Hybrid Scheduling with Mixed-Integer Programming at Columbia Business School*

Ciamac C. Moallemi
Columbia University
ciamac@gsb.columbia.edu

Utkarsh Patange
Columbia University
utkarsh.patange@columbia.edu

Current Revision: March 2023

Abstract

We describe the hybrid scheduling system that we implemented at Columbia Business School during the Covid-19 pandemic. The system allows some students to attend in-person classes with social distancing, while their peers attend online, and schedules vary by day. We consider two variations of this problem: one where students have unique, individualized class enrollments, and one where they are grouped in teams that are enrolled in identical classes. We formulate both problems as mixed-integer programs. In the first setting, students who are scheduled to attend all classes in person on a given day may, at times, be required to attend a particular class on that day online due to social distancing constraints. We count these instances as "excess." We minimize excess and related objectives, and analyze and solve the relaxed linear program. In the second setting, we schedule the teams so that each team's in-person attendance is balanced over days of week and spread out over the entire term. Our objective is to maximize interaction between different teams. Our program was used to schedule over 2,500 students in student-level scheduling and about 790 students in team-level scheduling from the Fall 2020 through Summer 2021 terms at Columbia Business School.

1. Introduction

The Covid-19 pandemic ushered in an era of online events. After spending some time in a fully online mode, Columbia Business School (CBS) decided to start a hybrid mode of instruction wherein a set of students are invited to attend in-person classes, while others attend online. The intention was to allow MBA students an opportunity to network and extract maximum benefits from their program. The in-person attendance capacity was limited due to social distancing guidelines. We therefore assign each student a mode of attendance for each of their classes: either in person, or online. Since many students travel to campus from afar, if a student was scheduled to attend one class in person, it was desirable to allow them to attend all their classes scheduled on that day in person as much as possible. Furthermore, Each student deserved an equal opportunity to attend classes in person an equal number of times. In this paper, we discuss this hybrid scheduling as it was implemented at Columbia Business School starting Fall 2020 through Summer 2021.

At first glance, there are several ways to implement such a schedule. For example, a customized schedule could be designed for each student that lists the days that student should attend classes in

*We thank Jonah Rockoff and Rachel Horton for helpful discussion and guidance. We also thank anonymous reviewers for their suggestions that have improved the paper.

person. This would allow considerable flexibility in terms of many relevant objectives that we care about, but would be hard to implement as each student would need to consult their personalized schedule every day, and planning any group activity would need to take into account the combined schedules of everyone in that group. Our objective is to come up with a solution that is easier to implement and communicate, but could be more restrictive. We consider two variations of this problem for different student populations and address each with a separate scheduling approach to better fulfill the needs of the respective populations.

Second-year students at Columbia can have widely varying schedules. Here, we prioritize simple implementation over other objectives. We divide the students into groups, and cycle the groups for in-person attendance throughout the term. While this means that students from one group never get to interact with those from other groups in person, we are willing to allow this loss of interaction in exchange for ease of implementation. An advantage of this approach is that other in-person events can now be arranged using these same groups. For example, suppose that the students are divided into three groups, and that an organizer is organizing an event for all students to attend in person. Such an organizer can schedule this event for three consecutive days. Then, the students scheduled to attend classes in person on these days can attend this event in person as well.

For the first-year students, who are new to the school, networking is arguably more important than it is for second-year students. First-year students at Columbia are divided into "clusters" with identical schedules. Each cluster is further subdivided into learning teams that are expected to spend a significant amount of their time together working on group assignments, etc. Since each cluster has the same number of learning teams, we can schedule these students by their learning teams, and each cluster can follow the same schedule by learning team. We can then try to maximize networking between learning teams.

We refer to the two formulations above as student-level and team-level scheduling, respectively.

1.1. Contributions

For student-level scheduling, as discussed above, our solution involves assigning each student to one of M groups, where M is predetermined, and scheduling the groups one by one to attend classes in person on each day of the term. This ensures that each student attends classes in person for exactly a fraction $1/M$ of the time, which is fair since it is the same for all students. In some cases, even if only one group is assigned to attend the day's classes in person, there may be some classes where in-person attendance would exceed the social distancing capacity. In such a case, we assign the excess students at random to attend those classes online. We formulate the problem of group assignments as an optimization problem that minimizes the total excess (TE), summed over all groups and courses. Because the excess students are still be on campus on their scheduled days, the school needs to provide a separate room from which they can attend the class online. Again, due to social distancing considerations, this room is limited in the number of students it can house simultaneously, making it imperative to minimize the amount in which the simultaneous excess (SE), i.e., the maximum excess summed across all classes at the same time, exceeds this limit. We define the amount of SE over the excess-room capacity as the surplus simultaneous excess (SSE). Finally, we should avoid any imbalance between the number of students attending a class in person across different groups. We define total deviation (TD) as a measure of this imbalance and also minimize this quantity. In summary, we make the following contributions to the student-level scheduling problem:

- We present a novel formulation of the problem as a mixed-integer program.
- We analyze the resulting program and, in particular, present a closed-form solution to the relaxed linear program.
- As a preliminary validation, we use our optimization approach to conduct counterfactual group assignment experiments on data before the Covid-19 pandemic (Fall 2019) and present the results in Section 2.3.
- We present empirical results from the application of our method to second-year MBA students at Columbia Business School. In Fall 2020, we divided 1,302 second-year students enrolled in 67 hybrid classes into 3 groups according to our optimization method to get a TE of 20 and an SE of 5. In Spring 2021, we divided 1,035 students enrolled in 61 hybrid classes into 3 groups to get a TE of 0 and an SE of 0. The numbers were much smaller in Summer 2021: we had 226 students enrolled in 8 hybrid classes that we divided into 2 groups to get a TE of 12 and an SE of 4. In all terms, the excess room capacity was 140, and so the SSE was 0. We report the final results for Fall 2020 through Summer 2021 in Section 2.3

For team-level scheduling, as discussed earlier, each learning team is assigned to attend classes in person or online together for each day of the term. For first-year students, networking is an important part of the MBA experience, and so we schedule to allow students to meet with as many other students as possible while attending classes together. We make the following contributions to the team-level scheduling problem.

- We present a novel formulation of the problem as a mixed-integer program. Our program tries to maximize networking opportunities between students while satisfying fairness criteria that we develop.
- We present empirical results from the application of our method to first-year MBA students at Columbia Business School. Our method was used for hybrid scheduling of 570 first-year students in Fall 2020 and 220 first-year students in Spring 2021.

For both of these problems, we used the Gurobi optimization software (Gurobi Optimization, LLC, 2022) version 9.5.1 and Python version 3.9.7 to solve the mixed-integer programs.

1.2. Other Applications

Aside from the present setting, our models can be used in other applications as well. The student-level scheduling model can be useful in “mode selection” problems where participants can be assigned to one of two modes, but there is a capacity constraint on the preferred mode. An example is traffic rationing. Guerra et al. (2022) review the simple license-plate-based driving restrictions that are employed in many cities around the world either for short-term or long-term traffic rationing to combat congestion, reduce pollution, and minimize fuel consumption. From the perspective of minimizing congestion, such restrictions are inefficient, however, since different motorists tend to use different sets of roads. An alternative to such restrictions can be a more complex individual restriction scheme. Under this scheme, each driver would submit a source (e.g., residence) and a destination (e.g., workplace) for their daily commute. We then compute the geographical regions each driver travels through during their commutes. The drivers are then treated as students and the regions as courses to apply our student-level scheduling model. Once drivers are assigned to a group, groups are scheduled to drive using their own vehicle in a rotational manner. Those not scheduled to drive are required to take public transit or face penalties. The penalties could be administered using technology currently employed by electronic toll collection networks throughout the world.

The team-level scheduling model can be applied to worker assignment to remote versus online modes on a day-to-day basis. Tsipurkey (2023) observe that only 6% of the remote-capable workers desire a traditional office. Average office use in 10 major US cities is barely above 50% of its pre-Covid level as of February 2023 (Pisani and Rhone, 2023). Some companies are already moving towards desk sharing (see, for example Elias, 2023) to save on office real-estate needs. Deciding who works online and who comes to office in person is a mode selection problem. These workers are often divided into teams, and for maximum intra-team collaboration, it is desirable to schedule everyone in the same team together. Thus our team-level scheduling model can be applied, allowing a planner to schedule a subset of teams to work from office. Our objective ensures that inter-team interaction is maximized.

1.3. Literature Review

Educational scheduling literature can be broadly divided into three categories: timetabling, course allocation, and classroom allocation. Timetabling deals with assigning time slots to various educational activities, include teaching sessions or assessments. By far, most of the scheduling literature deals with timetabling; Babaei et al. (2015) and Bettinelli et al. (2015), survey this area. Solutions have been proposed for this problem include the work of Burke and Bykov (2016); Bellio et al. (2016); Asín Achá and Nieuwenhuis (2014); Abdullah and Turabieh (2012); Mirhassani (2006). On the empirical side, schools have analyzed and applied mixed-integer programming to solve timetabling problems in their specific situations (e.g., García-Sánchez et al., 2019; Tripathy, 1984; Stallaert, 1997).

Course allocation is the problem of assigning courses to students or teachers. In many cases, a course is divided into smaller sections for manageability, and students and teachers are assigned to these sections. We consider these to be instances of the course allocation problem. Optimization has been applied to this problem as well (e.g., Yekta and Day, 2020; Budish et al., 2017; Dinkel et al., 1989; Kannan et al., 2012). On the applied side, this problem has been studied along with timetabling (e.g., Hinkin and Thompson, 2002) and classroom allocation (e.g., Gonzalez et al., 2018; Martin, 2004; Kassa, 2015).

Classroom allocation, the problem of assigning appropriate classrooms to educational activities, has also been studied individually (e.g., Phillips et al., 2015; Carter and Tovey, 1992), and in tandem with timetabling and course allocation (e.g., Martin, 2004; Kassa, 2015; Strichman, 2017; García-Sánchez et al., 2019).

Closer to our work are papers that deal with challenges posed by the Covid-19 pandemic. One line of literature broadly considers various different problems experienced by different schools related to the pandemic. Chen et al. (2022) apply mixed-integer programming models to redesign routes and schedules for the campus bus system during the pandemic. Frazier et al. (2022) study Covid-19 related interventions in university populations. Kacapyr (2021) consider the problem of classroom design with social-distancing considerations.

Specifically with respect to scheduling, there have been a number of different approaches taken by various schools to implement social distancing. These range from fully online to fully in person classes (UNESCO, 2020). Barnhart et al. (2022) use integer optimization for timetabling and term planning during the pandemic. They expand their university's two-term schedule by adding a third term and redistribute the courses in these three terms so as to reduce in-person student population on campus. They also decide timetables for these courses to implement required social distancing protocols while also allowing as many students as possible to attend their classes in person. Navabi-Shirazi et al. (2022) study the problem of deciding course modes for each course. These could be fully remote (online), fully in person, or a hybrid of the two. They also reassign classrooms under the capacities, newly-reduced to implement social distancing, to the non-remote courses.

