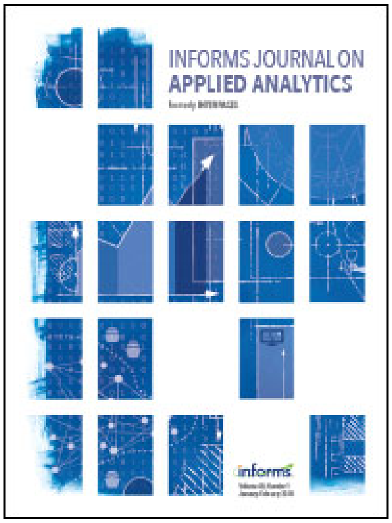


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Developing Optimal Student Plans of Study

R. Alan Bowman

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 51, No. 6, November–December 2021, pp. 409–421
ISSN 2644-0865 (print), ISSN 2644-0873 (online)
http://pubsonline.informs.org/journal/inte

Developing Optimal Student Plans of Study

R. Alan Bowman[a]

[a] Reh School of Business, Clarkson University Capital Region Campus, Schenectady, New York 12308
Contact: abowman@clarkson.edu, https://orcid.org/0000-0001-7961-6323 (RAB)

Received: April 27, 2020
Revised: July 9, 2020; December 1, 2020
Accepted: January 28, 2021
Published Online in Articles in Advance: July 22, 2021

https://doi.org/10.1287/inte.2021.1083

Abstract. Advisors in a small graduate program needed to be able to help students with a wide variety of needs and preferences in terms of starting term, pace of study, program of study, and mode of course delivery to identify plans of study in a dynamic fashion and enable them to follow those plans. Course sections were limited and needed to serve multiple programs and all types of students in those programs. Last-second schedule changes due to overly large or small registration numbers were problematic. Special arrangements to allow students to graduate on time were frequent and costly and lowered academic quality. Analytical tools were developed to help with the planning and alleviate these issues. The tools and the overall approach should be of interest to educational institutions and programs that need to offer a wide variety of students extensive flexibility and choices within a highly constrained scheduling environment.

History: This paper was refereed.

Keywords: course scheduling • integer programming • study plans • timetabling • simulation • applications: simulation • applications: integer programming

Union Graduate College (UGC; which has since become part of Clarkson University) faced many difficulties associated with finding plans of study that met the needs of students in its business and management programs. The primary causes of these difficulties were the highly variable needs of the students and the very limited number of course sections that could be offered across all programs. Little could be done about either cause because UGC needed all of its many types of students to achieve its enrollments and could not expand its course offerings and still process those enrollments in an economically feasible manner.

This paper describes analytical tools that were developed in the dozen years before UGC's merger into Clarkson to help with these issues. During that time, UGC offered 80–90 course sections per year spread over four terms (fall, winter, spring, and summer), with average enrollments per section of 16–18. This volume of enrollments could only be achieved by offering several degrees and certificate programs in a flexible manner. By flexible, I mean that students could start any program in any term with different courses waived based on their background and could proceed at any pace. Courses were offered in multiple delivery modes, and students could mix modes in and between terms. To limit the number of sections in total and achieve sufficient enrollments in each section, all class sections were shared by part-time and full-time students, and the majority of course sections served multiple programs.

Newly admitted students were assigned an academic advisor (a professor in the program), who would help them develop a study plan that needed to take into account their specific needs and preferences, as well as prerequisites and the course schedule. By study plan, I mean the identification of all classes a student planned to take and when they planned to take them to complete their degree. All professors served as advisors, and they possessed and exhibited a wide variety of skills and knowledge levels related to advising. Advising was originally a manual process that resulted in many problems:

- Prerequisites were missed. Although they were picked up by the student information system (SIS) at the time of registration, this led to last-second plan changes.
- Feasible (much less optimal) plans of study for students were difficult to find, especially for students moving quickly or not starting in the summer or fall terms.
- Enrollments overloaded some sections, and some sections were too small to economically run. Last-second changes to schedules were chaotic for all involved.
- Soft prerequisites (that could be violated if necessary) were not uniformly known by the advisors or agreed on by professors and were difficult to exploit, even when known.
- Expensive, time-consuming independent-study courses were often used to patch up scheduling and advising issues.

After a review of related literature, I describe three analytical tools in the same order that they were developed chronologically for use at UGC. In some sense, each tool was designed to supplement and complement the previous tool(s) based on experience with those tool(s) and issues that remained unresolved. They were not conceived or developed simultaneously as a package. Only a preliminary version of the third tool was developed at the time UGC merged into Clarkson; I describe the vision for that tool, as well as its status at the time of the merger. Both of the first two tools were mature and had generated significant benefits across multiple years before the merger.

The tools I describe were developed specifically to address UGC's situation and needs. As such, they will be of most interest to institutions or programs with the following characteristics:

- Student choices are numerous (electives, modes of delivery, etc.), and considering student preferences is of prime importance.
- Student schedules are flexible with little or no cohort structure. New students may start any term and proceed at their own pace.
- Course sections are limited and not easily expanded for resource and budgetary reasons.
- Closing courses as "full" may be problematic in terms of both student course preferences and time of program completion.

These characteristics are suggestive of smaller institutions or programs. Larger programs and institutions with multiple offerings of courses throughout the year also face difficult scheduling problems, but they are different in nature and have received considerable and ongoing attention in the study of timetabling.

Related Work

The vast majority of the literature that addresses educational scheduling focuses on what is known as timetabling. Timetabling addresses both class scheduling and exam scheduling and typically focuses on allocating resources such as courses, classrooms, instructors, and time blocks. The literature on timetabling applied to class scheduling alone is huge and growing. This is undoubtedly partly because timetabling is an NP-hard problem (Bardadym 1996)—and, hence, very difficult to solve efficiently—and because institutions face similar, but not exactly the same, problems and have differing objectives. The literature is ripe with a wide variety of both methodological approaches and applications. See Junginger (1986), Wren (1996), Burke et al. (1997), Burke and Petrovic (2002), and Pongcharoen et al. (2008) for thorough discussions of timetabling and the associated literature. Pongcharoen et al. (2008) and Kassa (2015) also provide excellent literature reviews. Good descriptions of actual applications are numerous; examples include Stallaert (1997), Kassa (2015), and Strichman (2017).

Two timetabling articles of particular interest here are Sampson et al. (1995) and Ogris et al. (2016), in that they included getting students into their most desired electives as part of the timetabling problem structures. The former was even for a Master of Business Administration (MBA) program, albeit a much larger one (Darden School at the University of Virginia) than UGC's, whereas the latter was for students aged 6–15 years. This objective of getting students the courses they prefer is a core part of the tools I will describe, but there are crucial differences in the setting and the approach. At Darden, students expressed their preferences among the courses that were to be offered in a single term, and the courses were then scheduled with those preferences taken into account, along with other standard timetabling issues previously described. Also, the first year was lockstep through the required courses, and it was only the second year with known, continuing students that Sampson addressed.

