JOURNAL OF BLOCK AND INTENSIVE LEARNING AND TEACHING, 2 (3), 1-22, 2024

Timetabling Courses in Single-Subject Intensive Block Scheduling: Challenges and Solutions

Brian England[a], Debra H. Huber[a], and Rick Rochester[a]

Corresponding author: Brian England (brian.england@umwestern.edu)
[a]Business and Technology Department, University of Montana Western, Dillon, MT, 59725, USA

Abstract

Block scheduling employs non-traditional approaches to the structure of class meeting times. Developing the schedule of courses offered during an academic term presents unique challenges for single-subject intensive block scheduling compared to traditional scheduling. This paper analyzes the key challenges, offers solutions, and identifies areas for future research.

Keywords: block model, single-subject intensive, timetabling, schedule, UCTP

Published by The International Block and Intensive Learning and Teaching Association: www.ibilta.org

URL https://ibilta.vu.edu.au

Introduction

Innovation is a critical competency for successful organizations (Drucker, 1985), including institutions of higher education. One innovation gaining ground among higher education institutions is single subject intensive (SSI) block scheduling. Under this immersive structure, students take one course at a time during a temporally contiguous subdivision of the academic year (a *block*) and progress through their coursework over multiple sequential blocks (Klein et al., 2019; Muscat & Thomas, 2023). In the base SSI approach, instead of the traditional semester-based scheduling approach of taking four different classes over 16 weeks, each meeting for three to four hours per week, students take a single class at a time, meeting for three to four hours per day, for three to four weeks before moving to the next course/class.

The extended and concentrated class period structure encourages the use of innovative pedagogy (Konjarski et al., 2023), and presents unique challenges when creating the course schedule (timetabling). The most impactful challenge in SSI timetabling is exclusivity. In the SSI block scheduling approach, a student, by definition, takes a single course at a time. This creates exclusivity between one section and another.

In this article, we provide a short terminology and literature review, followed by an exploration of the exclusivity challenge and what we view as the prime considerations unique to the SSI-specific university course timetabling problem (UCTP). We then conduct a review of possible methodological approaches to address these challenges, including offering mechanisms and procedures for scoring schedule flexibility. Additional considerations that impact SSI timetablers are discussed before addressing limitations of the work and opportunities for future research.

Literature Review and Terminology

Following Schaerf (1999) and others, we will refer to the process of determining course schedules as *timetabling* and the university course timetabling problem as the UCTP. The output of the process of timetabling is a final list or timetable (sometimes referred to as the course schedule) of the course offerings and class sections for an academic time period. Additional terms are defined below.

Taxonomy and Terminology:

The current major taxonomic divisions (Queen, 2008) of block scheduling approaches, and the taxonomic structure we use in this paper, are as follows:

- **Single subject intensive (SSI):** one course at a time.
- **Dual subject intensive (DSI):** two courses at a time.
- **Rotating block schedule (RB):** more than two courses at a time, on a rotating schedule.

We define a block-term as a particular block within an academic term. We add a suffix to the major taxonomic divisions (SSI, DSI, RB) in order to compactly define the particular structure of a block scheduling model. For example, a schedule with four block-terms per semester would have a suffix of -4S.

We use SSI-4S to indicate a twice per academic year student registration process—students register for classes for a semester at a time, not for a full year or a single block-term—to stay consistent with the definition of a term as an administrative subdivision of an academic year and to allow a compact suffix to encode additional information beyond the number of blocks per span of time.

We will generally restrict our discussion to the SSI model, and for brevity, when context is sufficient the suffix is dropped. We will retain block-term terminology for clarity, although we may drop the suffix when context or alternative nomenclature removes all ambiguity, such as Block 8 (or B8) instead of B4-Spring.

Definitions Regarding Courses and Classes:

Program: a program of study, culminating in an award of a degree or other recognized credential. A program is composed of a set of courses required to establish sufficient knowledge and skill justifying conferral of the degree.