The Three Cohort Model (3CM) of Gore et al. (2022) is closest to part of our work in that 3CM considers a similar problem as our student-level scheduling problem. However, our contribution differs from their work in the following ways:

- The excess (GE) of 3CM is the sum of "course excesses", where each "course excess" is defined to be the maximum of the excess for that course over all cohorts (groups). In contrast, our TE metric is the sum of all excesses over courses and groups. TE is more operationally relevant since each instance of excess is an instance where a student is invited to campus, but was asked to attend a teaching session online, and thus creates disutility for that student. Minimizing the total number of such instances would better serve the students, since it more directly corresponds to maximizing an aggregate measure of welfare.

Ensuring that SE does not exceed the capacity of the excess room is also directly operationally relevant. Otherwise, the excess students would not have a Covid-safe place to attend their classes online while present on campus.

Finally, a low TD is desirable to create a more equal learning and teaching experience for students and teachers.

The 3CM objective is to minimize the GE, whereas our objective is to minimize a linear combination of the TE, the TD, and to ensure that the SE does not exceed a specified constant — namely the capacity of the excess room.

- In their setting, the number of groups is held fixed at 3, whereas we take this as a parameter, and discuss the trade-offs in deciding this value. Based on these trade-offs, we allow for changing this value from term-to-term. In practice, we divided the students in 3 groups in Fall 2020 and in Spring 2021, and 2 groups in Summer 2021.
- We provide theoretical insight by studying the relaxed linear program and give conditions that ensure that the optimal objective value of the relaxed program is 0. In other words, we identify necessary conditions so that zero excess can be achieved.
- Gore et al. (2022) state that their model "may be well suited for situations in which all courses meet on the same set of days", but we do not believe this to be a requirement to apply this model. In fact, CBS course schedule does not satisfy this requirement, and yet we were able to apply a very similar model while being fair to the students. The concern cited by Gore et al. (2022) is that "students would not be guaranteed an in-person course meeting each week" with this model. However, over the term, each student can still be guaranteed to attend each of their classes in person an equal number of times regardless of their group. Based on the counterfactual performance of the 3CM, Gore et al. (2022) say that the model "is inappropriate if we desire to maximize the number of students attending some in-person classes." In contrast, we are able to guarantee that each student enrolled in hybrid classes and willing to attend them in person, is scheduled to attend each of their classes in person at least once every M class meetings throughout the term.
- Gore et al. (2022) ultimately abandon the 3CM approach for three reasons:
    1. They could not guarantee that each student attends each of their classes each week. This was true for us as well. However, we were still able to guarantee that over the term, each class was attended by each student an equal number of times.
    2. Their approach did not accommodate all classes, such as those that meet once a week. This was not an issue for us. In fact, in Fall 2020, almost 75% of classes scheduled to be hybrid met once a week.
    3. They found that the optimal solution produced by the 3CM exceeded the designated social distancing capacities of the classrooms by up to 40%. We eliminate this potential

issue by recommending course enrollment capacities and number of groups based on our experiments on past data.

The solution finally implemented at Clemson University is markedly different from the 3CM approach or our student-level scheduling approach. It consists of assigning each student to an in-person attendance group individually for each course. This allows for schedules in which some students would need to attend one class in person and others online on the same day of the week for the entire term, and so represents a set of trade-offs that we do not consider here.

As in many of the papers cited above, we apply mixed-integer programming techniques. Mixed-integer programming has been extensively studied. Conforti et al. (2014) and Nemhauser and Wolsey (1988) are two excellent books that provide a comprehensive overview of mixed-integer programming and its applications to optimization. It has also been applied to scheduling problems in a wide variety of domains. Beaumont (1997) applies mixed-integer programming to the problem of scheduling a workforce to meet varying demand, Saraç and Tutumlu (2022) apply it to the parallel machine-scheduling problem, and Yadav and Tanksale (2022) apply it to scheduling problems in healthcare.

2. Student-Level Scheduling

We first consider the student-level scheduling problem. In this setting, different students can enroll in different classes and hence they are scheduled individually.

2.1. Formulation

We have a set of students, a set of classes, as well as the enrollment information, calendar schedules, and social distancing capacities of these classes. With this data, we consider the following scheduling method. We assign each student to exactly one of M groups where M is some positive integer. This is what we call an integral assignment. A fractional student assignment is one where each student is fractionally assigned to multiple groups and the mass of this student in these groups adds up to 1. A relevant fractional student assignment is the uniform one, which we now define.

Definition 1 (Uniform fractional student assignment). *A uniform fractional student assignment is a fractional student assignment where each student is assigned to each of the M groups with the student's mass in each group being $1/M$.*

The uniform fractional student assignment is ideal in that it is perfectly balanced and symmetric across both classes and students, and hence is a useful benchmark. However, being fractional, it cannot be implemented.

For each day of the term, one group is assigned to attend classes in person on that day, and the particular group will vary day-to-day. Ideally, each student will attend each of their classes in person an equal number of times. Because each class in the school follows a schedule that repeats weekly, this can be achieved only if each student attends class in person each day of the week a roughly equal number of times throughout the term. This is straightforward if M is co-prime with the number of days classes are scheduled. Otherwise, we need to vary the repeating sequence appropriately to accomplish this. Figure 1 shows the group assignment calendar for the first half

Figure 1: Group assignment calendar for the first half of Fall 2020. Here, for each teaching day (Monday–Thursday), one of the 3 groups (C, B, or S) is assigned to attend classes in person. Note that September 18th was included as a teaching day (running a Monday schedule), in order to make up for the Labor Day holiday (September 7th).

of Fall 2020 group calendar,with 3 groups (named C, B, and S) as it was implemented at CBS. Note that classes take place on 4 days of the week (Monday–Thursday) and the assignment is rotated through the 3 groups — C, followed by B, followed by S. Then, for any day of the week, the group assigned for that day also rotates through the same sequence (e.g., group C is assigned for the first Monday, followed by group B for the second Monday, etc.). This ensures that each day of the week sees an equal number of each of the three groups scheduled throughout the term. In Summer 2020, there were 2 groups scheduled over the same 4 teaching days. Since 2 and 4 are not co-primes, we need a different approach to group scheduling. In particular, the 2 groups (named A and B) are scheduled in an ABAB sequence in one week and a BABA sequence the next, alternating throughout the term.

As can be seen above, on any day of the week, any one of the M groups may be assigned an in-person schedule. Hence, when considering in-person attendance for any class, we must assume that any one of the M groups may be assigned in-person attendance for that class. We also note that once we have an integral assignment, the protocol described above produces an inherently fair schedule that ensures that every student attends classes in person for a fraction $1/M$ of the days. Our problem then reduces to that of computing a group assignment. We now discuss ways to evaluate a given assignment by computing some quantities of interest. Please see Appendix A.1 for a formal treatment.

- **Surplus Simultaneous Excess** (SSE). If a class has more assignments than the social distancing guidelines allow, we assign the excess students chosen randomly to an "excess room." SE is the maximum number of students assigned to the excess room at any given time. The excess room has a finite capacity, and we define the amount of SE exceeding this capacity as the SSE.
- **Total Excess** (TE). Every time a student is scheduled to attend classes in person, but is assigned to an excess room, it creates disutility for that student. TE measures this disutility by summing up the excess across all classes assuming each of the groups is scheduled to in-person attendance.
- **Total Deviation** (TD). It is undesirable to have too many in-person students on one day and too few on another. To avoid such outcomes, we also need to minimize TD, which is the sum of the absolute difference between an assignment and the uniform fractional student assignment that divides each student's mass equally among all groups (see definition 1).

Figure 2 shows an example assignment of 13 students, each enrolled in a subset of 3 specific classes divided into two groups: group 1 (green) and group 2 (white). In this example, we see an excess of 1 each from classes 2 and 3 when group 1 is scheduled, and an excess of 1 from class 2 when group 2 is scheduled. The TE is thus 3. We note that classes 2 and 3 overlap on Thursdays between 13:00 and 13:30, and if group 1 is scheduled to attend classes in person, both these classes would have an excess of 1 each. The SE is thus 2. Since the excess-room capacity is 2, the SSE is 0. The contribution to the TD from class 1 is 0, from class 2 is 0, and from class 3 is 2, making the TD equal to 2.

Clearly, a desirable assignment would have a low SSE, a low TE, and a low TD. To achieve such an assignment, we formulate an optimization problem that tries to minimize a linear combination of these three quantities while being constrained so that each student is assigned to exactly one group. In other words, our optimization problem is as follows:

$$\begin{aligned} \text{minimize} \quad & \text{A weighted sum of SSE, TE, and TD} \\ \text{subject to} \quad & \text{Each student being assigned to exactly one group.} \end{aligned}$$

Please see Appendix A.1 for a formal integer programming formulation with auxiliary variables.

We note that, for us, the SSE is the most important metric since, if it is positive, that would indicate our inability to house excess students in a Covid-safe environment. TE is the next most important metric since it directly corresponds to social welfare. By comparison, TD is the least

Class 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Monday, Wednesday from 10:00–11:30
Social distancing capacity = 4

Class 2 | 1 | 6 | 7 | 9 | 10 | 13 |

Wednesday, Thursday from 12:00–13:30
Social distancing capacity = 2

Class 3 | 2 | 4 | 5 | 8 | 11 | 12 |

Tuesday, Thursday from 13:00–16:00
Social distancing capacity = 3

Excess-room capacity = 2.

a Class enrollments, timetable, capacities and group assignments.

Class	Group 1	Group 2	Total
Class 1	0	0	0
Class 2	1	1	2
Class 3	1	0	1
Total	2	1	TE=3

b TE computation. Each cell contains the excess from the class on the left when the group in the top row is scheduled to attend classes in person. Rightmost column contains contribution to TE from each class. Bottom row contains contribution to TE from each group. The bottom right cell contains the TE.

Class	Group 1	Group 2	Total
Class 1	0	0	0
Class 2	0	0	0
Class 3	1	1	2
Total	1	1	TD=2

c TD computation. Each cell contains the contribution to TD from the class on the left when the group in the top row is scheduled to attend classes in person. Rightmost column contains contribution to TD from each class. Bottom row contains contribution to TD from each group. The bottom right cell contains the TD.

Day and	Group 1				Group 2			
Time	Class 1	Class 2	Class 3	Total	Class 1	Class 2	Class 3	Total
Monday All day	0	0	0	0	0	0	0	0
Tuesday All day	0	0	0	0	0	0	0	0
Wednesday 10:00–11:30	0	0	0	0	0	0	0	0
Wednesday 12:00–13:30	0	1	0	1	0	1	0	1
Thursday 12:00–13:00	0	1	0	1	0	1	0	1
Thursday 13:00–13:30	0	1	1	2	0	1	0	1
Thursday 13:30–16:00	0	0	1	1	0	0	0	0
Maximum	0	1	1	2	0	1	0	1

d SE computation. Excess for each class is computed for each time instant assuming each group to be attending classes in person. Group totals are then computed for each time instant, followed by a maximum over these in the bottom row. Maximum of this row is the SE. In this example, SE = 2. Since excess room has capacity 2, we have SSE = 0.