At UGC, the selections were limited and tightly constrained, to the extent that course selections or even preferences for a single term simply could not be made without considering other terms. To do so would have frequently led to students being unable to find feasible schedules in subsequent terms that enabled them to get the classes they needed and wanted and finish in their desired time frame. Instead, advisors worked with students to develop study plans across multiple terms that would enable them to complete their program requirements and get the electives they most wanted. This was not an easy process. The difficulties were compounded by the fact that students entered the programs and needed plans of study throughout the year and often needed to be able to mix in electives early on to get a better plan. Even with the schedule provided for a full academic year, including the assignment of time slots and professors, finding good plans of study was difficult and time consuming, and the quality of the plans was variable. Thus, the focus at UGC was on student study-plan development, and the role of the institutional schedule was to support that as much as possible. There was a timetabling aspect to UGC's problem (very limited number of time slots), but it was not the primary issue that needed to be addressed.

The first two tools that I developed at UGC did not, therefore, address the development of the institutional schedule. Instead, they required it as an input and focused on enabling students (working with their academic advisors) to better take advantage of it. It was only the third tool that began to address the development of the institutional schedule itself. At that point, the first two tools had been in use for multiple years, and the third tool was undertaken to allow potential schedules to be evaluated based on the results that could be obtained if they were implemented using the first two tools.

As mentioned, the primary constrained resource issue at UGC was time slots. The majority of classes were distributed across only the four most popular time slots—one evening per week for 200 minutes on Monday, Tuesday, Wednesday, or Thursday, with five to seven course sections on each of those four nights. Weekend, late-afternoon, and online sections did ease the burden on those time slots a bit. Students and their advisors needed to know which classes would be in the same time slot each term in order to avoid scheduling conflicts that would make their study plans infeasible. The institutional schedule for an entire year, therefore, was set (with minimal adjustments throughout the year) well in advance of the fall term, so that students could develop their study plans for the full academic year. Even with the full detailed schedule for the entire year, the development of study plans was difficult and time-consuming, owing to the limited sections, limited time slots, and wide variety of student needs and preferences; improving this process was the focus of the first two tools.

The institutional class-schedule development process itself revolved around altering the previous year's schedule. This is not uncommon; the American Association of Collegiate Registrars and Admissions Officers (AACRAO 2016) survey of undergraduate and graduate programs across all 50 states and many countries found that more than 70% of undergraduate programs and almost 70% of graduate programs considered the previous year's schedule to be either very important or important in developing a new schedule. Having some stability in the schedule was crucial for UGC's students to be able to develop plans of study across more than one year. The schedule did, however, require some changes year to year owing to changes in enrollments, faculty availability, and other factors. The institution scheduling need, therefore, was focused more on how to make the necessary changes in the schedule to enable students to come up with study plans that met their needs than it was on many of the typical features of timetabling, such as classroom or teacher availability. This was historically done manually and intuitively at UGC; the objective of the third tool was to make that process more analytical.

The American Association of Collegiate Registrars and Admissions Officers (AACRAO) survey found that slightly more than half of the institutions did not employ any sort of software in their scheduling, and the majority that did utilized software that was homegrown. A goal of this paper is to both provide a possible structure for homegrown approaches and motivate development of commercially available packages to include approaches that face challenges similar to UGC's, in addition to the ones faced by large institutions that have received the lion's share of attention to date.

The AACRAO survey also addressed what they called student scheduling/planning technology, which refers to the students developing a schedule or plan of courses to take. This is more at the heart of the approach I will describe than is timetabling, so it is helpful to review what AACRAO meant by this and to review the literature in this area.

AACRAO defined student scheduling/planning as technology that informs the student of all possible conflict-free schedule combinations of their preferred classes available for immediate registration. They noted that respondents seemed to confuse this with two related topics that they were not specifically asking about:

- Degree audit: Informs students of progress made toward completion of their degree/major; and
- Study-plan development (our term): Allows students to plan courses for terms beyond those that are available for immediate registration.

Even with this expanded focus, only 28% of respondents said that they offered a tool to assist with student scheduling/planning. For the situation faced by UGC, a student-scheduling/planning tool that was separate from the degree audit and study-plan capabilities was practically useless; all three had to be part of an integrated tool. In addition, it made sense to somehow incorporate this tool into the process for developing the institutional schedule, as one of the primary purposes of the schedule was to allow good student scheduling/planning.

Research related to planning from the student's standpoint is more limited and specialized. Head and Shaban (2007) develop the institutional course schedule and the students' schedules simultaneously (which would be ideal), but their approach is for the first year of a program where all students take the same courses. Chen et al. (2014) present an approach to planning coursework from the student's perspective, and they take into account student preferences for time slots and elective courses; their model, however, only considers a single term, and they identify it as a limitation of their model that it cannot be applied to multiple terms. Dechter (2007, 2009) considers student course-plan development across multiple terms and includes both core and elective courses and a maximum course load per term. However, the model does not start with the institutional course schedule, but, rather, assumes that the courses included in the model solution will, in fact, be offered during the terms scheduled by the model and in such a manner that the schedules will be feasible for the students (no overlaps of courses or problems with student availability). Kumar (2017) describes a model that is most similar in both spirit and approach to the one that is at the heart of the approach presented in this paper. They both use integer-programming models as tools for advisors to help students develop course plans across multiple terms. As such, they both include constraints to ensure that course prerequisites are followed and that program requirements are met. These are essential requirements of any approach to developing a multiterm study plan, whether any sort of optimization is attempted. The approaches differ in important ways, however, and these differences stem primarily from the situation that was being addressed and the desired objectives. Kumar (2017) developed and exemplified the approach for several programs in the California State University system, with the explicitly stated goal to increase four-year and six-year graduation rates. Consequently, the objective function was to minimize the time required to complete the degree. Student preferences (beyond choosing the degree program itself) were not treated. In addition, the study plans (called graduation roadmaps) were viewed as suggestions to help students make their choices for the upcoming term, rather than linking directly to registrations in any way. Students could make choices that differed from their roadmap and might have to if they could not follow it either for personal reasons or because they were shut out if a course reached capacity. Although actual implementation was not discussed in the paper, the stated intention was that the students would update their roadmap each term before making choices for that term, so that they would always at least have a feasible roadmap for graduating as quickly as possible.

