two-semester calendar for some courses across the four departments of interest, as well as relevant courses in related departments. Courses appear under the column headers of terms when they are offered; note that some courses, such as CHM.12 (Organic Chemistry I), are offered in both fall and spring. The bottom left panel shows typical student schedules under the usual calendar. Again, courses appear under the column headers of the terms when they are taken.

The top right panel of Figure 3 shows the proposed three-semester calendar, 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, and 75% in the winter and spring. The bottom right panel shows the student schedules induced by the three-semester calendar. Many student schedules are simply translated across semesters in order to preserve prerequisite relationships. Notably, each student's coursework is confined to only two of the three semesters. This is achieved thanks to redundancy in MIT's existing schedule; for example, without the second offering of CHM.12, all students would need to be on-campus in the semester of its single offering.

For the relatively small number of courses and students in Figure 3, it is possible to manually find a working three-term schedule. However, with larger numbers of courses and students at the scale of an entire university, optimization becomes critical. We now 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, 31 students), then added students from the related departments of Figure 3 (246 courses, 385 students). At each step, we added

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



Note. Sample courses from the 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 that 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 are indicated by black arrows.

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

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

Courses	Students	Variables	Constraints	Objective	Running Time (min)
115	31	2,664	5,124	247	0
246	385	26,808	52,577	2,579	3
372	787	45,729	92,401	4,699	11
479	1,682	97,362	198,267	10,271	133
620	4,287	241,917	514,212	26,854	720

Time limit of 720 minutes, gap tolerance of 0.5%; the last example achieved a gap of 2.4%.

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 smaller instances within a few minutes, 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 a three-semester schedule on the MIT undergraduate experience. We considered several policies with either two or three semesters. The baseline two-semester policies allowed each student on campus for exactly one of two semesters, with campus capacity not exceeding 50%. Under the three-semester policy, 50% of students attend in the fall and 75% attend in the winter and spring, with each student on-campus for two semesters and off-campus for one semester.

We also considered several partitions of students across semesters. The first partition was fully flexible and decided exactly which individual students to invite on campus each semester. The second partition invited students in the same major and year to campus in the same semesters. The final partition enforced exactly which class years would be invited each semester as follows. Under a two-semester calendar, first-years and sophomores would attend in the fall, and juniors and seniors in the spring. Under a three-semester calendar, first-years and sophomores would attend in the fall; sophomores, juniors, and seniors in the winter; and first-years, juniors, and seniors in the spring. The latter two partitions were considered more actionable relative to the flexible partition.

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. 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, mostly due to the additional prerequisites added to each students' required courses.

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

Time limit: 720 min. 3-term flexible and major-year opt. gaps were 2.4% and 5.1%.

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 town halls and focus groups. Although our model's performance was convincing, there was concern that our approach would be too difficult in what had already been a year of unprecedented disruption. As such, MIT decided to keep the two-semester calendar. Only seniors were invited to campus in the fall, and undergraduate courses were held online with limited in-person teaching when necessary (Song 2020).

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 (100 students). However, Sloan needed to accommodate the diverse needs of a graduate body of over 1,000 students, located across the globe because of the pandemic.

For many Sloan students, a dynamic classroom experience is a key attraction, as they interact with faculty and classmates in discussion-based courses. Therefore, it was feared that a fully online experience could fall short of students' expectations, particularly for first-year students who had never met in person. Through the spring of 2020, many incoming students voiced strong interest in maintaining in-person education. The challenge faced by the school in the early summer of 2020 was thus to develop a course timetable that would enable students to fulfill their degree 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 a 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 courses 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, 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). To mitigate the lengthened calendar, faculty could specify unavailability 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. Further, certain large classrooms were not directly managed by Sloan, but instead shared across multiple MIT departments. We refer to these classrooms as "expanded capacity," as compared to the "base capacity" corresponding to classrooms 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 and was planning for a class size of 490 students. Before choosing when to schedule courses for each cohort, it was thus necessary to determine how many cohorts to create. A large cohort might not fit in any room block, while many small cohorts could overwhelm classroom and teaching resources.

We solved the timetabling and room assignment problem for different numbers of first-year cohorts to evaluate the scheduling implications. Since each cohort attends one section of each core course, the number of cohorts can be increased by adding a section to the set of available sections $\mathcal{J}(c)$ for each core course c. Each new section j then has associated lessons $\mathcal{L}(j)$ that must be scheduled. But, since the size of each section would be smaller, each lesson ℓ would have more available room blocks $\mathcal{B}(\ell)$ in our formulation, adding flexibility to the optimization problem.