Figure 2: An example showing 13 students enrolled in 3 classes along with the weekly schedules and social distancing capacities of these classes. The colors depict one possible assignment of the students to groups: the students in group 1 are indicated in green and the students in group 2 are indicated in white. In this example, we have a TE of 3, an SE of 2, SSE of 0, and a TD of 2.

important metric. The objective of our integer program is to minimize a weighted sum of these three metrics. This weighted sum is our objective function. In practice, however, we observe that SE takes very small values relative to the excess-room capacity even when we do not include it in the objective function. Therefore, we set the weight of SSE to 0 in the objective function, and compute it after the resulting optimal solution is obtained. We weight the rest of the metrics as follows:

- The weight of the TE metric is normalized to be 1.
- The weight of the TD metric needs to be set to be between 0 and 1. In practice, we set it to[1] 0.25.

Finally, the choice of M is important as well. M should be kept small enough so that students can hope to attend as many classes in person as possible. However, it should not be so small that the excess is large.

2.2. Analysis

Let us consider a simplified version of the problem: one without the integer constraints (called the relaxed linear program). Essentially, this allows for each student to be fractionally assigned to multiple groups. One such assignment would be the uniform fractional student assignment. In fact, it turns out that this is the optimal fractional student assignment for the relaxed linear program. We define the TE in the uniform fractional student assignment as the uniform excess (UE). See Appendix A.1 for a formal definition. This gives us insights into the original problem as the optimal objective value of the integer program must be lower bounded by the optimal objective value of the relaxed linear program. We can prove the following theorem.

Theorem 1.

1. *If TE is the only metric with non-zero weight in the objective function, and integer constraints are relaxed, the optimal objective value is equal to the UE. Therefore, the TE of any feasible allocation for the integer program is lower bounded by the UE.*
2. *The optimal objective value of the relaxed linear program is attained by the uniform fractional student assignment.*

The proof follows from applying strong duality to the relaxed linear program. See Appendix A.2 for the detailed proof.

Example 1 below shows that even when we restrict the objective function so that TE is the only metric with non-zero weight, the optimal objective value may not always be equal to the UE.

Example 1. *Suppose that we have* 3 *students, each enrolled in 2 of* 3 *classes, and class has a social*

[1] In practice, the optimization program managed to achieve the minimum possible TE, and so the weight proved not to be so important.

distancing capacity of 1*. The enrollments are such that for every pair of students, there is a unique class that pair is enrolled in. Suppose that we set* M *to* 2*. Clearly, we must have at least one group with at least two students. Then, we will have at least one class with an excess of* 1*. Thus the* TE *is at least* 1*, whereas the* UE *is* 0 *since assigning half of each student's mass to each of the two groups results in an excess of* 0.

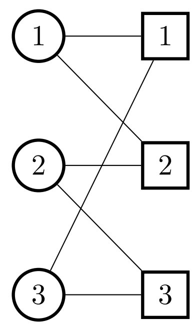

Figure 3: Graph showing student enrollments for Example 1.

We observe that the integer constraints are binding in this case, since otherwise both the TE and the UE would be equal to 0. Note that in the same example, if we increase the number of groups to 3, we can assign each student to a different group, achieving a TE of 0. This suggests that the enrollment of students to classes and the choice of M both play a role in making the integer constraint binding.

We can also come up with a lower bound on TD. If we consider each course in isolation, the problem of minimizing its contribution to TD is a fair division problem — the problem of dividing its students into M groups equally as much as possible. This problem can be solved by assigning students one-by-one to a group in a round-robin fashion. This way, we can compute the minimum possible contribution to the TD from each course and aggregate it to get the minimal deviation (MD). We can prove the following theorem.

Theorem 2. *The TD is lower bounded by the MD.*

See Appendix A.3 for the formal definition of MD, and the proof of Theorem 2.

Getting a TE of 0: From the above discussion, we conclude that to get a TE of 0, we must first have a UE of 0, but that alone does not guarantee a TE of 0. To get a UE of 0, it suffices to set a registration limit on each class so that dividing each student's mass equally among M groups gives an excess of 0 from that course. We call this the "minimal-capacity" condition. See Appendix A.1 for a formal definition. In the next section, we look at past data, and observe that this may actually lead to a TE of 0 in real data even though that is not theoretically guaranteed.

2.3. Empirical Results

We tested our methodology on Fall 2019 data before using it in production in the Fall 2020 term. The data (provided to us by the CBS Dean's Office) consisted of 739 students, each enrolled in 5–6 classes for a total of 4,204 enrollments. Each class had an average of ≈ 60 students enrolled in it. The data was high quality, requiring little to no cleaning. When conducting the experiments, we assumed all 117 classes to be in a hybrid mode. In contrast, in Fall 2020 and Spring 2021, there were over a thousand students enrolled in 115 and 113 classes, respectively. Not all of these classes were held in a hybrid mode, however: about half of the classes were held fully online, while the rest were held in a hybrid mode. Based on our Fall 2020 data indicating a low turnout of students to in-person classes, we conducted a survey prior to the beginning of classes in Spring 2021 to find out if any students wanted to opt out of in-person attendance. Of the 1,553 students registered, 1,035 opted to attend classes in person. In Summer 2021, we had only 336 students enrolled in 8 classes, of which 226 opted to attend classes in person, and only those were considered for hybrid scheduling. Table 1 summarizes these statistics.

Term	Students	Total Classes	Hybrid Classes	Total Enrollments
Fall 2019	739	117	117	4204
Fall 2020	1302	115	67	5815
Spring 2021	1035 (of 1553)	133	61	7544
Summer 2021	226 (of 336)	8	8	565

Table 1: Student enrollment statistics. Our method was tested on pre-Covid Fall 2019 data. For this test, all classes were assumed to be in a hybrid mode of attendance.

We implemented the optimization program using the gurobipy (Gurobi Optimization, LLC, 2022) library version 9.5.1 in the Python programming language version 3.9.7. To run the programs, we used AMD EPYC 7532 3.2–3.3 GHz processors. Each program used up to 30 threads and was allocated up to 100 GB RAM. An optimal solution was found in almost all cases within a tolerance of 10^{-4} in the objective value. Only in one case, the program ran out of memory before reaching the optimal solution. The run time of each program ranged from 2 hours to 4 days on the Fall 2019 data set, but was less than 5 minutes on Fall 2020, Spring 2021, and Summer 2021 data sets.

While testing on the Fall 2019 data, we considered 3 potential social-distancing protocols (the final protocol had not been decided at the time of these experiments), namely, 1)"6ft": 6ft of distance between every two students, 2) "4ft": 4 ft of distance between every two students, and 3) "50sf": 50 sq ft of area given to each student. In general, 6ft capacity was less than 50sf capacity, which in turn was approximately equal to the 4ft capacity. These experiments are summarized in Table 2. The MIP Gap column reports the maximum difference between the objective value of the solution obtained and the optimal objective value in that setting. Note that in all except one case, the optimizer reached an optimal solution (MIP Gap = 0). In the case when we used the 50sf social distancing protocol and divided the students into 6 groups, the optimizer ran out of memory before reaching an optimal solution.

Social-Distancing Protocol	M	Total Excess	Uniform Excess	Simultaneous Excess	Total Deviation	Minimal Deviation	Objective Value	MIP Gap
6ft	6	1	1	1	252.00	252.00	64.00	0
6ft	5	25	25	2	180.80	180.80	70.20	0
6ft	4	263	263	10	158.00	158.00	302.50	0
6ft	3	753	753	33	100.00	100.00	778.00	0
6ft	2	1595	1595	79	60.00	60.00	1610.00	0
4ft	6	0	0	0	252.00	252.00	63.00	0
4ft	5	0	0	0	180.80	180.80	45.20	0
4ft	4	3	3	1	158.00	158.00	42.50	0
4ft	3	58	58	5	100.00	100.00	83.00	0
4ft	2	601	601	41	60.00	60.00	616.00	0
50sf	6	0	0	0	253.00	252.00	63.25	0.25
50sf	5	0	0	0	180.80	180.80	45.20	0
50sf	4	0	0	0	158.00	158.00	39.50	0
50sf	3	1	1	1	100.00	100.00	26.00	0
50sf	2	417	417	28	60.00	60.00	432.00	0

Table 2: Summary of experimental results of hybrid scheduling with different values of M, and different social distancing policies. All assignments were computed with setting the weight of the TD metric to 0.25, and the weight of the surplus SE to 0.0 in the objective function. The capacity of the excess room was determined to be 140, and hence, the surplus SE was 0 in all cases. An optimal solution was reached in all except for the one with 50sf social distancing protocol and 6 groups. The MIP Gap indicates the maximum difference between the objective value of the obtained solution and the optimal objective value.

As can be expected, the TE, UE and SE decrease with increasing number of groups. We observe that keeping the number of groups constant, these excesses decrease when going from 6ft to 4ft, and from 4ft to 50sf social-distancing protocols. Figure 4 shows this trend for TE.

Two more interesting observations can be made from the results. The first observation is that the TD in the solution obtained is equal to the MD in almost all the experiments. The only case where we did not obtain a solution with TD equal to MD is also the only case when the optimizer ran out of memory before reaching the optimal solution.

The second, and more operationally relevant, observation is that the UE is equal to the TE of the assignment computed by our optimizer in all of our experiments on the past data, meaning that the relaxation of the integer constraint is tight as far as minimizing the TE is concerned. We could hope, then, that the optimal TE would be 0 if the UE were set to 0.

To ensure a UE of 0, we need to ensure that the minimal-capacity condition is satisfied for each class. That is, the enrollment limit of each class is upper bounded so that the social distancing capacity is sufficient to allow a fraction $1/M$ of the enrolled students to attend that class in person.

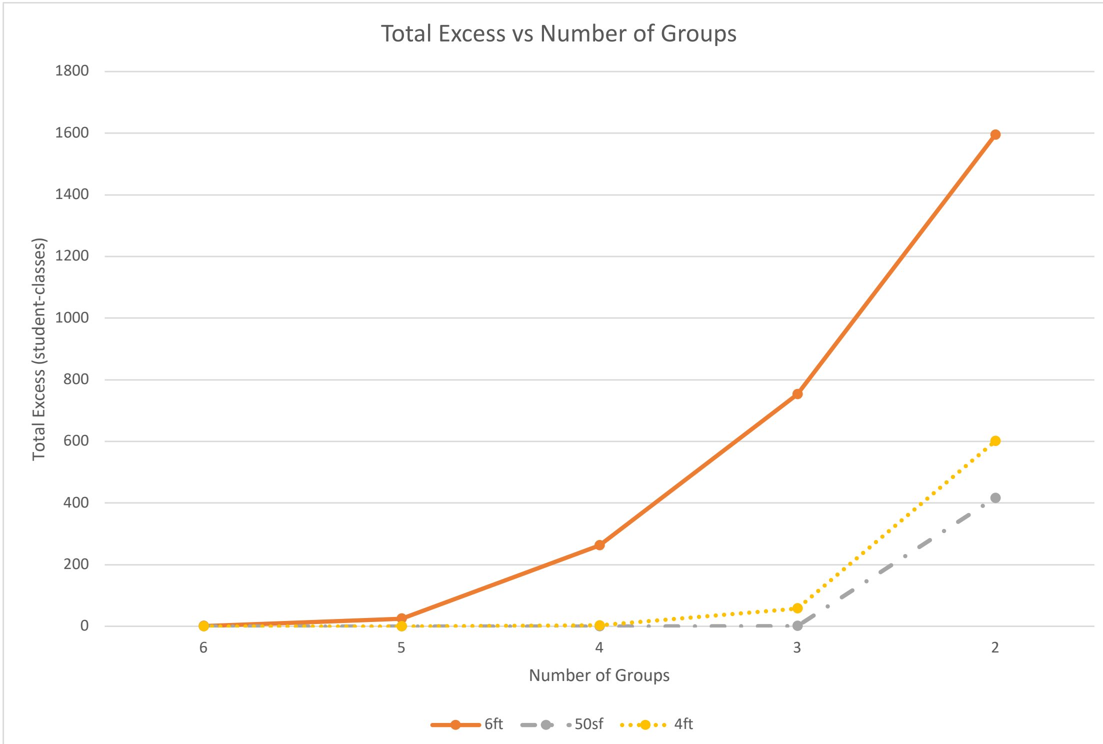

Figure 4: Effect of reduction in the number of groups on total excess for different social-distancing protocols. The TE increases with a decreasing number of groups. For the same number of groups, 6ft social-distancing protocol has the largest TE, followed by the 4ft social-distancing protocol, followed by the 50sf social-distancing protocol. In the final implementation, the classroom capacities were somewhere between 6ft and 50sf/4ft protocols, and the number of groups was set to 3.