At UGC, the idea for optimizing study plans followed several years of advisors using an advising application (the first of the three tools I describe in this paper) and arose out of a desire to improve the power and usefulness of that tool by accomplishing the technical aspects of their task far better than advisors could do, while also freeing them up for better interaction with their advisees. Student preferences were always the main focus of advisement, and the preferences needed to consider not only the courses, but also their modes of delivery, because students could freely mix and match modes within and between terms. Study plans were developed as students were admitted (on a rolling basis), and the plans were integrally linked to actual course registration. Students could work with their advisor to update their plan at any time and were asked to do so as soon as they identified the need for an update, but they were only allowed to actually register for the courses on their plan. The advising application tracked intended enrollments from study plans in each course section on a real-time basis, and the linkage between the study plans and registration was crucial to the accuracy of these numbers. Advisors were asked not only to help students develop study plans that best met their preferences, but also to use the advising application to monitor these planned course registrations and to avoid filling up course sections when possible. This was also directly related to meeting student preferences; it was crucial that the schedule retain full flexibility so that future student-advising sessions would not have to work with diminished choices. I describe later in the paper the significant progress that was made using the advising application, but advisors still faced an overwhelming task; optimality was not considered, and the definition of a good plan was simply one that the student agreed to.

Personalized interaction between advisor and student is crucial to good advisement, but the advisor often had to focus so much on the technical aspects of the task that this interaction was diminished. In addition to improving the quality of the study plans by better accomplishing the technical aspects, it was desired to free up the advisors to focus more on this interaction. As the difficulties associated with the technical aspects of advising stemmed from its combinatorial nature and the large number of aspects the advisor was asked to take into account, a study-plan optimization tool using an integer-programming model was developed to achieve both objectives. The model prioritized student preferences both in the objective function and in using constraints to allow for all feasible prerequisite and corequisite flexibilities to be utilized when necessary to better meet student preferences. In addition, as described in detail later in the paper, some model parameters were calculated based on real-time data on planned registrations obtained from the advising application tool. By so doing, the model was able to protect course capacities for future student advisement without affecting the ability to meet the preferences of the student currently being advised. The time to completion (the objective function in Kumar 2017) was a by-product of the student's maximum load per term inputs. A common question that students asked was what their plans would look like if their time to completion was varied. This type of sensitivity analysis was time consuming and difficult to do well, much less optimally, before the model, but was quick and natural with it. This is one example of how the model freed the advisors from the technical aspects of their task (and did that part better than they could) and enabled them to interact with students in a more personalized manner.

Winch and Yurkiewicz (2014) also suggested an integer-programming model to maximize the schedule rating from the student's standpoint, but they focus on a small set of courses and a single term and present their approach primarily as a case study that could be used in a management science course for educational value. Although our approach, with its constantly changing model coefficients, is not particularly suitable to actually build in a classroom setting, the nature of the problem is very intuitive to students, and I have found it to be useful as an example in an MBA management science course.

Scheduling and Advising Tools

Before any tools were developed, UGC converted its scheduling and advising from a term-by-term registration-approval basis to an annual basis. A study plan for each student was developed officially for a full academic year (all four terms, starting with fall), and tentative plans were developed for a student's entire program when it extended beyond the academic year. This original study-plan system was paper-based and, although a significant improvement on the term-by-term registration process, was time consuming and error-prone and had many problems, as described in the opening section.

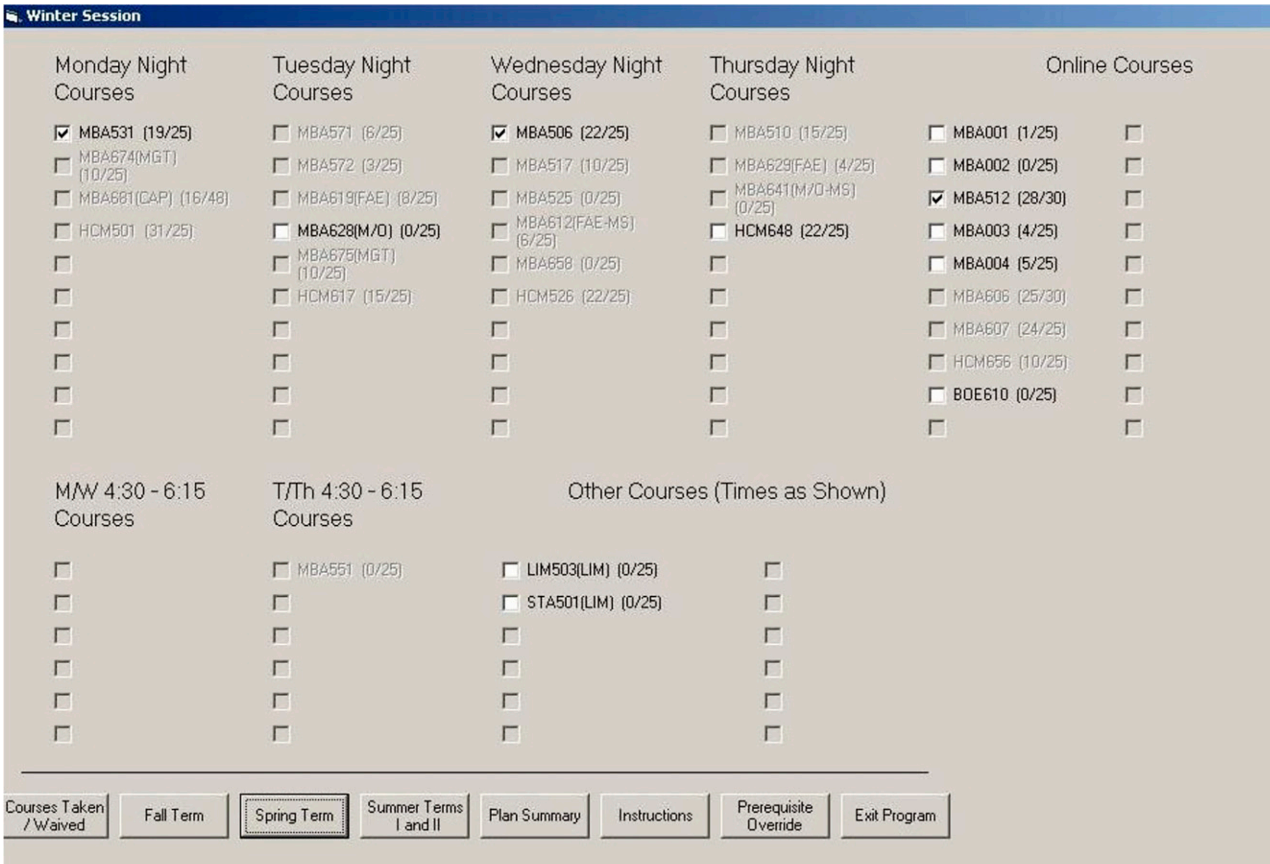
Study-Plan Advising Application

I developed a Visual Basic computer application that allowed the students and advisors to develop study plans much more quickly and accurately. The application was first used for the 2005–2006 academic year. After an opening screen gathered basic initial data, the application used one screen per term (an example is shown in Figure 1) for course selections, while enforcing all logic features (only one course per slot, prerequisites, and each course chosen in at most one slot). The example screen shows these logic features being enforced via disabling illogical selections. One can see how tightly constrained the problem was with only four primary time slots each term and prerequisites disallowing many courses from being selected. The user could toggle from screen to screen to explore various possibilities and do this knowing that the current selections were always feasible and that all enabled course selections could safely be chosen. A summary screen showed the plan for all four terms and progress toward the degree (see Figure 2) and could be easily viewed at any point. Once the selections were satisfactory, they were submitted, and the plan was saved and printed, if desired. The application was immediately successful at getting the students valid plans that did not violate prerequisites and that assured each student was completing all degree requirements. Before its implementation, whenever students found themselves unable to finish their program in their desired time frame by taking actual scheduled courses, it was UGC's policy to conduct independent studies (student working one-on-one with a professor) to enable them to do so. The professors were paid $500 per independent study, their time was taken away from other duties, and the quality was almost always lower, owing to a lack of interaction with other students. Such independent studies were virtually eliminated after previously averaging 10 per year.