Course: a sub-unit of the program of study that completes a program degree or certificate and is typically measured in credit-hours (e.g., Economics 101).

Class section: a particular offering of a course. There may be multiple sections per term or block (e.g., the Economics 101 section offered in Block 1).

Class session: a meeting of a class section. If referring to a particular day during the class section, an SSI course offering would use the specific block-day.

Core course: a course that is required for all variants of a particular program (e.g., Accounting I is a core course for all business majors).

Elective course: a course that students can choose or elect. Distinguished from a core course in that an elective is not required for all majors. A *required elective* is an elective required for a sub-program (e.g., a minor) and a *free elective* is a course that is not required, but can satisfy all or part of a general program requirement.

Definitions Regarding Time Periods (SSI-specific):

Term: A period of administrative division. This may be a year, an academic year, a semester, a quarter, or month, a week or other period. Typically associated with class registration cadence.

Block-term: a particular block within an academic period (e.g., Block 2 is the second four-week block of the Fall semester).

Block-day: the specific day in a block. For example, B1-D1 is the first class session of Block 1. B2-D8 is the eighth class session of Block 2.

Time-slot: a segmentation of the day reserved for class sessions. Example: AM session (8-11 a.m.), PM session (12-3 p.m.), and evening session (6-9 p.m.).

Definitions Regarding Schedules and Timetabling:

Student course schedule: a student's registration schedule of class sections.

Instructor course schedule: an instructor's teaching assignment schedule of class sections.

Timetabling: the process of determining what class sections are offered during a particular set of block-terms, also known as the process of solving the UCTP.

Timetable: a list of class sections offered during an academic term or year. Includes, at a minimum, course designator code, class session meeting times, instructor, and location.

Cohort crossing: programs of study typically have a recommended sequence and if that sequence is violated, we refer to that as cohort-crossing. For example, at our university, recommended courses for first-year students are general education classes, not core courses for their major. If a first-year student at our university takes a core course recommended for a sophomore or junior, the student crosses the recommendations for her cohort. In this paper, we use the term "cohort-crossing" to specifically refer to students taking courses out of the recommended sequence, irrespective of the specifics of how the sequence mixes general education, core, and elective courses across the program years of study.

Block History and Literature Review

The block scheduling innovation in high school curricula dates to the 1980s, where two main variations of block scheduling developed. The *4x4* variation has students take four 70 to 90-minute classes each day, with a total course duration of a full semester (Lawrence, 2000; Queen, 2008; Zepeda, 2006). A variation known as the *A/B block* also has students take classes for 70–90 minutes; however, students will take six to eight courses, and they are scheduled intermittently, either on Monday–Wednesday or Tuesday–Thursday, and the courses last for the entire academic year (Zepeda, 2006). As many as 52 block design variations have been identified stemming from the 4x4 or A/B designs (Queen, 2008). Both Zepeda (2006) and Jenkins (2002) provide an interesting history of the block system in high schools.

In the United States, Colorado College pioneered block scheduling in the 1970s and teaches SSI 18-day courses, four in each semester or eight per year (Konjarski et al., 2023). Key leaders today in block scheduling at the university level include Colorado College, Victoria University, University of Montana Western (UMW), and the University of Suffolk.

Victoria University, UMW, and Cornell College employ similar length SSI models, while other universities offer variations. For example, Spalding University in the United States offers six-week courses (Spalding University, 2024). Recently, Aquinas College in Michigan piloted the block teaching model beginning in Fall 2023 (Ravaglia, 2023).

The UCTP (sometimes referred to as the UCTTP) is an NP-hard assignment problem (Chen et al., 2021) that is well-studied in operations research as well as other related disciplines. Chen et al. (2021) and Babaei et al. (2015) provide comprehensive surveys of optimization methods, which Chen et al. (2021) divide into six categories: operations research techniques, single-solution heuristics, population heuristics, hyper-heuristics, multi-criteria, and hybrid. The sub-categories of the six main categories include traditional approaches such as mixed integer linear programming as well as more advanced approaches such as genetic algorithms and colony or swarm optimization methods. Both Chen et al. (2021) and Babaei et al. (2015), recognize the widespread use of heuristic approaches to address the UCTP.