In this scenario, a large M allows more students to enroll in classes of their choosing, and a small M gives more in-person attendance opportunities to students.

When the social distancing protocol and the corresponding social distancing capacities were finalized (these turned out to be between the 6ft and 50sf/4ft capacities described above), we observed that these were 25%–30% of the original capacities of almost all of the classrooms. This meant that if M was set to 4, enrollment capacities would have increased or remained the same as previous years. Furthermore, based on the finalized classroom capacities, we expected a low impact on TE when going from 4 groups to 3 groups. A policy decision was made to set M to 3. This caused a reduction in enrollment capacities while allowing students to attend classes in person once every 3 days.

Another important decision was the setting of the weights of the different metrics in the objective function. We normalized the weight of the TE metric to 1. As discussed earlier, the SSE metric is important only if the SE values in different settings, computed on an assignment that was optimal without the SE metric, are comparable to the social distancing capacity of our excess room. The social distancing capacity of the designated excess room was determined to be 140 seats. As can be seen in Table 2, all of the settings we tried on the past data resulted in an optimal assignment (computed without the SE metric in the objective function) having a simultaneous excess much less than the excess room capacity. Therefore, we concluded that the weight of the SSE did not matter in our case, and we set it to 0. The weight of the TD metric was set to 0.25, though we observe that the exact value did not matter in the resulting problem since we always managed to achieve a TE equal to the UE.

Table 3 shows the final results of hybrid scheduling as implemented in the Fall 2020, Spring 2021, and Summer 2021 terms at Columbia Business School for second-year MBA students. Students enrolled only in fully online classes were also assigned to a group since groups were used for non-instructional gatherings among students as well (for example, if a club activity was scheduled on a day assigned to group 1, only group 1 students could attend it in person). It can be seen that in all years, the TE was always equal to the UE. While care was maintained in making sure that the minimal-capacity condition was satisfied by controlling registration for most courses, this constraint was slackened for some, which led to the positive UE and TE in the Fall 2020 and Summer 2021 terms.

In Fall 2020, we observed that many students decided to stay at home even when given an option to attend classes in person. In light of this, a survey was conducted before the start of the Spring 2021 term, where students indicated their preferred mode of attendance. Only those who were interested in attending classes in person were considered for scheduling. While we hoped to be able to set M to 2 for scheduling in Spring 2021, it turned out to be infeasible since many students opted to attend classes in person. In Summer 2021, however, the number of students who opted to attend classes in person was low enough to allow us to set M to 2, while still maintaining higher enrollment limits.

Term	M	Total Excess	Simultaneous Excess	Uniform Excess	Total Deviation
Fall 2020	3	20	5	20	382.67
Spring 2021	3	0	0	0	719.33
Summer 2021	2	12	4	12	6

Table 3: Summary of the final results of hybrid scheduling in the Fall 2020, Spring 2021, and Summer 2021 terms.

2.4. Deployment

The deployment took place as per the following steps:

1. As noted earlier, not all classes were held in a hybrid mode. The first step was to determine which classes were to be held fully online and which in a hybrid mode. This was determined based on course content and instructor preference.

2. In the next step, the social distancing capacities were determined for each classroom. The meeting times, and classroom assignments for each class were also finalized.

3. A tentative number of groups, M, was determined. Along with classroom capacities, this decided the enrollment limit for each class as per the minimal-capacity constraint.
4. Students were allowed to enroll in classes. There enrollments were not in our control except for deciding the enrollment limits.
5. (only in Spring 2021 and Summer 2021 terms) Students indicated whether they want to opt out of in person classes altogether.
6. We computed group assignments for students who did not opt out, and were enrolled in at least one hybrid class, based on actual enrollments with M groups. We also checked if the number of groups could be lowered without increasing TE and SE to unmanageable levels. We did this by recomputing the assignments with lower values of M.
7. Based on the results, we decided the final value of M.
8. We randomly assigned a group to the students who were not assigned one so far.
9. The Dean's Office disseminated the group assignments for each student along with the group calendar for the term.

In all three terms, the tentative number of groups in Step 3 was determined to be 3. In Fall 2020 and Spring 2021 terms, this was also the final number of groups. As discussed in the previous section, based on low student turnout in Fall 2020, students were given an option to opt out of in-person classes in Spring 2021 and Summer 2021 terms. In both terms, we tried group assignments with 2 groups in Step 6. In Spring 2021 this resulted in a high value of TE and the number of groups was finalized to be 3. In Summer 2021, group assignment with 2 groups produced a manageable level of TE and SE, and the number of groups was finalized to be 2. As a result of this, we were able to accommodate more willing students to attend in-person classes without reducing class enrollment limits. Because each computation only took less than 5 minutes to run, such recomputations were not a challenge.

To increase efficiency further, students were given another chance to opt out of in-person classes on any day of their choosing through an app. Other students were then chosen to attend those classes in person on that day. This was handled by the Dean's Office.

3. Team-Level Scheduling

MBA programs prepare students for the world of business by teaching them collaboration and networking. To aid collaboration, first-year MBA students at Columbia are divided into learning teams (LTs), which are assigned group tasks and attend classes together. Networking is usually not an issue since in-person class attendance automatically encourages that. With a hybrid model, however, we need to ensure that every student has a chance to meet every other student in person as well. We call this **mixing** among students. Mixing is particularly important for first-year students, and we need to encourage it along with considerations such as fairness. We formally define these considerations in the next section.

3.1. Problem Formulation

Since all first-year students attend the same set of classes, we can treat each team as an individual entity, drastically reducing the number of variables. In our case, we had 12 teams, 4 of which were to be scheduled to attend classes in person each day for a 44-day term. We observe that there are 495 distinct subsets of size 4 that can be drawn from a set of 12 teams. We call each such subset an LT group. Exactly one LT group should attend classes in person on each day of the term, and we can again use integral variables to ensure this. In this case, a fractional group assignment is one where on any given day, multiple LT groups are attending classes in person fractionally, while their total mass adds up to 1. A relevant fractional group assignment is the uniform one, which we now define.

Definition 2 (Uniform Fractional Group Assignment). *A uniform fractional group assignment is a fractional group assignment in which on each day of the term, a fraction* $1/495$ *of each LT group is attending classes in person.*

We have two notions of fairness. See Appendix B.1 for a formal treatment.

- Longitudinal balance. Days of in-person attendance should be equally distributed throughout the term. Since we have 12 teams attending classes in person in groups of 4, each team should attend classes in person exactly once every 3 days.
- Balance over days of the week. Since classes follow a schedule that repeats weekly, each team should attend classes in person on each day of the week an equal number of times.

As mentioned earlier, our objective is to encourage mixing among students. We can do this by maximizing the minimum number of times any two teams attend classes in person together. Putting it all together, we can write the optimization problem as

$$\begin{aligned} \text{maximize} \quad & \text{Minimum number of in-person meetings between any two teams} \\ \text{subject to} \quad & \text{Exactly four LTs attending classes in person on each day} \\ & \text{Longitudinal balance being maintained} \\ & \text{Balance over days of the week being maintained.} \end{aligned}$$

See Appendix B.1 for a formal treatment.

3.2. Insights

Again, we can draw useful insights into the problem by solving the relaxed linear program. In effect, this would allow fractional group assignments. We are able to prove the following theorem.

Theorem 3.

1. *The team-level scheduling problem with integer constraints relaxed has an optimum when LT groups are assigned a uniform fractional group assignment.*
2. *The optimal objective value when integer constraints are relaxed is equal to* 4.
3. *In any solution where the objective value is* 4*, each team meets with every other team in person exactly* 4 *times.*

We defer the proof to Appendix B.2. We are now able to observe that we must have an optimality gap when solving this integer program. Suppose to the contrary that there is a solution where every pair of teams meet at least 4 times. For any given team, we know that it meets with exactly 3 other teams every day. Then, to meet each of the 11 other teams 4 times, the team needs to attend classes in person at least 15 days (rounded up from 14.67). Thus, we would need each of the 12 teams to attend classes in person for at least 15 days of the term. Since every day 4 teams attend classes in person, this would require at least 45 days in the term, while we have only 44 days. Thus, an objective value of 4 is impossible and we are left with an optimality gap. In fact, as we discuss in the next section, the optimizer was able to find a solution with an objective value of 3 as the optimum.

3.3. Results

The only data required for this problem was the number of days of the term, the number of learning teams, and the number of teams which are to be scheduled to attend classes in person on each day. We formulated the optimization program using the gurobipy (Gurobi Optimization, LLC, 2022) library version 9.5.1 in Python version 3.9.7, and used AMD EPYC 7532 3.2 GHz processor to compute a solution. The optimizer was allocated up to 30 threads and up to 100 GB RAM. The optimal schedule thus computed had an objective value of 3. That is, our schedule ensured that each team met with every other team at least three times throughout the term. Table 4 shows the number of times each team met with every other team. The diagonal entries show the number of times each team attended classes in person throughout the term.

We observe that, while every pair of teams meet at least three times, the actual number of meetings varies quite a bit from pair to pair, ranging from 3 to 7. Also, we observe that the problem as it is written above can have multitudes of distinct solutions that all have the same objective values. We now discuss some alternate objective functions and how their choice affects the imbalance noted above. See Appendix B.3 for a formal treatment.

- F1: Maximize the minimum number of times any pair of teams meet. This is the objective function described above, and used by us in real life. While it does encourage different teams meeting multiple times, it may lead to an imbalance due to some pairs meeting many more times than others.

Learning Team	1	2	3	4	5	6	7	8	9	10	11	12
1	15	4	4	5	5	3	7	3	4	3	4	3
2	4	15	3	3	5	6	6	4	5	3	3	3
3	4	3	15	5	5	4	3	5	3	4	4	5
4	5	3	5	14	4	4	4	3	4	3	3	4
5	5	5	5	4	15	3	4	5	4	3	4	3
6	3	6	4	4	3	15	4	4	5	5	3	4
7	7	6	3	4	4	4	15	4	3	3	3	4
8	3	4	5	3	5	4	4	15	4	4	4	5
9	4	5	3	4	4	5	3	4	15	6	4	3
10	3	3	4	3	3	5	3	4	6	14	5	3
11	4	3	4	3	4	3	3	4	4	5	14	5
12	3	3	5	4	3	4	4	5	3	3	5	14

Table 4: Meeting matrix between teams according to the optimal schedule. Each element of the matrix shows the number of times the learning team numbered in the top row meets with the team numbered in the leftmost column in person.