The application tallied the number of students planning to take each section of each course and displayed the updated numbers (as seen in Figure 1). This theoretically allowed real-time knowledge and control of student flow into course sections well in advance of registration. It was hoped that this would help with what the dean at the time described as the "constant stress (and sometimes chaos)" (Chudzik 2020) that preceded each term and was due to finding out very late how many people would take each course. There was one situation where a course was split into two sections on the night of the first class meeting, and half the students were forced to go with an adjunct professor, who had agreed to the arrangement that morning. The dean "got a lot of very negative feedback from the students from that situation" (Chudzik 2020). Although the initial implementation helped right away, the results the first year were limited by two related factors: Many students did not create study plans, and their registration was not tied in any way to even having a plan. Subsequently, an arrangement was made with the registrar's office so that registrations were approved if and only if students had a study plan and their registration matched what was on the study plan. Students were told this and were also guaranteed that they would not be closed out of a course section owing to course-capacity limits if they had it on their study plan, so that it was effectively a reservation. They were told that they could always update their plan (it was not a commitment on their part), but that it should always reflect their best current intentions. Section capacities were enforced within the advising system and were only violated if absolutely necessary to avoid forcing a student to attend an additional term.

It was also hoped that advisors would use the real-time numbers of students planning to take each section to proactively protect capacity by advising students into smaller sections when it did not affect their other preferences. To do this well required advisors to look at these numbers for different sections of each course across all terms, and, in practice, they rarely paid attention to the numbers because they focused on the difficult task of finding feasible schedules that mostly met the students' needs. In fact, this tool never was successful at getting students into smaller sections. This feature did, however, allow problem sections to be identified and addressed well in advance of registration. Overly large sections were split, and professors lined up to teach them well in advance of

Figure 1. (Color online) Advising Application Term Screen



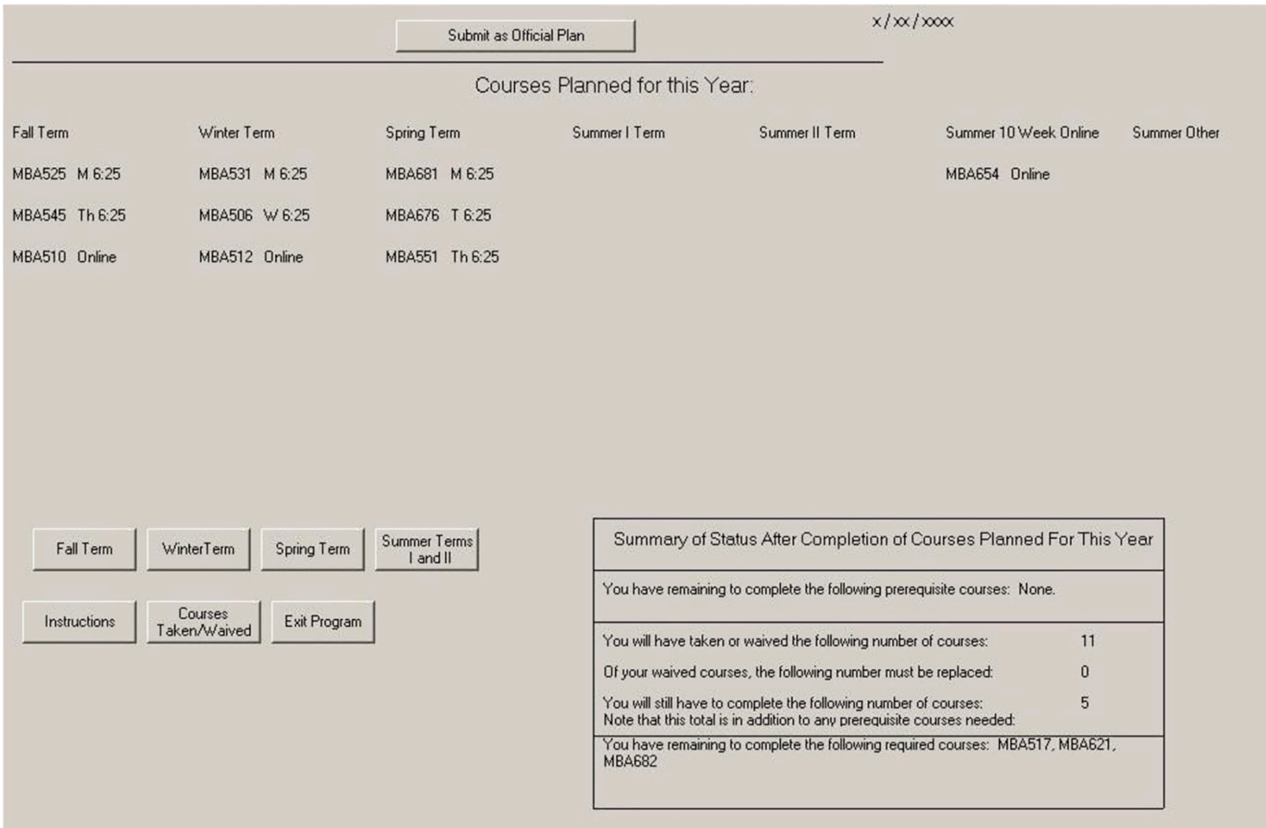
term starts. Small classes were canceled when necessary well in advance and only after students intending to take the course had their study plans satisfactorily adjusted. The second year of use was much more successful after these changes. The "chaos" was virtually eliminated. After the initial adjustments just described, this tool was used with only minor updates and continues to be used at Clarkson after the merger.

Study-Plan Optimization

Although the advising application was very successful, study-plan development was still a tightly constrained, difficult, and time-consuming process. Some advisors were better than others, and some had more knowledge of which prerequisites could be violated if absolutely necessary and when to take advantage of this flexibility. It was especially difficult to develop plans for students who needed to complete their program quickly and for students who started in the winter or spring terms; the course sequencing was better suited for summer and fall starts because they were the majority. International students were an example of those who were particularly difficult. The advisor for international students reported: "Prior to their arrival their 9 or 12 month study plan had to be finalized because it had to be approved by their (overseas) advisor. They had courses waived but often had 12 to 14 courses to fit into three trimesters or possibly four (summer term). It often took me hours to shuffle course schedules around and also change elective courses to find a fit. I would send the schedule to them and often they would request changes to the electives. That made me go through the whole process again. Each iteration took hours" (Chudzik 2020).