The research on timetabling course schedules in higher education has evolved to provide extensive direction and recommendations for universities using a traditional scheduling approach (Babaei et al., 2015; Chen et al., 2021; Kingston, 2022). However, the unique challenges facing institutions using SSI block scheduling have not been sufficiently studied. While traditional and SSI institutions share certain UCTP constraints, such as time-slot exclusivity, the specific path dependency and stacking issues unique to this work, do not appear in the existing literature. This paper analyses the unique SSI timetabling, presents proposed solutions, and explores future opportunities for research.

SSI Timetabling Challenges

The Exclusivity Problem

The most impactful issue in SSI versus traditional timetabling is the problem of exclusivity, as, by the definition of SSI, a student takes a single class section in each block-term (ie, one class at a time). If institutional course demand (cohort size) is such that every course cannot be offered every block-term, then student choice—in terms of the number of possible schedule variations, or *paths,* through the timetable—is reduced with the SSI approach as compared to the semester approach.

Consider an academic year subdivided into two semesters compared to an academic year subdivided into eight blocks. In semester based scheduling, the annual schedule path for a student passes through two decision nodes: Fall and Spring. In SSI block scheduling using eight blocks, the annual schedule path passes through eight decision nodes.

The sequential nodes in the decision tree for an SSI-4S approach are each of the block-terms, resulting in four sequential decisions in a semester, or eight nodes in a year. Instead of making the single choice of choosing, each semester, r sections from n available, a student must choose a single section from each of four different sets of sections, making a total of four choices of a single r_i section from each n_i set. This results in a path-dependency that reduces feasible student schedule combinations. As schedule flexibility increases and feasible combinations increase, we can score the flexibility of a schedule by counting the number of feasible combinations, or paths.

To illustrate the difference in choice for students, we consider a simple model of a single junior year cohort and, in the subsections below, score student schedule flexibility for the exact same semester section offerings for traditional versus SSI-4S scheduling.

The simple model cohort is expected to take a standard set of eight core courses. We make the following assumptions in order to keep our model straightforward in exposition:

- Cohort size and class size require two sections per year to meet demand.
- Each course is offered in a single section each semester.
- There are no prerequisite chains.
- Each course may be taken only once.
- Four courses are taken per semester.
- All students pass all courses.
- All sections are enrolled exactly at capacity.
- Class registration (student schedule choice) occurs once per semester.

In our illustration, CC1 refers to a core course recommended for the junior cohort. CC2 refers to Core Course 2. The load charts in Figures 1 and 2 indicate the distribution of sections across semesters and block-terms.

Comparing Semester and SSI

The exclusivity problem results in SSI timetabling having significantly fewer feasible student schedules than traditional scheduling. Our illustrative model consists of eight total core courses that are offered in two sections per year and one section per semester. The following combinatorics analysis of the feasible paths for semester scheduling compared to SSI scheduling shows the exact same semester timetable results in fewer feasible combinations for the SSI student than the traditional semester student.

Semester Scheduling

The load chart for the semester example is illustrated in Figure 1. Note that there are two sections of each course, and there is a single section for CC1 through CC8 in each semester. There are no exclusions created due to time-slot overlaps, and hence a student may register for any four of the eight core class sections in Fall.

Figure 1. Traditional Scheduling Load Charts, Expositional Model

DAILY LOAD CHART (COURSE, DAY, TIME-SLOT)

TIME-SLOT	MON	TUE	WED	THU	FRI
8:00 8:30 9:00 9:30	CC1	CC5	CC1	CC5	
10:00 10:30 11:00 11:30	CC2	CC6	CC2	CC6	
12:00 12:30 13:00 13:30	CC3	CC7	CC3	CC7	
14:00 14:30 15:00 15:30	CC4	CC8	CC4	CC8	