F2: Minimize the maximum number of times any pair of teams meet as a secondary objective. This would mitigate the imbalance between different pairs of teams in terms of their number of meetings.

F3: Minimize the maximum absolute difference between the number of times any pair meets and 4. This again reduces the imbalance between different pairs of teams by forcing their meetings to be as close to 4 as possible. Here, 4 is picked because of Theorem 3 Part 3.

Figure 5 shows a comparison of these objective functions. We see that both of the alternate objective functions F2 and F3 mitigate the imbalance by restricting the number of pairwise meetings to a number between 3 and 5. While we see that the two lead to different solutions, we also observe that they need not do so by the nature of the respective functions alone. Both solutions have the same objective values when evaluated with either of the objective functions. The difference, then, seems to be due to some quirk of the optimizer itself. This again points to the fact that there are many degrees of freedom in this problem that we have not explored so far.

Another possible objective function, we designate as F4 is one that minimizes the sum of the absolute difference between the number of times each pair of teams meet and 4 (instead of minimizing the maximum absolute difference as in F3). This would then penalize each pair of teams for meeting more or less than 4 times. The optimizer when trying to solve the problem with objective function F4 ran out of memory before converging to a solution.

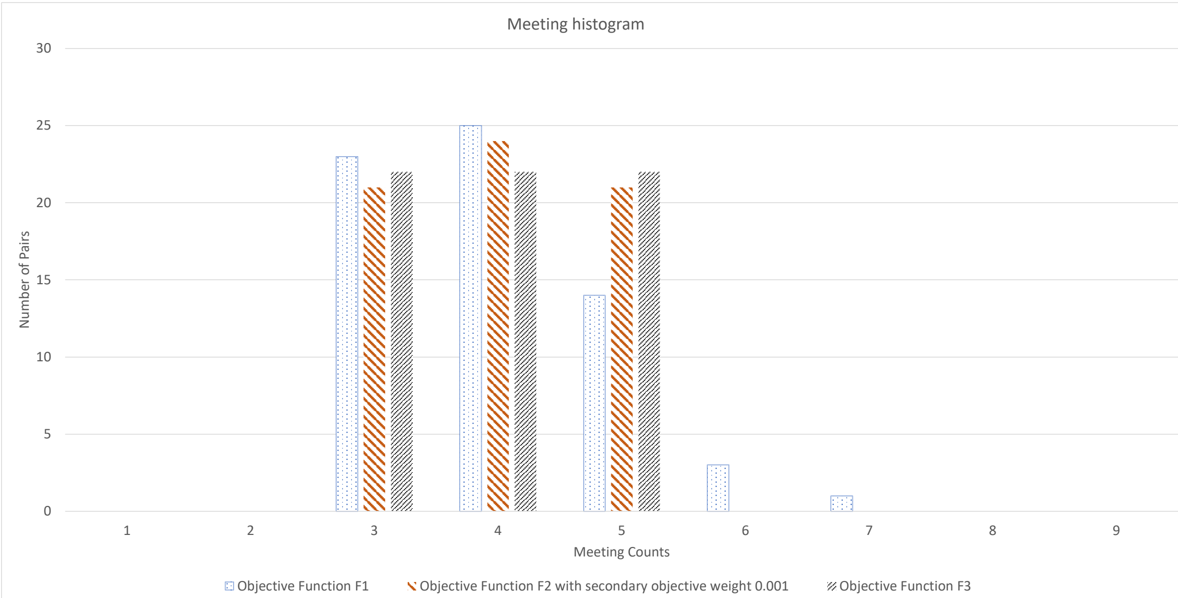

Figure 5: Histogram depicting number of meeting times between any pair of distinct learning teams when the schedule is evaluated using different objective functions.

4. Conclusion

We were able to use mixed-integer programming to formulate and solve the hybrid scheduling problem. We considered two settings of the problem: student-level and team-level scheduling. In the case of student-level scheduling, we were able to use insights obtained from our analysis of the relaxed linear program to make policy decisions that helped us get to a more efficient solution, i.e., one with low excess. The solution we computed was used to schedule over 2,500 second-year MBA students enrolled in 136 hybrid classes across three terms at Columbia Business School.

In the case of team-level scheduling, we were able to use the greater control such scheduling affords due to the homogeneity of the data in order to obtain an even more efficient schedule. Specifically, we were able to guarantee that each learning team met with every other learning team at least three times during the course of the 44-day term. Our model was used to schedule 790 first-year MBA students to their classes across two terms.

References

Salwani Abdullah and Hamza Turabieh. On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Inf Sci*, 191:146–168, may 2012. ISSN 00200255. doi: 10.1016/j.ins.2011.12.018.

Roberto Asín Achá and Robert Nieuwenhuis. Curriculum-based course timetabling with SAT and MaxSAT. *Ann Oper Res*, 218(1):71–91, 2014. ISSN 15729338. doi: 10.1007/s10479-012-1081-x.

Hamed Babaei, Jaber Karimpour, and Amin Hadidi. A survey of approaches for university course timetabling problem. *Computers and Industrial Engineering*, 86:43–59, 2015. ISSN 03608352. doi: 10.1016/J.CIE.2014.11.010.

Cynthia Barnhart, Dimitris Bertsimas, Arthur Delarue, and Julia Yan. Course Scheduling Under Sudden Scarcity: Applications to Pandemic Planning. *Manufacturing and Service Operations Management*, 24(2): 727–745, nov 2022. ISSN 15265498. doi: 10.1287/msom.2021.0996. URL `https://pubsonline.informs.org/doi/abs/10.1287/msom.2021.0996`.

Nicholas Beaumont. Scheduling staff using mixed integer programming. *European Journal of Operational Research*, 98(3):473–484, 1997. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(97)00055-6. URL `https://www.sciencedirect.com/science/article/pii/S0377221797000556`.

Ruggero Bellio, Sara Ceschia, Luca Di Gaspero, Andrea Schaerf, and Tommaso Urli. Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers and Operations Research*, 65:83–92, jul 2016. ISSN 03050548. doi: 10.1016/j.cor.2015.07.002.

Andrea Bettinelli, Valentina Cacchiani, Roberto Roberti, and Paolo Toth. An overview of curriculum-based course timetabling. *TOP*, 23(2):313–349, jul 2015. ISSN 18638279. doi: 10.1007/S11750-015-0366-Z/TABLES/3. URL `https://link.springer.com/article/10.1007/s11750-015-0366-z`.

Eric Budish, Gérard P. Cachon, Judd B. Kessler, and Abraham Othman. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2):314–336, mar 2017. ISSN 15265463. doi: 10.1287/opre.2016.1544.

Edmund K. Burke and Yuri Bykov. An adaptive flex-deluge approach to university exam timetabling. *INFORMS Journal on Computing*, 28(4):781–794, sep 2016. ISSN 15265528. doi: 10.1287/IJOC.2015.0680.

Michael W. Carter and Craig A. Tovey. When Is the Classroom Assignment Problem Hard? *Operations Research*, 40(1-supplement-1):S28–S39, feb 1992. ISSN 0030-364X. doi: 10.1287/opre.40.1.s28. URL `https://www.jstor.org/stable/3840833`.

Gongyu Chen, Xinyu Fei, Huiwen Jia, Xian Yu, and Siqian Shen. The University of Michigan Implements a Hub-and-Spoke Design to Accommodate Social Distancing in the Campus Bus System Under COVID-19 Restrictions. *INFORMS Journal on Applied Analytics*, nov 2022. ISSN 2644-0865. doi: 10.1287/INTE.2022.1131.

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*, volume 271 of *Graduate texts in mathematics*. Springer, 2014. ISBN 9783319110073.

John J. Dinkel, John Mote, and M. A. Venkataramanan. Efficient decision support system for academic course scheduling. *Operations Research*, 37(6):853–864, 1989. ISSN 0030364X. doi: 10.1287/opre.37.6.853.

Jennifer Elias. Google asks some employees to share desks amid office downsizing. *CNBC*, 2023. URL `https://www.cnbc.com/2023/02/22/google-asks-some-employees-to-share-desks-amid-office-downsizing.html`. Accessed: 2023-03-03.

Peter I. Frazier, J. Massey Cashore, Ning Duan, Shane G. Henderson, Alyf Janmohamed, Brian Liu, David B. Shmoys, Jiayue Wan, and Yujia Zhang. Modeling for Covid-19 college reopening decisions: Cornell, a case study. *Proceedings of the National Academy of Sciences*, 119(2):e2112532119, 2022. doi: 10.1073/pnas.2112532119. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2112532119`.

Álvaro García-Sánchez, Araceli Hernández, Eduardo Caro, and Gonzalo Jiménez. Universidad politécnica de Madrid uses integer programming for scheduling weekly assessment activities, 2019. ISSN 1526551X.

Gerardo Gonzalez, Christopher Richards, and Alexandra Newman. Optimal course scheduling for United States Air Force Academy cadets. *Interfaces*, 48(3):217–234, may 2018. ISSN 1526551X. doi: 10.1287/inte.2017.0935.

Amy B. Gore, Mary E. Kurz, Matthew J. Saltzman, Blake Splitter, William C. Bridges, and Neil J. Calkin. Clemson University's Rotational Attendance Plan During COVID-19. *INFORMS Journal on Applied Analytics*, 52(6):553, sep 2022. ISSN 2644-0865. doi: 10.1287/inte.2022.1139. URL `https://pubsonline.informs.org/doi/abs/10.1287/inte.2022.1139`.

Erick Guerra, Andrew Sandweiss, and Seunglee David Park. Does rationing really backfire? a critical review of the literature on license-plate-based driving restrictions. *Transport Reviews*, 42(5):604–625, 2022. doi: 10.1080/01441647.2021.1998244. URL `https://doi.org/10.1080/01441647.2021.1998244`.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL `https://www.gurobi.com`.

Timothy R. Hinkin and Gary M. Thompson. SchedulExpert: Scheduling courses in the Cornell University School of Hotel Administration, 2002. ISSN 00922102.

Syl Kacapyr. The unsung engineering behind Cornell's fall 2020 schedule. `https://www.engineering.cornell.edu/spotlights/unsung-engineering-behind-cornells-fall-2020-schedule`, 2021. Accessed: 2022-05-12.

Anjuli Kannan, Gerald Van Den Berg, and Adeline Kuo. ISchedule to personalize learning. *Interfaces*, 42 (5):437–448, sep 2012. ISSN 00922102. doi: 10.1287/INTE.1120.0643.

Biniyam Asmare Kassa. Implementing a class-scheduling system at the College of Business and Economics of Bahir Dar University, Ethiopia. *Interfaces*, 45(3):203–215, may 2015. ISSN 1526551X. doi: 10.1287/inte.2014.0789.