New advisors especially found the entire task to be daunting. One new (at the time) advisor reported: "I was not familiar with which courses students should take and when within their program they should take them. So, I was behind the 8-ball and pretty clueless. I did not want to screw up a student's study plan just because I was new. I was concerned about making some huge mistake and then the student would not graduate on time all because of my poor advising" (DeJoy 2020).

Figure 2. (Color online) Advising Application Plan Summary Screen



Also, as mentioned, the application was not successful at spreading the enrollments into smaller sections.

The first step was to standardize and clarify the course relationships as strict prerequisite; recommended as prerequisite, but could be taken simultaneously (corequisite); or recommended prerequisite, but not required. A set of courses was also identified for each program that should be taken as late as possible, notably, including most electives and the capstone course. These features were agreed on by all professors. I then developed an integer-programming study-plan optimization package, which was implemented in Excel using OpenSolver. The complete model is provided in the appendix.

The objective function (Equation (A.1)) was to maximize the sum of the student preference ratings (both course and mode of delivery) across all courses selected, with small rewards for selecting "late program" courses (courses that are better taken late in the program) late, small penalties for selecting a course before a recommended prerequisite was taken, small penalties for selecting a course in the same term as a corequisite course, and a small penalty proportional to the section size (number of students who had already selected that section in their study plans). The penalties and rewards were all chosen to be small enough that their total effect could not exceed one. This ensured that student preferences in terms of both electives and mode of delivery were always maximized, as the preferences were specified as integers. Operationally, this meant that flexibilities with requisite courses were always used; late-program courses were always moved earlier if they enabled a higher student preference sum, and students were put into smaller sections only if this did not reduce their preference sum. There was also a hierarchy among these add-ons; allowing courses to be taken in the same term as a corequisite or before a recommended prerequisite had the largest effect, followed by selecting late-program courses early, followed by section size. The section-size penalty, as historically implemented, subtracted the average section size for the course from that specific section's size. I later realized that this subtraction was redundant for required courses because the total subtraction was a constant, and it did not at all affect electives with only one section for the year. The improved formulation shown would have favored a small over a large elective if and only if the student preferences for them were equal. The constraints (Equations (A.2)–(A.9)) consisted of the following:

Equation (A.2): (A) Each elective course could be selected once at most and only if it was not already taken or waived, and (B) each required course had to be selected exactly once or waived or already taken.

Equation (A.3): One course at most could be selected in each time slot in a term, with the exception of online asynchronous classes, which did not require specific times of the students.

Equation (A.4): The number of courses selected in each term had to fall in a range specified by the student (could differ from term to term).

Equation (A.5): The total number of courses selected plus those waived or taken before the plan had to equal the total number required for the degree.

Equation (A.6): (A) At least one finance/economics elective and (B) one non-finance/economics elective had to be selected or waived or already taken. (C) In addition, at least one elective designated as global had to be selected or waived or already taken. All global electives were also either finance/economics or not and could satisfy those requirements as well.

Equation (A.7): All prerequisites for a course must have been completed before the term that the course was selected.

Equation (A.8): (A) All corequisites for a course must have been completed before or during the term that the course was selected. (B) This constraint defines the number of corequisites completed during the term.

Equation (A.9): This constraint defines the number of recommended prerequisites for a course not completed before the term the course was selected.

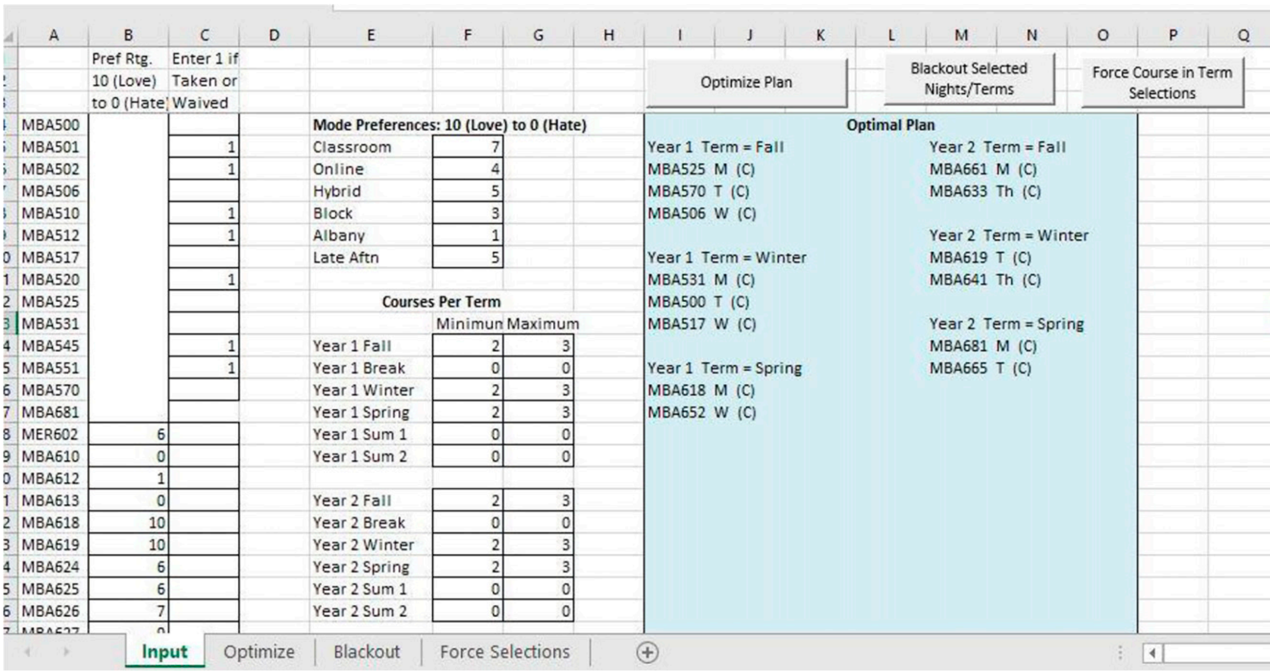
To give some idea of the size of the problem, in one typical year, there were 33 total sections of the 12 courses required of the MBA program. There were 23 electives offered, with only two of them having two sections. The program required seven electives to be taken. Across the entire school, there were 90 sections of courses offered over the four terms (the majority of courses served multiple programs). The programs ranged from six-course certificates to the 19-course MBA and healthcare MBA. There were eight time slots per term and six modes of delivery. The sets of exact terms, time slots, and modes of delivery are provided in the appendix. As mentioned, the majority of the class sections were in the four evening time slots. About 75% of the courses had requisites of some kind. The typical number was one or two, but the capstone MBA course had four prerequisites and seven recommended prerequisites, making up the 11 other required courses. A prominent difficulty in the advising process was to come up with a plan where the four required courses and as many of the seven recommended course as possible were taken before the capstone, while still mixing in the desired electives and obeying the other requisites and course conflicts.