For each number of cohorts, 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 constrained students to occupy at most one room block per day to prevent crowding during room switches, and we relaxed the constraint confining repeating lessons to the same room block.

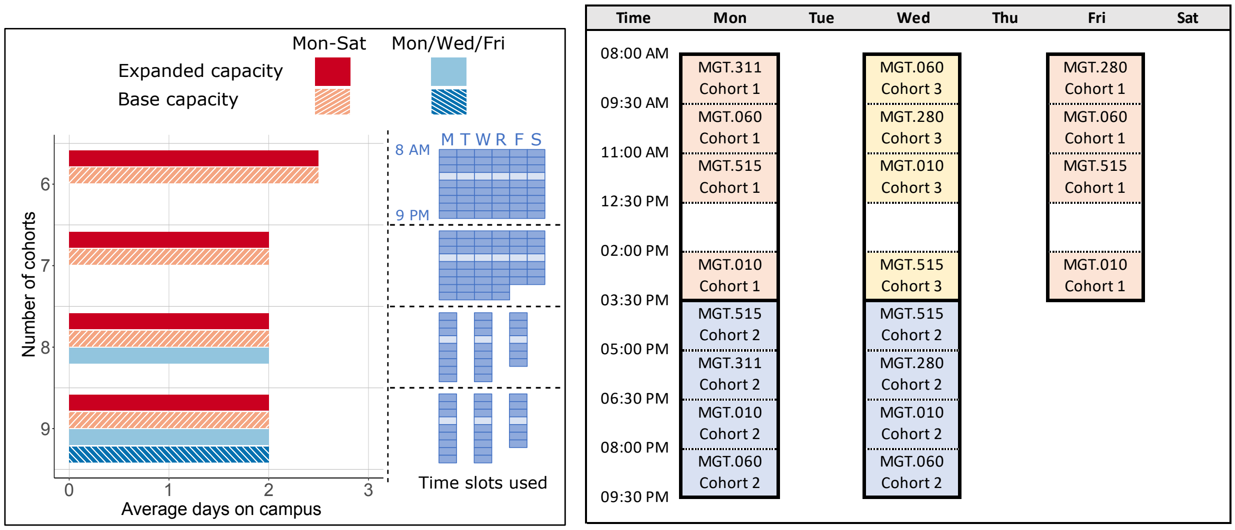
The largest instance of the core timetabling model had 60,582 variables and 314,271 constraints. Each of the first two steps (minimizing schedule conflicts and gaps in student schedules) took less than one minute to solve. The last step of minimizing the number of days each student spends on campus generally took longer, but still terminated with an optimal solution in seven minutes. Because all first-year MBA students take the same core courses, in this section we take the super-students approach as described in Section 2.4, with each section corresponding to a super-student with size equal to the size of the section.

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, reserving Tuesday, Thursday and Saturday for other programs. 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 4a. 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 4b 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 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 lessons are completely contiguous in time (excluding lunch), facilitating physical distancing on campus.

Figure 4 **Selecting the number of MBA first-year cohorts with scheduling and room usage implications.**



(a) Effect of number of cohorts

(b) Example first-year MBA room usage

Note. The left panel analyzes the scheduling implications of the number of MBA first-year cohorts. On the left, we show the average number of days per week students spend on campus (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. The right panel presents the usage of an example block comprising two classrooms, for nine cohorts under a three-day scheduling model. Cohorts 1 (red) and 2 (blue) stay entirely in the two classrooms. Half of the lectures for Cohort 3 (yellow) are not shown; on Friday, this cohort moves to another block so as not to conflict with Cohort 1.

3.3.2. Other MIT Sloan courses. We now discuss courses outside the first-year MBA core, which we call "non-core" courses for brevity. Each course could specify that either lecture or recitation should be in-person or online, or that either mode was allowable. By default, since teaching assistant (TA) preferences were unknown, we set recitations online unless otherwise requested. Fully online courses included MBA electives, which have substantial cross-enrollment from first- and second-year students; courses with significant undergraduate enrollment due to institute-wide directives; and courses whose instructors indicated in a survey they would not find a physically-distanced classroom effective. Overall, 47 out of 109 non-core courses were set to be purely online.