Clarence H. Martin. Ohio University's College of Business uses integer programming to schedule classes. *Interfaces*, 34(6):460–465, nov 2004. ISSN 00922102. doi: 10.1287/INTE.1040.0106.

S. A. Mirhassani. A computational approach to enhancing course timetabling with integer programming. *Appl Math Comput*, 175(1):814–822, apr 2006. ISSN 00963003. doi: 10.1016/j.amc.2005.07.039.

Mehran Navabi-Shirazi, Mohamed El Tonbari, Natashia Boland, Dima Nazzal, and Lauren N. Steimle. Multicriteria Course Mode Selection and Classroom Assignment Under Sudden Space Scarcity. *Manufacturing and Service Operations Management*, nov 2022. ISSN 1523-4614. doi: 10.1287/msom.2022.1131.

George L Nemhauser and Laurence A Wolsey. *Integer programming and combinatorial optimization*, volume 191. Springer, 1988.

Antony E. Phillips, Hamish Waterer, Matthias Ehrgott, and David M. Ryan. Integer programming methods for large-scale practical classroom assignment problems. *Comput Oper Res*, 53:42–53, 2015. ISSN 03050548. doi: 10.1016/j.cor.2014.07.012.

Joseph Pisani and Kailyn Rhone. U.S. return-to-office rate rises above 50% for first time since pandemic began. *The Wall Street Journal*, 2023. URL `https://www.wsj.com/articles/u-s-return-to-office-rate-rises-above-50-for-first-time-since-pandemic-began-11675285071`. Accessed: 2023-03-02.

Tuğba Saraç and Busra Tutumlu. A mix integer programming model and solution approach to determine the optimum machine number in the unrelated parallel machine scheduling problem. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37(1), 2022.

Jan Stallaert. Automated timetabling improves course scheduling at UCLA. *Interfaces*, 27(4):67–81, 1997. ISSN 00922102. doi: 10.1287/inte.27.4.67.

Ofer Strichman. Near-Optimal Course Scheduling at the Technion. *Interfaces*, 47(6):537–554, nov 2017. ISSN 1526551X. doi: 10.1287/inte.2017.0920.

Arabinda Tripathy. School timetabling - a case in large binary integer linear programming. *Management Science*, 30(12):1473–1489, 1984. ISSN 00251909. doi: 10.1287/mnsc.30.12.1473.

Gleb Tsipurkey. The four horsemen of the mandated return to office. *Forbes*, 2023. URL `https://www.forbes.com/sites/glebtsipursky/2023/01/04/the-four-horsemen-of-the-mandated-return-to-office/`. Accessed: 2023-03-03.

UNESCO. Covid-19 response – hybrid learning. `https://en.unesco.org/sites/default/files/unesco-covid-19-response-toolkit-hybrid-learning.pdf`, 2020. Accessed: 2022-05-11.

Niteesh Yadav and Ajinkya Tanksale. An integrated routing and scheduling problem for home healthcare delivery with limited person-to-person contact. *European Journal of Operational Research*, 2022.

Hoda Atef Yekta and Robert Day. Optimization-based mechanisms for the course allocation problem. *INFORMS Journal on Computing*, 32(3):641–660, mar 2020. ISSN 15265528. doi: 10.1287/IJOC.2018.0849.

A. Student-Level Scheduling

A.1. Mathematical Formulation

We have a set $\mathcal{S}$ of students and a set $\mathcal{C}$ of classes. For each class $k \in \mathcal{C}$, $\mathcal{A}_k$ is the set of students enrolled in it and c_k is its social distancing capacity. We denote by $\mathcal{C}_t$ the set of classes scheduled to be ongoing at time t, where t denotes an hour in a week. Sunday midnight corresponds to $t = 0$ and $t = T \triangleq 167$ corresponds to Saturday at 11:00 PM. $\mathcal{T}$ is defined to be the set of hours in a week. A class is ongoing at $t \in \mathcal{T}$ if t falls between its start-time and end-time with a 10-minute buffer on each side.

Given this data, we consider the following scheduling method. Each student $i \in \mathcal{S}$ is assigned to a group, j, $1 \leq j \leq M$, for some integer $M > 0$. Let $\pi_{i,j}$ be an indicator variable that is 1 if i is assigned to group j and 0 otherwise. Since each student can be assigned to only one group, we have

$$\sum_{j=1}^{M} \pi_{i,j} = 1. \tag{1}$$

For each day of the term d, we pick a group j to be assigned to in-person attendance for all of the day's classes. We cycle through all groups one by one for each day of the term. Let $\mathcal{T}_d \subseteq \mathcal{T}$ be the set of hours on day d. Then, the set of students assigned to in-person attendance when group j is assigned to day d is $\{i \in \mathcal{S} : \pi_{i,j} = 1, i \in \cup_{t \in \mathcal{T}_d} \cup_{k \in \mathcal{C}_t} \mathcal{A}_k\}$. If a class has more assignments than the social distancing capacity, we assign some students randomly to an "excess room" for the duration of that class. At time t, if group j is assigned to in-person attendance, the excess room will have s_j^t students, where

$$s_j^t \triangleq \sum_{k \in \mathcal{C}_t} \left(\sum_{i \in \mathcal{A}_k} \pi_{i,j} - c_k \right)^+ . \tag{2}$$

The excess room has capacity E, and if s_j^t exceeds E at any time t or group j, the surplus students would not be able to attend classes online while being on campus.

By design, this produces a fair schedule since every student attends classes in person for a fraction $1/M$ of days. Our scheduling problem then reduces to a group assignment problem. Now, it is imperative that we assign groups in a way that minimizes the excess occupancy. Furthermore, it is undesirable to have a large imbalance in attendance in any given class when different groups are scheduled to attend classes in person. We define the following performance metrics for a group assignment.

- **Surplus Simultaneous Excess (**SSE**)**. This is the maximum number of students assigned to the excess room at any given time. We have

$$\text{SE} \triangleq \max_{1 \leq t \leq T, 1 \leq j \leq M} s_j^t. \tag{3}$$

We are interested in how much SE exceeds over E, the excess-room capacity. That is, SSE =

$(\mathrm{SE} - E)^+$ is our metric of interest.

- **Total Excess (TE).** This is the sum of excesses from each class when each of the groups is scheduled. We have

$$\mathrm{TE} \triangleq \sum_{1 \le j \le M} \sum_{k \in \mathcal{C}} \left(\sum_{i \in \mathcal{A}_k} \pi_{i,j} - c_k \right)^+ . \tag{4}$$

- **Total Deviation (TD).** This is the sum of the absolute difference between an assignment and the "uniform fractional student assignment" that assigns an equal number of students to each of the M groups. The sum is taken across all classes and groups. That is,

$$\mathrm{TD} \triangleq \sum_{1 \le j \le M} \sum_{k \in \mathcal{C}} \left| \sum_{i \in \mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M} \right| . \tag{5}$$

Clearly, a desirable assignment would have a low SSE, a low TE, and a low TD. We formulate this as an optimization problem in which we try to minimize a linear combination of these three metrics. Our decision variables are $\{\pi_{i,j}\}$'s and the only constraints are that they be binary and sum to 1 for each student i as per (1).

Putting it all together, we have the following optimization problem (P):

$$\begin{aligned}
\text{minimize} \quad & \sum_{k \in C} \sum_{j=1}^{M} e_{jk} + \lambda \sum_{k \in C} \sum_{j=1}^{M} \delta_{jk} + \mu s && && (6) \\
\text{subject to} \quad & \sum_{j=1}^{M} \pi_{ij} = 1 && \forall i \in \mathcal{S} && (7) \\
& \sum_{i \in \mathcal{A}_k} \pi_{ij} - c_k \le e_{jk} && 1 \le j \le M,\ \forall k \in \mathcal{C} && (8) \\
& -\delta_{jk} \le \sum_{i \in \mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M} \le \delta_{jk} && 1 \le j \le M,\ \forall k \in \mathcal{C} && (9) \\
& s \ge s_j^t - E && 0 \le t \le T,\ 1 \le j \le M && (10) \\
& s_j^t \ge \sum_{k \in \mathcal{C}_t} e_{jk} && 0 \le t \le T,\ 1 \le j \le M && (11) \\
& \pi_{ij} \in \{0, 1\} && \forall i \in \mathcal{S},\ 1 \le j \le M && (12) \\
& \delta_{jk},\ e_{jk}, s \ge 0 && 1 \le j \le M,\ \forall k \in \mathcal{C} && (13)
\end{aligned}$$

where, $e_{jk}, \delta_{jk}, s_j^t$, and s are auxiliary variables. At the optimum, e_{jk} is set to the excess in class k when group j is scheduled to attend classes in person; i.e., $e_{jk} = \left(\sum_{i \in \mathcal{A}_k} \pi_{i,j} - c_k\right)^+$, δ_{jk} is set to the contribution to TD by class k when group j is scheduled in-person; i.e., $\delta_{jk} \left|\sum_{i \in \mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M}\right|$, s_j^t follows equation (2), and s is set to $\left(\max_{t,j} s_j^t - E\right)^+$. Thus, at the optimal solution, we have $\mathrm{TE} = \sum_{j,k} e_{jk}$, $\mathrm{TD} = \sum_{j,k} \delta_{jk}$, $\mathrm{SE} = \max_{t,j} s_j^t$, and $\mathrm{SSE} = s = (\mathrm{SE} - E)^+$. By equation (6), the above mixed-integer program's objective is to minimize a weighted sum of TE, TD, and SSE. We weigh the three metrics as follows:

- The weight of the TE metric is normalized to be 1.
- The weight of the TD metric, λ, is set so that $\lambda << 1$. In practice, we set $\lambda = 0.25$
- The weight of the SSE metric, μ, is set to so that $\mu >> \lambda$. In practice, we set it to 0 as discussed in Section 2.1.

We define the "uniform fractional student assignment" as the one that sets $\pi_{i,j} = \frac{1}{M}$ for $i \in S$, $1 \leq j \leq M$. The TE of the uniform fractional student assignment is defined to be the uniform excess (UE). That is,

$$\mathrm{UE} \triangleq \sum_{k \in \mathcal{C}} M \left(\frac{|\mathcal{A}_k|}{M} - c_k \right)^+ . \tag{14}$$

Clearly, UE is 0 when, for each $k \in \mathcal{C}$, we have

$$|\mathcal{A}_k| \leq M c_k. \tag{15}$$

If this holds for class k, we say that class k satisfies the "minimal-capacity condition."