SEMESTER LOAD CHART

COURSE	FALL	SPRING
CC1	X	X
CC2	X	X
CC3	X	X
CC4	X	X
CC5	X	X
CC6	X	X
CC7	X	X
CC8	X	X

There are therefore two sequential registration choices (Fall and Spring) for the student, and in each of these choices, students choose a subset of sections from a larger set. The combinatorics formula to count the number of non-repeating combinations (student schedule) when picking four of eight choices (registration each semester) is given by the following formula, where n is the count of the pool of choices (all sections) and r is the number of choice picks (student section selections):

$$\frac{n!}{r!\,(n-r)!}$$

In the traditional model, a student chooses four sections from eight available each semester:

First (Fall) semester:

$$\frac{8!}{4!\,(8-4)!} = 70$$

Second (Spring) semester:

$$\frac{4!}{4!\,(4-4)!} = 1$$

Academic year:

$$70 \times 1 = 70$$

Note that in the second semester, the pool of choices (*n*) collapses to four because the student has already completed four of the required eight courses, leaving only four remaining courses as feasible options.

SSI-4S Block Scheduling

We next evaluate the exact same semester timetable delivered in an SSI-4S block schedule format, illustrated in Figure 2. The same set of sections (CC1 through CC8) are offered each semester, but as the courses are delivered in the SSI format, two sections are offered each of four blocks. Again we employ combinatorics to count feasible schedule combinations.

Figure 2. *SSI-4S Scheduling Load Charts, Expositional Model*

DAILY LOAD CHART (COURSE, DAY, TIME-SLOT), BLOCK 1

TIME-SLOT	MON	TUE	WED	THU	FRI
8:00	CC1	CC1	CC1	CC1	CC1
8:30					
9:00					
9:30					
10:00					
10:30					
11:00					
11:30					
12:00	CC5	CC5	CC5	CC5	CC5
12:30					
13:00					
13:30					
14:00					
14:30					

SEMESTER LOAD CHART (COURSE, SEMESTER, BLOCK)

	FALL				SPRING			
COURSE	B1	B2	B3	B4	B5	B6	B7	B8
CC1	X				X			
CC2		X				X		
CC3			X				X	
CC4				X				X
CC5	X				X			
CC6		X				X		
CC7			X				X	
CC8				X				X

In the SSI-4S approach, there are two sets of four sequential choices, and the possible combinations are found in a similar manner. As before, n is the size of the set of possible choices, and r is the number of picked choices.

In the SSI model, a student chooses one section from two available each block-term:

First semester:

$$\frac{2!}{1!\,(2-1)!} \times \frac{2!}{1!\,(2-1)!} \times \frac{2!}{1!\,(2-1)!} \times \frac{2!}{1!\,(2-1)!} = 2^4 = 16$$

Second semester:

$$\frac{1!}{1!\,(1-1)!} \times \frac{1!}{1!\,(1-1)!} \times \frac{1!}{1!\,(1-1)!} \times \frac{1!}{1!\,(1-1)!} = 1^4 = 1$$

Academic year:

$$2^4 \times 1^4 = 16$$

For the exact same Fall and Spring semester schedule offerings, SSI scheduling results in a 73% reduction in feasible student schedules. SSI scheduling, compared to traditional scheduling, reduces student schedule flexibility and increases the potential for impeding students' progress through their program of study.

Mechanics of the Interaction of Exclusivity and Stacking

Exclusivity interacts with what we are calling *stacking* to influence the number of possible schedule paths. Stacking refers to the set of class sections for a given block-term (see Figure 3 - the stack is the vertical visualization of sections offered in a particular block-term).

The SSI-4 timetable from Figure 2 is reproduced as Variant 1 in Figure 3. As we have already shown via combinatorics analysis, this results in 16 possible feasible schedule paths. Variants 2 and 3 are different stacking options for the base model of 16 class sections offered in an academic year, and we count the feasible paths for each variant with a path counting algorithm developed by our colleague (see Acknowledgments, infra).