The majority of advising was accomplished with live one-on-one sessions. Gathering the required model inputs from the student took some time, but provided structure and more quantification to what was already an essential part of advising. The study-plan optimization tool was then utilized during the sessions, often multiple times to conduct sensitivity analyses. Models were solved in a matter of just a few seconds and had no effect on session flow; as a result, no attempt was made to either analyze or improve the solution times. I note that the implementation was done on a program-by-program basis, although they all accessed the same real-time database of planned registrations on the completed study plans. This was done both so that the student-input sheet could be tailored to the program and so that the model constraints could be formulated specifically for each program. The size of the problem depends largely on the number of courses in a program and the number of sections offered of each course. The largest UGC programs were 19 courses; larger programs would take longer to solve. Programs with more sections per course would also take longer to solve, but it is less likely that such programs would be interested in the model because readily available sections of all courses on an ongoing basis negate one of the primary needs for the model. Although I had no need to speed up model run times, institutions with similar, but larger, problems might want to at least do some preprocessing based on the individual student situations. For example, course sections for courses that the student had already taken or waived or that had zero preference ratings for both the course and the delivery mode could be eliminated from the model, reducing both the number of decision variables and constraints.

Figure 3 shows the student inputs and the optimized plan for an example student. The worksheet shown is cut off for space reasons; the omitted part is the continuation of preference inputs for all the elective courses. Note that the second worksheet in the file is where the model is solved. The other two worksheets allowed students to force selections into the model and to black out (prevent) selected slots from being used.

It should be noted that the objective function was only maximized based on the inputs from the students, and those inputs were not cut and dried. Assessing preferences for courses was subjective and took time and thought. The same was true for delivery mode. Also, adjusting the minimum and maximum number of courses per term if the optimization indicated that no feasible solution could be found or the solution identified was in some way not satisfactory

Figure 3. (Color online) Study-Plan Optimization Tool



to the student required thought from both student and advisor. I found it more difficult to "train" the other advisors on how and when to do a sensitivity analysis on the parameters than to actually use the tool itself. The majority of advisors had no background in mathematical programming, and sensitivity analysis was not intuitive to them. As an example, when presented with the optimized plan based on their original inputs, students would find something about it that they wanted to improve, such as more or fewer online courses. Adjustments to their inputs, if done in a reasonable manner, could often lead to a plan that they liked better, even though it had a lower objective function value with their original subjective inputs. As another example, one advisor put in the minimum number of courses per term across multiple terms, so that their sum exceeded the number of courses required to complete the program, and did not know why the optimizer was unable to find a feasible solution. In fact, the possibility of making the tool directly available to students was considered, but rejected, for two reasons. First, the students would have even less experience than the advisors with mathematical programming and might encounter situations that would frustrate them. Advisors could and did consult me and others in such situations, and we would share resultant tips with all, so they gradually developed the skills to handle them on their own, but students would not have the opportunity to develop such skills. Second, we wanted to encourage interaction between students and professors outside the classroom in general and thought they would be able to share useful information about courses that students would not know, and that might affect their preferences.

It is worth noting that UGC made a conscious decision to have professors be the advisors, rather than having a centralized advising function. Although we recognized that the latter approach has many benefits (and many professors advocated that approach), we wanted to increase student-professor interaction and not require an additional specialized resource. Consistent quality of advising is a definite advantage of centralized advising, and one clear benefit of the study-optimization tool at UGC was to improve consistency between professors, so that programs that have professors do the advising would reap more of this benefit. The main advantage of the tool, however, was that it optimally performed the technical, combinatorial optimization portion of the task better and faster than any advisor (professor or otherwise) could and freed the advisors up to interact more personally and successfully with the students.

Overall, the study-plan optimization tool was a success and was particularly helpful for newer advisors and those who advised students with the most difficult scheduling situations. The advisor for international students reported, "With the optimizer it took a matter of minutes. It also allowed me to easily initially offer more than one schedule which I did not do prior to the optimizer. Students appreciated that" (Chudzik 2020). The previously quoted new advisor said, "It helped to give me a framework or guide. It was like having an expert advising specialist walk me through the advising process" (DeJoy 2020). He also said the time and cost savings were "incalculable" and cited his own time in developing the plans, the time of more experienced advisors who would have had to help him with questions, the costs that would have been incurred to correct advising errors, and the loss of student satisfaction due to poor advising.

Savings in faculty time alone were estimated to exceed 200 hours per year. The benefits of improved quality of advising are more difficult to quantify. UGC administered the widely used Noel Levitz student-satisfaction survey, which has advising as one of its major scales. Unfortunately, this instrument was not used before the tools described here, so I cannot assess the impact, but after the tools were implemented, advising consistently scored 5.8 or above and ranked as the second-highest scale at UGC behind only instructional effectiveness.

Although largely invisible to both students and advisors, the penalty features also yielded benefits. The prerequisite flexibilities (with penalties) opened up possibilities for plans to better meet student needs that most advisors did not ever previously consider. The section-size penalty helped UGC avoid overly small or large classes by dynamically preserving section capacity for students who really needed it; the coefficient of variation of section sizes dropped from 0.49 to 0.33.

Schedule-Evaluation Tool

As mentioned, the institutional schedule was a required input to both of the tools previously described. As is true at many colleges, its development revolved around being able to effectively make adjustments to the previous year's schedule to take into account changes in anticipated enrollments and in faculty/course availability. Since becoming the part-time associate dean in 2009, I was responsible for developing the schedule each year and did so based mostly on intuition and experience with advising. During the third year of use of the study-plan optimizer, it occurred to me that a natural way to evaluate any potential institutional schedule would be to see how good the optimized study plans would be if the optimizer were used. To get started, I asked advisors to start storing their advisees' optimizer inputs into a database as they used the optimizer.

As the database of sample optimizer inputs was slowly building, I developed the initial (and, unfortunately, final) version of the schedule-evaluation tool. This version consisted of simply allowing the user to build a data set for any institutional schedule and then solving the integer-programming problem for each set of student inputs in the sample-optimizer input database for that schedule. As I was going to be the only (initial) user of this tool, I built it in Statistical Analysis System without any user-friendly features. The database of sample study-plan optimizer inputs was still small at the time that the initial version was developed. Keeping in mind this limited database, I used the tool to evaluate several schedule alternatives. The primary benefit of doing this was some comfort in knowing that the schedule that we went with would have resulted in a very similar quality of study plans (based on the sample-optimizer input database) as the previous year's schedule.