A.2. Proof of Theorem 1

Proof. We observe that Part 2 implies Part 1 since TE is equal to the objective function of the integer program when $\lambda = \mu = 0$, which must always be greater than or equal to the optimal objective value of the relaxed linear program. To prove Part 2, consider first the relaxed linear program with $\lambda = \mu = 0$. For simplicity, we can omit constraints (9), (10), and (11) as the objective function no longer has δ or s in it. Let α_i, for $i \in \mathcal{S}$, be the dual variables corresponding to (7) and let $\beta_{jk}, 1 \leq j \leq M, k \in \mathcal{C}$ be the dual variables corresponding to (8). We have the dual problem

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{S}} \alpha_i - \sum_{1 \leq j \leq M} \sum_{k \in \mathcal{C}} c_k \beta_{jk} & \\
\text{subject to} \quad & \alpha_i \leq \sum_{k : i \in \mathcal{A}_k} \beta_{jk} & \forall i \in \mathcal{S}, 1 \leq j \leq M \\
& 0 \leq \beta_{jk} \leq 1 & 1 \leq j \leq M, k \in \mathcal{C}.
\end{aligned}$$

At the dual optimal solution, we will have $\alpha_i = \min_{1 \leq j \leq M} \sum_{k : i \in \mathcal{A}_k} \beta_{jk}$. We can then rewrite the dual problem as,

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{S}} \min_{1 \leq j \leq M} \sum_{k : i \in \mathcal{A}_k} \beta_{jk} - \sum_{k \in \mathcal{C}} c_k \sum_{j=1}^{M} \beta_{jk} & \\
\text{subject to} \quad & 0 \leq \beta_{jk} \leq 1 & 1 \leq j \leq M, k \in \mathcal{C}.
\end{aligned}$$

We observe that if β is optimal, then so is $\bar{\beta}_{\cdot k} = \frac{1}{M}\sum_{j'=1}^{M}\beta_{j'k}$. First, it is obviously feasible. We can also see that the objective value does not decrease with this replacement. The first term is

$$\begin{aligned}
\min_{1\le j\le M}\sum_{k\in\mathcal{C}}\bar{\beta}_{jk}\mathbb{1}\left(i\in\mathcal{A}_k\right) &= \min_{1\le j\le M}\frac{1}{M}\sum_{j'=1}^{M}\sum_{k\in\mathcal{C}}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right)\\
&= \frac{1}{M}\sum_{j'=1}^{M}\sum_{k\in\mathcal{C}}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right)\\
&\ge \min_{1\le j'\le M}\sum_{k\in\mathcal{C}}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right).
\end{aligned}$$

Thus, without loss of generality, we may assume that $\beta_{jk}=\beta_k$. Then, the dual problem objective becomes,

$$\begin{aligned}
\max_{\beta\in[0,1]}\sum_{i\in\mathcal{S}}\sum_{k\in\mathcal{C}}\left(\beta_k\mathbb{1}\left(i\in\mathcal{A}_k\right)-M\beta_k\right) &= \max_{\beta\in[0,1]}\sum_{k\in\mathcal{C}}\left(|\mathcal{A}_k|-c_kM\right)\\
&= \sum_{k\in\mathcal{C}}\left(|\mathcal{A}_k|-c_kM\right)^+\\
&= \mathrm{UE}.
\end{aligned}$$

Since UE is attained by the uniform fractional student assignment, we conclude that the uniform fractional student assignment is optimal when $\lambda=\mu=0$.

Incidentally, for this assignment, $\delta_{j,k}=0$, $1\le j\le M$, $\forall k\in C$. Thus, the uniform fractional student assignment is also optimal when $\lambda>\mu=0$. To show this for the case $\mu,\lambda>0$, it suffices to argue that s is minimized by the uniform fractional student assignment as well. For any t, we observe that $\sum_{j=1}^{M}s_j^t$ is the TE for a problem (P_t) where $\mathcal{C}=\mathcal{C}_t$. For problem P_t, again, the uniform fractional assignment is optimal and sets $s_j^t=\sum_{k\in\mathcal{C}_t}\left(\frac{|\mathcal{A}_k|}{M}-c_k\right)^+$, $1\le j\le M$. Since s_j^t's have the same values for all j and $\max_{1\le j\le M}s_j^t\ge\sum_{j=1}^{M}s_j^t/M$, this assignment also minimizes $\max_{1\le j\le M}s_j^t$. As this holds for each t, we get that $s=\left(\max_{t,j}s_j^t-E\right)^+$ is also minimized by the uniform fractional student assignment. Hence, the uniform fractional student assignment is optimal for all $\lambda,\mu\ge 0$. ■

A.3. Proof of Theorem 2

Proof. We define the minimal deviation of a course k as

$$\mathrm{MD}_k \triangleq 2\cdot|\mathcal{A}_k| \bmod M\cdot\left(1-\left\{\frac{|\mathcal{A}_k|}{M}\right\}\right), \tag{16}$$

where $\{x\}$ denotes the fractional part. That is, $\{x\} = x - \lfloor x \rfloor$, where $\lfloor x \rfloor$ is the greatest integer less than or equal to x. The minimal deviation for the problem is given by

$$\mathrm{MD} \triangleq \sum_{k \in \mathcal{C}} \mathrm{MD}_k \,. \tag{17}$$

Our claim is that for any integral assignment, we have

$$\mathrm{TD} \geq \mathrm{MD}$$

It suffices to prove that MD_k is, in fact, the minimum contribution to the TD by course k. To that end, we fix a course k having non-zero number of students, and consider an optimization problem $(\mathrm{P}^{(k)})$ that minimizes its contribution to the TD. It is given by,

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{M} \left| d_j - \frac{|\mathcal{A}_k|}{M} \right|, & & (18) \\ \text{subject to} \quad & \sum_{j=1}^{M} d_j = |\mathcal{A}_k| \,, & & (19) \\ & d_j \in \mathbb{Z}_+ \cup \{0\} \,, \qquad 1 \leq j \leq M. & & (20) \end{aligned}$$

Here, d_j is the number of students of class k assigned to group j. We note that any feasible solution $\{\pi\}_{i \in \mathcal{S}, 1 \leq j \leq M}$ to (P) corresponds to a feasible solution $\{d\}^{(k)}_{1 \leq j \leq M}$ to $(\mathrm{P}^{(k)})$ for each k: we can set $d_j = \sum_{i \in \mathcal{S}} \pi_{ij}$. The sum of objectives of $(\mathrm{P}^{(k)})$'s is equal to the TD of the solution $\{\pi\}_{i \in \mathcal{S}, 1 \leq j \leq M}$. Thus, the sum of optimal objective values of $(\mathrm{P}^{(k)})$'s is a lower bound on the TD of any solution to (P).

To solve $(\mathrm{P}^{(k)})$, we observe that an optimal solution must exist since the constraint set is non empty. For example, the solution setting $d_1 = |\mathcal{A}_k|$ and $d_j = 0, j > 1$ is feasible.

If $|\mathcal{A}_k|$ is a multiple of M, we can divide the students into M groups of size $\frac{|\mathcal{A}_k|}{M}$ each to get an objective value of 0. The minimum contribution to TD by course k in this case is 0, which is also the value of MD_k.

If $|\mathcal{A}_k|$ is not a multiple of M, we claim that there exists an optimal solution $\left\{d_j^*\right\}_{1 \leq j \leq M}, 1 \leq j \leq M$ such that for each j, d_j is either $\left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil$ or $\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor$. Here $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. To see this, consider an optimal feasible solution $\{d_j\}_{1 \leq j \leq M}$ where this does not hold. That is, $\exists 1 \leq j_1 \leq M$, such that $d_{j_1} \notin \left[\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil \right]$. First, suppose that $d_{j_1} < \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor$. We must then have at least one $1 \leq j_2 \leq M$ such that $d_{j_2} \geq \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil$. We can create a new feasible solution $\left\{d'_j\right\}_{1 \leq j \leq M}$ by setting $d'_j = d_j, 1 \leq j \leq M, j \notin \{j_1, j_2\}$, and $d'_{j_1} = d_{j_1} + 1, d'_{j_2} = d_{j_2} - 1$. We note that this solution is also feasible since $\sum_{j=1}^{M} d'_j = \sum_{j=1}^{M} d_j$ and d'_j is a non-negative integer for $1 \leq j \leq M$. We also observe that the objective value for the solution $\left\{d'_j\right\}_{1 \leq j \leq M}$ is strictly less than that for the solution $\{d_j\}_{1 \leq j \leq M}$. Proceeding this way, we converge to a feasible solution

$\left\{\tilde{d}_j\right\}_{1\leq j\leq M}$ having an objective value no greater than the optimal objective value and the property that $\tilde{d}_j \geq \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, 1 \leq j \leq M$. In a similar fashion, we can get to a solution $\left\{d_j^*\right\}_{1\leq j\leq M}$ such that $d_j^* \in \left[\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil\right], 1 \leq j \leq M$ without increasing the objective value from the optimal objective value. Thus, our claim holds.

We now compute the value of the objective at the solution $\left\{d_j^*\right\}_{1\leq j\leq M}$. Let $L = \left\{j : d_j^* = \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor\right\}$, and $G = \left\{j : d_j^* = \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil\right\}$. To satisfy the feasibility constraint (19), we must have $|G| = |\mathcal{A}_k| \bmod M$, and $|L| = M - |\mathcal{A}_k| \bmod M$. The contribution to the objective value by a group $j \in L$ is given by $\left\{\frac{|\mathcal{A}_k|}{M}\right\}$, while for a group $j' \in G$, it is given by $1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}$. Thus, the total objective value is

$$\left\{\frac{|\mathcal{A}_k|}{M}\right\}(M - |\mathcal{A}_k| \bmod M) + (|\mathcal{A}_k| \bmod M)\left(1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}\right) = 2\left(|\mathcal{A}_k| \bmod M\right)\left(1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}\right).$$

Here, we used the fact that $\left\{\frac{a}{b}\right\} b = a \bmod b$, for any positive integers a and b. Thus, we conclude that the minimum contribution by any course k to the total deviation is given by MD_k. The total deviation is, therefore, bounded below by the minimal deviation, which is defined as $\sum_{k\in\mathcal{C}} \mathrm{MD}_k$. ∎

B. Team-Level Scheduling

B.1. Mathematical Formulation

Since all first-year students attend the same set of classes, we can treat each team as an individual entity, drastically reducing the number of variables. Suppose that we have N teams, and decide to ask K of them to attend classes in person each day for a term comprising T days, where $N \bmod K \equiv 0$. In our case, we have $N = 12$, $K = 4$, and $T = 44$. Let $\mathcal{S}$ be the set of all subsets of size K of $\{1, 2, \ldots, N\}$. We call each such subset an LT group. Then $|\mathcal{S}| = \binom{N}{K} = 495$. We define

$$\pi_{s,t} \triangleq \begin{cases} 1 & s \in \mathcal{S} \text{ is scheduled to attend classes in person on day } t \\ 0 & \text{otherwise,} \end{cases}$$

where $t \in \{1, 2, \ldots, T\}$. We have two notions of fairness.

- Longitudinal balance. Days of in-person attendance should be equally distributed throughout the term. One way to ensure this is to make sure that each team attends classes in person once every $N/K = 3$ days. More formally, we define $\mathcal{B}$ to be the set of $\left\lceil \frac{T}{N/K} \right\rceil$ blocks, each consisting of N/K consecutive days. Note that this may require the creation of additional dummy days. For example, in our case, we have $\mathcal{B} = \{\{1, 2, 3\}, \{4, 5, 6\}, \ldots, \{43, 44, 45\}\}$, where $t = 45$ is a dummy day added to make every element of $\mathcal{B}$ of equal size. We have $|\mathcal{B}| = \left\lceil \frac{T}{N/K} \right\rceil = 15$. Longitudinal balance then requires that

$$\sum_{s:i\in s} \sum_{t:\mathrm{bl}(t)=b} \pi_{s,t} = 1 \quad \forall b \in \mathcal{B}, 1 \leq i \leq N, \tag{21}$$

where $\mathrm{bl}(t)$ is the block in $\mathcal{B}$ containing the day t (which may be a dummy day). We assume that we need to create p dummy days. As noted above, in our case, $p = 1$.