The effect on feasible number of paths from Variant 1 to Variant 2 are stark, but stacking effects are not always as obvious or dramatic as comparing Variant 1 to Variant 2. Consider timetable Variant 3 in Figure 3, which also has two sections of each course during the academic year. Because of how sections are stacked, for timetable Variant 3 there are only four possible schedule combinations and depending upon registration choices, it may end up being impossible for a student to complete all required courses during the year.

Variant 3 illustrates how path dependency may preclude students from recommended program progression. For example, if a student registers for CC3 in Block 4 and CC7 in Block 5, in Block 7 they cannot register for a class, as they have already completed both CC3 and CC7. Failure to carefully evaluate how the academic year's timetable is stacked can result in students failing to progress according to recommendations. This may result in missed prerequisites, delays in graduation, increased tuition costs, and potential reductions in retention and graduation rates.

Figure 3. *SSI-4S Stacking Variants for Two Sections per Block*

Variant 1	FALL				SPRING			
	B1	B2	B3	B4	B5	B6	B7	B8
16 paths	CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
	CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8

Variant 2	FALL				SPRING			
	B1	B2	B3	B4	B5	B6	B7	B8
1 path	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8
	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8

Variant 3	FALL				SPRING			
	B1	B2	B3	B4	B5	B6	B7	B8
4 paths	CC1	CC2	CC8	CC3	CC5	CC4	CC3	CC4
	CC2	CC6	CC1	CC5	CC7	CC6	CC7	CC8

Comparing Variants 1 through 3, a clue emerges as to how we might use heuristics to timetable, without building a complex linear programming optimization model. Note that the first block of the spring semester timetable is mirrored to that of the first block of fall semester. This creates an either/or between CC1 and CC5 across two blocks. While this constrains the student as to what they can take in Block 1 and Block 5, it still provides choice and it prevents a student from accidentally ending up with a block-term where there is no feasible choice pick, leaving them unable to register for that block-term.

Secondary Considerations

We have illustrated how the stacking choice interacts with the exclusivity problem to affect feasible schedule path counts. As we consider additional factors that may affect an SSI UCTP, secondary considerations arise. Cohort size, program design, and prerequisite chains may have a meaningful impact on the UCTP for SSI scheduling.

Cohort Size

Figure 3 uses Variants 1 through 3 and our path counting algorithm approach to illustrate how stacking impacts student schedule flexibility for a cohort size with two class sections per year. To illustrate how stacking design and an increased cohort size can further interact to impact flexibility, we add a third annual section to each of Variants 1, 2, and 3 and consider the three new stacking designs as Variants 4 through 6, presented in Figure 4.

Adding one section to the base two-section timetable, in most cases, increases feasible path counts. Schedule flexibility and number of feasible paths increase as cohort size increases, due to an increase in the number of sections offered.

Figure 4. *SSI-4S Stacking Variants for Three Sections per Block*

	FALL				SPRING			
Variant 4	B1	B2	B3	B4	B5	B6	B7	B8
(variant 1 base)	CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
33 paths	CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8
	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC1

	FALL				SPRING			
Variant 5	B1	B2	B3	B4	B5	B6	B7	B8
(variant 2 base)	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8
1 path	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8
	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8

	FALL				SPRING			
Variant 6	B1	B2	B3	B4	B5	B6	B7	B8
(variant 3 base)	CC1	CC2	CC8	CC3	CC5	CC4	CC3	CC4
34 paths	CC2	CC6	CC1	CC5	CC7	CC6	CC7	CC8
	CC8	CC7	CC5	CC6	CC4	CC3	CC2	CC1

On one extreme lies a single section per course per cohort, leading to a single path through the schedule. On the other extreme is a cohort size that allows offering all courses every block. For a cohort size that supports offering all eight required sections each block, the number of feasible schedule paths increase to 40,320:

First semester:

$$\frac{8!}{1!\,(8-1)!} \times \frac{7!}{1!\,(7-1)!} \times \frac{6!}{1!\,(6-1)!} \times \frac{5!}{1!\,(5-1)!} = 1680$$

Second semester:

$$\frac{4!}{1!\,(4-1)!} \times \frac{3!}{1!\,(3-1)!} \times \frac{2!}{1!\,(2-1)!} \times \frac{1!}{1!\,(1-1)!} = 24$$

Academic year:

$$1680 \times 24 = 40{,}320$$

Note that even if we have a large cohort with eight sections per block-term, if we follow the implied heuristic in Variant 2 and offer all eight sections of a particular course in a particular block-term, the number of paths again collapses back to a single feasible path. Stacking choice impact does not automatically vanish as cohort size increases.