We limited in-person lectures to Tuesday, Thursday, or Saturday, with other days reserved for core courses. As mentioned in Section 2.4, we managed the scale of the problem by solving first for time periods, then rooms. We could not guarantee classroom allocations, especially in cases of large enrollments, but we increased objective weights on $\boldsymbol{z}^{\text{any}}$ for program requirements and courses with faculty committed to teaching in-person for the semester. We 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, achieving optimality gaps of at most 2%. Solving for classrooms took several seconds, while solving for time periods consumed the full time limit. The full model had 1.5 million variables and 1.3 million constraints, but each step fixed large subsets of the variables: the largest instance after presolve had 75,258 variables and 26,599 constraints.

Our end result was a schedule where 97% of students' non-core lectures could be scheduled without conflicts. Moreover, 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 some in-person component. These metrics are likely underestimates, since they ignore the fact that students seeking in-person learning can adjust their course selection accordingly. In accordance with public health guidelines, our schedule also kept campus density low. Mondays, Wednesdays, and Fridays had the lowest number of students (220 to 385 per day), since they were reserved for first-year MBAs. On Tuesdays, Thursdays, and Saturdays, the number of students ranged from 374 to 557, not exceeding a target of 600 people on campus including faculty and staff.

A summary of teaching modes for all courses in our optimized schedule is presented in Table 4. For comparison, it also includes the teaching modes that would have been used under the status quo schedule. This baseline was virtually unimplementable; the vast majority of courses, including MBA core courses, would have been forced online because of infeasible room assignments. By contrast, our optimized schedule offered about half of the courses with some in-person component, no mean feat given that campus capacity had decreased to about a quarter of its pre-COVID-19 levels. We can see from Table 4 that the new teaching modes of assigning courses to multiple simultaneous nearby classrooms, or having students rotate their attendance, saw more use in larger courses. These teaching modes, combined with the optimized decision-making, were critical for enabling more in-person course offerings compared to the status quo.

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 courses 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

Table 4 Teaching modes for all MIT Sloan courses for the schedule planned before the pandemic ("Status Quo"), and the schedule we produced ("Optimized").

Schedule	Student Attendance	Classrooms	Courses	Students per Section
Status Quo	In-Person	One classroom	13	9.3
	In-Person	Multiple classrooms	0	–
	Rotating Capacity	One classroom	11	34.0
	Rotating Capacity	Multiple classrooms	0	–
	Fully Online	–	90	45.2
Optimized	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

The first column indicates whether students in the courses have full access to 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 teaching mode.

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. Beyond modeling, implementing the schedule relied on an iterative process of community feedback. This process was necessary because course timetables are not typically constructed from scratch, and faculty preferences are usually not communicated in a standardized way. For instance, part-time faculty members typically work with area heads to identify teaching times compatible with external commitments, and faculty may coordinate informally with colleagues to accommodate child care or commuting routines. Some special situations were expressed in an initial faculty survey, then a draft schedule was made available for open comments. We then partially re-optimized the schedule, adjusting only courses for which concerns were raised. Given the final schedule, a few one-off scheduling changes were handled manually, by enumerating possible moves and choosing the one with the smallest cost.

While developing our methods, we benefited from frequent meetings with faculty, academic staff, and administrators. These experts reviewed our proposals and identified design flaws, which we could then promptly address. In these discussions, visualization proved to be a crucial tool, particularly for staff members accustomed to visually inspecting course timetables and room assignments.

Modeling limitations. The feedback process was particularly important, because it allowed the final schedule to overcome the limitations of our modeling assumptions. For instance, we relied on academic administrators to understand likely enrollment patterns for new courses, or courses recently designated as program requirements, that may not have been captured in historical data.

More generally, 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. This simply frees up more on-campus capacity for students who desire an on-campus component.

We also do 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 these students are relatively few, and this effect can be mitigated through more creative multi-week rotation schedules or other case-by-case adjustments.

Finally, as discussed in Section 2, a key assumption of our models is that student interest in a course is not affected by the time this course is offered, as long as it does not conflict with other courses the student wants to take. Given recent interest in mechanism design for course selection (Budish et al. 2017), probing this assumption empirically and adjusting course selection mechanisms accordingly could be an exciting new research direction.

4. Conclusions

In this paper, we introduced a two-stage view on course scheduling, encompassing several existing problems in the operations research literature. Our formulation can accommodate a variety of problem-specific constraints, and we demonstrate its efficacy, as well as its limitations, on two case studies at MIT. In particular, we evaluated the following new policies:

- **From two to three semesters.** We showed that 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 to reduce disruption to students and faculty.
- **Partitioning of class years.** At the Sloan School of Management, we demonstrated that a moderate increase in teaching staff would allow first- and second-year MBA students to attend courses 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.** We constructed a schedule which was implemented at the Sloan School of Management in the fall of 2020, incorporating hybrid in-person and online instruction while adhering to safety standards. Despite a fourfold reduction in classroom capacity, the schedule allowed a safe in-person component in half of Sloan courses, and granted two out of every three students the opportunity for in-person learning in at least half of their courses.