- Balance over days of the week. Since classes follow a schedule that repeats weekly, each team should attend classes in person on each day of the week an equal number of times. Let $\mathcal{D}$ be the set of days of the week, and let $\mathrm{dow}(t)$ denote the element of $\mathcal{D}$ corresponding to day t (day of the week on day t). For $d \in \mathcal{D}$, let $\mathrm{dow}^{-1}(d) \triangleq \{t \in [T] : \mathrm{dow}(t) = d\}$. Then each team should have between $l_{\min}^d \triangleq \left\lfloor \frac{|\mathrm{dow}^{-1}(d)|K}{N} \right\rfloor$ and $l_{\max}^d \triangleq \left\lceil \frac{|\mathrm{dow}^{-1}(d)|K}{N} \right\rceil$ days of in-person attendance falling on day d of the week. That is,

$$l_{\min}^d \leq \sum_{s:i \in s} \sum_{t:\mathrm{dow}(t)=d} \pi_{s,t} \leq l_{\max}^d \quad \forall d \in \mathcal{D}, 1 \leq i \leq N. \tag{22}$$

  For example, in our case $\mathcal{D} = \{M, T, W, R\}$ and $\left|\mathrm{dow}^{-1}(M)\right| = 11$. This means that each team should be able to attend classes in person on Mondays between $l_{\min}^M = 3$ and $l_{\max}^M = 4$ times.

Finally, because we intend to invite exactly one subset of size K to attend classes in person each day, we need,

$$\sum_{s \in \mathcal{S}} \pi_{s,t} = 1 \qquad 1 \leq t \leq T + p. \tag{23}$$

As mentioned earlier, our objective is to encourage mixing among students. We can do this by maximizing the minimum number of times any two teams attend classes in person together. That is, we maximize the quantity

$$\min_{1 \leq i,j \leq N} \sum_{s:i,j \in s} \sum_{t \in [T]} \pi_{s,t}. \tag{24}$$

Note that dummy days are not to be counted when computing the number of times any two teams meet. Putting it all together, we can write the optimization problem as follows:

$$\begin{aligned}
\max_{\pi, m_{\min}} \quad & m_{\min} & & \\
\text{subject to} \quad & \sum_{s \in \mathcal{S}} \pi_{s,t} = 1 & 1 \leq t \leq T + p, & \qquad \text{(23 revisited)} \\
& \sum_{s:i \in s} \sum_{t \in [T+p]:\mathrm{bl}(t)=b} \pi_{s,t} = 1 & \forall b \in \mathcal{B}, 1 \leq i \leq N, & \qquad \text{(21 revisited)} \\
& l_{\min}^d \leq \sum_{s:i \in s} \sum_{t \in [T]:\mathrm{dow}(t)=d} \pi_{s,t} \leq l_{\max}^d & \forall d \in \mathcal{D}, 1 \leq i \leq N, & \qquad \text{(22 revisited)} \\
& \sum_{s:i,j \in s} \sum_{t \in [T]} \pi_{s,t} \geq m_{\min} & 1 \leq i,j \leq N, & \qquad (25) \\
& \pi_{s,t} \in \{0, 1\} & \forall s \in \mathcal{S}, \forall t \in [T+p], & \qquad (26)
\end{aligned}$$

where $m_{\min}$ is an auxiliary variable that, at the optimum, is set to the minimum number of days any two teams are together in person. That is, at the optimum, we have

$$m_{\min} = \min_{1 \le i,j \le N} \sum_{s:i,j \in s} \sum_{t \in [T]} \pi_{s,t},$$

which is consistent with our choice of objective function according to (24).

B.2. Proof of Theorem 3

First, we restate Theorem 3 using the notation established at the beginning of this appendix.

Theorem 3.

1. *The team-level scheduling problem with relaxed integer constraints has an optimum when each day of the term has a uniform fractional group assignment, that is, one where* $\pi_{s,t} = \frac{2}{N(N-1)}$, $s \in \mathcal{S}, 1 \le t \le T + p$.
2. *The optimal objective value when integer constraints are relaxed is* $\frac{TK(K-1)}{N(N-1)}$.
3. *In any optimal solution, each team meets with every other team an equal number of times.*

Proof. Consider the dual of this problem after relaxing the integer constraints. It is given by

$$\min_{\alpha,\beta,\gamma,\delta,\eta} \quad \sum_{i=1}^{N} \sum_{d \in D} l_{\max}^{d} \alpha_{i,d} - \sum_{i=1}^{N} \sum_{d \in D} l_{\min}^{d} \beta_{i,d} + \sum_{i=1}^{N} \sum_{b \in B} \gamma_{i,b} + \sum_{t=1}^{T+p} \delta_t, \tag{27}$$

$$\text{subject to} \quad \delta_t + \sum_{i \in s} \left(\alpha_{i,\mathrm{dow}(t)} - \beta_{i,\mathrm{dow}(t)} + \gamma_{i,\mathrm{bl}(t)} - \sum_{j \in s} \eta_{ij} \right) \ge 0 \qquad \forall s \in \mathcal{S}, \forall t \in [T], \tag{28}$$

$$\delta_{T+r} + \sum_{i \in s} \gamma_{i,\mathrm{bl}(T+r)} \ge 0 \qquad \forall s \in \mathcal{S}, 1 \le r \le p, \tag{29}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \eta_{ij} = 1, \tag{30}$$

$$\alpha_{i,d} \ge 0, \quad \beta_{i,d} \ge 0, \quad \eta_{ij} \ge 0 \qquad \forall d \in D, 1 \le i,j \le N. \tag{31}$$

First, we propose a solution to the dual. We can set,

$$
\begin{aligned}
\alpha^*_{i,d} = \beta^*_{i,d} &= 0, && 1 \le i \le N, d \in \mathcal{D}, \\
\gamma^*_{i,b} &= 0, && b \in \mathcal{B}, \\
\delta^*_t &= \frac{K(K-1)}{N(N-1)}, && 1 \le t \le T, \\
\delta^*_{T+r} &= 0, && 1 \le r \le p, \\
\eta^*_{ij} &= \frac{1}{N(N-1)}, && 1 \le i \ne j \le N, \\
\eta^*_{ii} &= 0, && 1 \le i \le N.
\end{aligned}
$$

We observe that in this setting,

$$
\sum_{i \in} \sum_{j \in s} \eta_{ij} = \frac{K(K-1)}{N(N-1)} \qquad \forall s \in \mathcal{S},
$$

and constraint (28) is satisfied. Likewise, all the other constraints are also satisfied. We observe that the objective value is $\frac{TK(K-1)}{N(N-1)}$.

Now, we need only prove that the uniform fractional group assignment is feasible and achieves the same objective value in the primal. If true, this would imply that it is optimal for the primal and we would have proved Parts 1 and 2.

To that end, we go back to the definition of the uniform fractional group assignment (Definition 2). In this solution, we have that $\pi^*_{s,t} = \frac{1}{\binom{N}{K}}, \forall s \in \mathcal{S}, 1 \le t \le T+p$ and the constraint (23) is automatically satisfied. Now, any $i \in \{1, \ldots, N\}$ is present in $\binom{N-1}{K-1}$ LT groups, and for any $b \in \mathcal{B}$, $|b| = \frac{N}{K}$. Thus, the LHS of constraint (21) becomes

$$
\frac{N\binom{N-1}{K-1}}{K\binom{N}{K}} = 1,
$$

and it is also satisfied. Finally, for any $d \in \mathcal{D}, 1 \le i \le N$,

$$
\sum_{s: i \in s} \sum_{t \in [T]: \mathrm{dow}(t) = d} \pi^*_{s,t} = \sum_{t \in [T]: \mathrm{dow}(t) = d} \frac{\binom{N-1}{K-1}}{\binom{N}{K}} = \frac{\left|\mathrm{dow}^{-1}(d)\right| K}{N},
$$

and by the definitions of $l^d_{\min}$ and $l^d_{\max}$, constraint (22) is also satisfied. We observe that for any $1 \le i \ne j \le N$,

$$
\sum_{s: i,j \in s} \sum_{t \in [T]} \pi^*_{s,t} = \frac{T\binom{N-2}{K-2}}{\binom{N}{K}} = \frac{TK(K-1)}{N(N-1)}.
$$

For $i = j$, this quantity will be higher, and so we need to set $m^*_{\min} = \frac{TK(K-1)}{N(N-1)}$ which is also the value of the objective. Thus, Parts 1 and 2 are proved.

The solution to the dual tells us an interesting thing about the relaxed LP. Because we found an optimal solution with $\eta_{ij} > 0, 1 \leq i \neq j \leq N$, we can conclude, by complementary slackness, that in any primal optimal solution $\tilde{\pi}$, we must have $\sum_{s:i,j\in s} \sum_{t\in[T]} \tilde{\pi}_{s,t} = m_{\min}$ for every pair of distinct i and j, since η_{ij} is the dual variable for primal constraint (25). Thus, Part 3 is also proved. ■

In our case, the dual optimal objective value is equal to $\frac{44\times4\times3}{12\times11} = 4$. Suppose that there exists a solution to the original mixed-integer program with $m_{\min} = 4$. Since this is also an optimal solution of the relaxed LP, we must have $m_{ij} = 4, 1 \leq i \neq j \leq N$. This would mean that every team i is attending classes in person at least $\frac{11\times4}{3} = 14.67$ times. But, every team attending classes in person 15 times is impossible since we would need at least $\frac{15\times12}{4} = 45$ days. Thus, we are left with an optimality gap in our solution. More generally, a necessary condition for a vanishing optimality gap is that both $\frac{TK}{N}$ and $\frac{TK(K-1)}{N(N-1)}$ be integers.

B.3. Other Objective Functions

The problem as it is given above can have multitudes of solutions that all give an optimal $m_{\min}$. Deciding which of these solutions to implement can then be an interesting problem, though in our implementation, we did not pursue this. Since increasing the number of interactions between students is our goal, we can come up with many different objective functions than the one described above. We list some of them here. Let m_{ij} denote the number of times teams i and j both attend classes in person. That is,

$$m_{ij} \triangleq \max_{1\leq i,j\leq N} \sum_{s:i,j\in s} \sum_{t\in[T]} \pi_{s,t}, \tag{32}$$

and let $m_{\max} \triangleq \max_{1\leq i,j\leq N} m_{ij}$.

F1: Maximize $m_{\min}$. This is the objective function described above, and used in real life. While it does encourage different teams meeting multiple times, it may lead to an imbalance due to some pairs of teams meeting many more times than others.

F2: Maximize $m_{\min} - \alpha m_{\max}$. This would mitigate the imbalance between different pairs of teams in terms of their number of meetings.

F3: Minimize $\max_{1\leq i,j\leq N} |m_{ij} - 4|$. This again reduces the imbalance between different pairs of teams by forcing each $m_{i,j}$ to be as close to 4 as possible. Here, 4 is picked because it is the value of $m_{ij}, 1 \leq i \neq j \leq N$ in the optimal solution of the relaxed linear program as per Theorem 3 Part 3.