When UGC merged into Clarkson, the structure of the program changed dramatically. In particular, the full-time, on-campus MBA was a lockstep program with day classes, and a separate online MBA was for part-time students. The scheduling issues from both the individual student and institutional standpoint changed fundamentally. The study-plan-advising Visual Basic application tool was retained and continues to be very helpful to proactively manage large and small sections, but the study-plan optimization and schedule-evaluation tools were not carried forward. Admittedly, this schedule-evaluation tool was in its early stages, so I think it is important to at least briefly describe my intentions for further development, so that readers from institutions with situations more similar to UGC's than Clarkson's can properly consider whether they might want to take a similar approach.

Essentially, the tool as it existed had two major limitations that I intended to address. First, it was basically a static simulation because the study-plan optimization problem was solved for each of the sample-optimizer inputs in one batch using the schedule to be evaluated. In reality, students started the programs and needed to develop study plans throughout the year, and the study-plan optimizer used real-time information on section sizes. To properly mimic the actual process required a dynamic simulation. Most of the work to accomplish this would have revolved around getting study-plan-optimizer inputs to "arrive" (have their study plan optimized) over time throughout an academic year to mimic the anticipated arrivals in the coming year. This would have meant having a comprehensive single database composed of the inputs for the continuing (from the current year) students, plus sample inputs for the forecasted student starts in the next year. The former would have been obtained from advisors interacting with their advisees who would be returning and the latter by matching the forecasts by student type (degree, full- or part-time, term start, etc.) with corresponding student-sample-optimizer inputs from the sample-input database that we were slowly building. This was going to be a nontrivial amount of work, but the numeric forecasts of new student starts by term and program were already being produced for other purposes. The schedule-evaluation tool would then, instead of solving the optimization problem for all sample inputs in a batch, solve them in a dynamic simulation, where study plans were developed over (simulated) time. Class-section sizes would be tracked so that the smoothing of enrollments, which was not a part of the static simulation, would be a part of the dynamic simulation. This would have enabled institutional schedule alternatives to have been compared not only in terms of the objective function values, but also in terms of various class-size metrics, and potentially large or small sections could have been identified and adjustments made as part of the schedule-development process. In addition, summaries of the student inputs in terms of the preference ratings for both electives and modes of delivery could have been very useful in selecting which courses to offer and in what format. Note that the demand estimates in this case would have been not only unconstrained (as emphasized by Thompson 2005), but also not binary in nature.

The idea was that the dynamic simulation as described would have allowed alternatives to be compared and sensitivity analyses to be conducted, as is typical of dynamic simulations. As an example of sensitivity analysis, the forecasted student mix could have been systematically altered over likely ranges to see how robust a schedule would be to deviations from estimates. I also intended to attempt some institutional schedule-improvement capability within the tool, perhaps with pairwise exchanges of time slots between courses. In both its more limited static version at the time of the merger and its intended eventual dynamic version, this tool to improve the institutional schedules is very different from timetabling approaches, in that it evaluates alternatives and enables them to be compared, but does not attempt to optimize the schedule in some sense, as do timetabling approaches. The two are not necessarily mutually exclusive; dynamic simulation of a schedule generated by a timetabling approach could have benefits in terms of both evaluation and sensitivity analysis.

Conclusions and Recommendations

UGC developed a series of tools that helped with student scheduling issues associated with small programs that serve a wide variety of students with different needs and preferences. The tools helped UGC achieve more balanced enrollments across course sections with fewer problem (large or small) sections, reduced the need for independent studies, and facilitated the ability to identify plans of study that better met student needs more responsively and inexpensively with less advisor time spent on technical work. These tools should be of interest to programs that face similar issues to those that UGC attempted to address.

Although successful, the tools were homegrown and did not make it to the level of a truly integrated package. Each tool used different software, and the only linkages were databases. Furthermore, none of the tools linked directly to the SIS; the only link was the registrar's office accessing the study plans as part of the registration-approval process. Putting these (or similar) tools into a truly integrated package would have greatly improved both their ease of use and their usefulness. It is hoped that the description of the tools in this paper will not only inform other institutions that face similar structures as they develop their own solutions, but might also aid in the development of a standardized package that could be of more widespread use.

Appendix. Integer-Programming Model

Notation for Courses, Time Slots, Terms, and Modes of Delivery

Courses

c = instance, C = {all courses}, R = {required courses}, FE = {elective courses in finance/economics}, NFE = {elective courses not in finance/economics}, G = {elective courses with global content}, and L = {courses to be taken as late as possible}.

Requisite Courses

r = instance, P_c = {prerequisite courses for course c}, RP_c = {recommended but not required prerequisite courses for course c}, and CR_c = {corequisite courses for course c}.

Time Slots

s = instance, and S = {all time slots} = {Monday, Tuesday, Wednesday, and Thursday nights, Monday/Wednesday late afternoon, Tuesday/Thursday late afternoon, Weekend (Friday–Sunday just before term start), online asynchronous}.

Terms

t = instance, and T = {all terms} = {1, 2, …, 8} = {fall year 1 through summer year 2}.

Modes of Delivery

m = instance, and M = {all modes} = {one evening per week, 2 late afternoons per week, online, hybrid (mix of classroom and online), block(weekend), Albany(one evening per week but in Albany)}.

Decision Variables

• x_i = 1 if course section i is selected, and zero otherwise. (i is the section index; each checkbox in Figure 1 and the other term screens corresponds to a unique i. c_i, s_i, t_i, m_i, and ss_i refer to the course, time slot, term, mode of delivery, and section size for section i. The sections were numbered from term one to eight, in order of the time slots, as listed for set S in the time slots definitions, and from top to bottom within a time slot. So, for example, we see in Figure 1 that MBA506 has been selected on Wednesday night of the winter term of the first year, which is the second term of the planning horizon. There were 33 sections in the first term, fall of year 1, and MBA506 is the 11th section of the second term, so $i = 44$. For this section, c_{44} = MBA506, s_{44} = Wednesday night, $t_{44} = 2$, m_{44} = one evening per week, and $ss_{44} = 22$. This last element is updated in real time, as are section averages, as study plans are added or revised.)

- y_i = the number of corequisites of course c_i taken in the same term as c_i if section i is chosen, and zero otherwise.
- z_i = the number of recommended prerequisites of course c_i not taken before c_i if section i is chosen, and zero otherwise.

Student-Specific Inputs

- $PR(c)$ = preference rating for course c (0–10 integer, higher is more preferred).
- $PR(m)$ = preference rating for mode of delivery m (0–10).
- $W(c)$ = 1 if course c was waived or taken before the two-year plan, and zero otherwise.
- $Min(t)$ = minimum number of courses to be taken during term t.
- $Max(t)$ = maximum number of courses to be taken during term t.

Other Variables

- Tot = total number of courses required for the degree.