Our work shows 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, Di Gaspero 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.

Hnich B, Kiziltan Z, Walsh T (2002) Modelling a Balanced Academic Curriculum Problem. *Proceedings of the Fourth International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems*, 121–131.

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.

Monette JN, Schaus P, Zampelli S, Deville Y, Dupont P, et al. (2007) A CP approach to the balanced academic curriculum problem. *7th International Workshop on Symmetry and Constraint Satisfaction Problems*, volume 7 (Citeseer).

Mühlenthaler M, Wanka R (2016) *Fairness in academic course timetabling*, volume 239. ISBN 9783319127989.

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.

Stallaert J (1997) Automated timetabling improves course scheduling at UCLA. *Interfaces* 27(4):67–81.

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.

Course Scheduling under Sudden Scarcity: Appendix

Our timetabling formulation (4) can accommodate specialized constraints, examples of which follow.

Repeating lesson patterns. It may be desirable to spread the sessions of a repeating lesson across the week at a particular cadence (e.g., to allow students a day between lectures to review material). To prevent the sessions of a repeating lesson ℓ from being spaced apart by k days, we can write:

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

For instance, we can apply this constraint with $k = 1$ to prevent repeating lessons on consecutive days.

International time zones. COVID-19 travel restrictions kept many international students from traveling abroad. It was therefore important to offer core courses during periods acceptable for multiple time zones. To ensure that course $c \in \mathcal{C}$ starts in a pre-specified set of acceptable periods $\mathcal{P}_{\text{INTL}}$, we write

$$\sum_{p \in \mathcal{P}_{\text{INTL}}} \phi_{\ell,p} \geq 1, \tag{1b}$$

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 for at least one section. The constraints can also 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. In such cases, it may be desirable for students to attend all or most of their courses in the same rooms so as to reduce unnecessary movement within the building. To this end, we introduce auxiliary binary variables $\boldsymbol{\eta}$, such that $\eta_{s,d,r} = 1$ if student $s \in \mathcal{S}$ uses room $r \in \mathcal{R}$ on day $d \in \mathcal{D}$ (0 otherwise). We impose that a student uses a room if they attend any lesson in that room, then cap the number of rooms used by a student each day:

$$\sum_{b \in \mathcal{B}(r)} x_{\ell,b,d,p} \leq \eta_{s,d,r} \qquad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d), \ell \in \mathcal{L}(s). \tag{1c}$$

$$\sum_{r \in \mathcal{R}} \eta_{s,d,r} \leq R^{\max} \qquad \forall s \in \mathcal{S}, d \in \mathcal{D}, \tag{1d}$$

where $R^{\max}$ designates the maximum number of rooms used per student per day. Alternatively, we can penalize the sum of the $\boldsymbol{\eta}$ variables in the objective. We impose this constraint on rooms instead of blocks to acknowledge the difference between blocks that share a room and blocks that do not. To avoid overly constraining the problem, we can relax constraint (4k), allowing lessons to occur in blocks of comparable size instead of the same block.

Gaps in student schedules. Another scheduling priority may be to reduce gaps, i.e., idle periods between courses. While simply a convenience in normal times, avoiding schedule gaps is important in a reduced-capacity pandemic setting. Following Da Fonseca et al. (2017), we can introduce auxiliary variables $\boldsymbol{\beta}$ such that $\beta_{s,d,p} = 1$ if student $s \in \mathcal{S}$ is busy on day $d \in \mathcal{D}$, at period $p \in \mathcal{P}(d)$ (0 otherwise), where "busy" means either attending a lesson or between lessons, i.e.,

$$\sum_{\ell \in \mathcal{L}(s)} \sum_{p'=p-N_\ell+1}^{p} z^{\text{any}}_{s,\ell,d,p'} \leq \beta_{s,d,p} \qquad \forall s \in \mathcal{S}, d \in \mathcal{D}, p \in \mathcal{P}(d), \tag{1e}$$

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

The max operators on the left-hand side of constraint (1f) can be linearized Bertsimas and Tsitsiklis (1997). We can then de-incentivize gaps by penalizing the sum of the $\boldsymbol{\beta}$ variables in the objective.

References

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization* (Athena Scientific Belmont, MA).

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.