Objective Function

$$Max \sum_i [(PR(c_i) + PR(m_i) - 0.00001 ss_i) x_i - 0.01 y_i - 0.01 z_i] + 0.001 \sum_{i:c_i \in L} t_i x_i. \tag{A.1}$$

Constraints

$$\begin{aligned} &A: \sum_{i:c_i=c} x_i + W(c) \le 1, \quad \forall\, c \in C \backslash R,\\ &B: \sum_{i:c_i=c} x_i + W(c) = 1, \quad \forall\, c \in R. \end{aligned} \tag{A.2}$$

$$\sum_{i:t_i=t,\, s_i=s} x_i \le 1, \quad \forall\, t \in T,\ s \in S \backslash \{Online\ asynchronous\}. \tag{A.3}$$

$$Min(t) \le \sum_{i:t_i=t} x_i \le Max(t), \quad \forall\, t \in T. \tag{A.4}$$

$$\sum_i x_i + \sum_c W(c) = Tot. \tag{A.5}$$

$$\begin{aligned} &A: \sum_{i:c_i \in FE} x_i + \sum_{c \in FE} W(c) \ge 1,\\ &B: \sum_{i:c_i \in NFE} x_i + \sum_{c \in NFE} W(c) \ge 1,\\ &C: \sum_{i:c_i \in G} x_i + \sum_{c \in G} W(c) \ge 1 \text{ (only for MBA).} \end{aligned} \tag{A.6}$$

$$\sum_{j:c_j \in P_{c_i},\, t_j < t_i} x_j \ge |P_{c_i}|\ x_i, \quad \forall\, i. \tag{A.7}$$

$$\begin{aligned} &A: \sum_{j:c_j \in CR_{c_i},\, t_j \le t_i} x_j \ge |CR_{c_i}|\ x_i, \quad \forall\, i,\\ &B: y_i = \sum_{j:c_j \in CR_{c_i},\, t_j = t_i} x_j, \end{aligned} \tag{A.8}$$

$$\sum_{j:c_j \in RP_{c_i},\, t_j < t_i} x_j + z_i \ge |RP_{c_i}|\ x_i, \quad \forall\, i. \tag{A.9}$$

References

American Association of Collegiate Registrars and Admissions Officers (2016) Class scheduling (aka timetabling) practices and technology. Accessed January 27, 2020, https://www.aacrao.org/docs/default-source/research-docs/aacrao-sept-2016-class-scheduling-practices-60-second-survey-report.pdf?Status=Temp&sfvrsn=b4a30dda_10.

Bardadym VA (1996) Computer-aided school and university timetabling: The new wave. Burke E, Ross P, eds. *Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science*, vol. 1153 (Springer, Berlin), 22–45.

Burke EK, Petrovic S (2002) Recent research directions in automated timetabling. *Eur. J. Oper. Res.* 140(2):266–280.

Burke E, Jackson K, Kingston JH, Weare R (1997) Automated university timetabling: The state of the art. *Comput. J.* 40(9):565–571.

Chen B, Wang L, Chen W, Luo X (2014) The university course selection problem: Efficient models and experimental analysis. *Analytics Now* 2013-2014:60–76.

Chudzik M (2020) Personal communication via email, June 15.

Dechter A (2007) Model based student academic planning. *Internat. J. Appl. Management Tech.* 5(1):87–104.

Dechter A (2009) Facilitating timely completion of a college degree with optimization technology. *AACE J.* 17(3):215–229.

DeJoy J (2020) Personal communication via email, June 20.

Head C, Shaban S (2007) A heuristic approach to simultaneous course/student timetabling. *Comput. Oper. Res.* 34(4):919–933.

Junginger W (1986) Timetabling in Germany—A survey. *Interfaces* 16(4):66–74.

Kassa BA (2015) Implementing a class-scheduling system at the college of business and economics of Bahir Dar University, Ethiopia. *Interfaces* 45(3):203–215.

Kumar R (2017) A spreadsheet-based scheduling model to create individual graduation roadmaps. *J. Supply Chain Oper. Management* 15(2):165–188.

Ogris V, Kristan T, Škraba A, Urh M, Kofjač D (2016) iUrnik: Timetabling for primary educational institutions in Slovenia. *Interfaces* 46(3):231–244.

Pongcharoen P, Promtet W, Yenradec C, Hicks C (2008) Stochastic optimisation timetabling tool for university course scheduling. *Internat. J. Production Econom.* 112(2):903–918.

Sampson S, Freeland J, Weiss E (1995) Class scheduling to maximize participant satisfaction. *Interfaces* 25(3):30–41.

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

Strichman O (2017) Near-optimal course scheduling at the Technion. *Interfaces* 47(6):537–554.

Thompson GM (2005) Using information on unconstrained student demand to improve university course schedules. *J. Oper. Management* 23(2):197–208.

Winch J, Yurkiewicz J (2014) Case article: Class scheduling with linear programming. *INFORMS Trans. Ed.* 15(1):143–147.

Wren A (1996) Scheduling, timetabling and rostering—A special relationship. Burke E, Ross P, eds. *Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science*, vol. 1153 (Springer, Berlin), 46–75.

Verification Letter

Dr. John W. Huppertz, Associate Professor and Chair, MBA Healthcare Management Program, Clarkson University, Capital Region Campus, Schenectady, New York 12308, writes:

"I am writing to verify that the paper, 'Developing Optimal Student Plans of Study,' that Professor Alan Bowman is submitting for consideration to be published in the *INFORMS Journal of Applied Analytics*, accurately describes work that was completed at Union Graduate College (UGC) prior to its merger into Clarkson University. I was President of Union Graduate College for a portion of the time that he describes and Director of the Healthcare MBA, as well a professor and student advisor for the rest of the time. In these roles, I was able to utilize the tools he describes directly as well as to see the overall benefits they provided to our graduate business programs. We were able to work with students to develop plans of study that better met their needs at the same time that we were able to either avoid or at least anticipate overly large or small course sections and deal with them in ways that were operationally efficient and educationally sound. The tools he developed, implemented, and maintained were a major improvement on the inconsistent and predominantly manual approaches that we had been using. They helped us to both improve student outcomes and achieve efficiencies. They were easy to use (after some brief initial training) and saved us time as student advisors. New professors with no experience at student advising and without detailed curricular knowledge found them to be especially helpful in avoiding errors and helping students get schedules that best met their needs. As he describes in his paper, the tools were homegrown, and improvements both in terms of usefulness and ease of use would have been possible if they could have been integrated into a single package, especially if that package included the Student Information System that UGC utilized.

"When UGC merged into Clarkson (where I am now Director of Healthcare Management), the tools were very helpful in enabling UGC's and Clarkson's business programs to merge, and portions are very helpful even today. Some degree programs underwent significant structural changes, moving toward having separate course sections for individual programs that were lockstep in nature. The tools Professor Bowman describes were geared toward the UGC situation—extremely flexible programs with shared course sections with a fairly small, but critical mass, of total enrollments. His paper should be especially helpful to other programs with those characteristics."

R. Alan Bowman is professor of operations and information systems in the Reh School of Business at the Capital Region Campus of Clarkson University in Schenectady, NY. He received his PhD in operations research from Cornell University in 1990. He has published in many areas, including production scheduling, quality management, and simulation, and has worked as a consultant helping many businesses use OR models to improve the quality and productivity of their operations.