Our analysis makes clear that timetabling issues are not salient merely for a department, but may even be more impactful for students when we consider university-wide timetabling issues. A department, or major, cohort is a subset of the entire university's enrollment. A university-wide cohort has a substantially higher member count. This has significant implications for general education programs in the SSI model. If the model is changed from a particular third-year degree cohort with eight required classes, to a first-year university-wide cohort with eight required general education subject matter areas, it becomes clear that university-wide coordination of general education timetabling may produce significant benefits for student schedule flexibility. Failure to consider stacking effects in general education timetabling can limit first year flexibility and diminish students' experiences.

Program Requirements

A potentially overlooked source of problems in both schedule flexibility and instructor capacity issues is program design. A typical program in the Montana University System (MUS) is composed of general education (one year), core program requirements (approximately two years), and electives or minors (approximately one year). General education at UMW is a university-wide cohort wherein students typically select one course from each of eight subject categories in order to satisfy their general education requirements. Core program requirements are department-level (major) cohorts, and minors are sub-cohorts of the major cohort.

Program designs for minors may further complicate matters. Consider a minor with a cohort size that limits course offerings to a single section per year. If the student does not take that single section available, they jeopardize their progression, which may impact retention and graduation rates. This problem is amplified if a minor only supports a single section every other year.

This issue can also result in cancelled courses or inability to offer a particular course, depending upon department enrolment. If courses fail to fill, a program design may need to collapse from multiple minors to a single offering, which precludes students from exploring those subject matter areas that are relevant to their intellectual interests or future career plans. While this issue is not unique to block scheduling, program design can act to reduce cohort sizes, and reduced cohort sizes have an outsized impact on block scheduling flexibility.

Prerequisite Chains

Prerequisite chains add an ordering problem to the timetabling process. When students cannot take a class section due to a missing prerequisite, this may eliminate potential paths. Using Variant 1 in Figure 1 as an example, if CC1 is a prerequisite to CC5, then eight potential paths are eliminated from the possibilities set (as the student MUST take CC1 in block 1). If all students are following recommended progression, there will be no students registered for Block 1 CC5 as they must delay taking CC1 until Block 5. This set of delayed students must wait until the next academic year to finish the CC1 to CC5 prerequisite chain. The longer the prerequisite chain, the larger the potential impact on the path possibilities set.

If prerequisite chains are not offered sequentially during a term, this may affect student program progression and increase the instances of cohort-crossing. For example, a business curriculum often will include a sequence of two accounting classes that must be taken before any finance classes. If the finance class is offered in Block 1 and Block 5, and the accounting classes are offered in Blocks 3/4 and 7/8, not only is there potential for students to fail to realize that they have to plan ahead an entire semester, but also program progression is impaired. With this timetable, it is impossible for students to take finance in the block after they finish their second accounting prerequisite. This feature of SSI timetabling may be used either to increase flexibility or to limit cohort crossing.

Solutions

During the process of apprehending and addressing UCTP issues unique to SSI timetabling and our program, we have developed approaches that increase student schedule flexibility, timetabling process efficiency, and other elements of the UCTP. In this section, we proffer what we consider to be the most impactful solutions: stacking heuristics, prerequisites evaluation, program design guidelines, interdepartmental coordination, and evaluation of the timetable using scoring mechanisms.

Stacking Heuristics

We have developed distinct heuristics for core courses and elective courses. In our discussion, *type* refers to whether the course is core, a required elective, or a free elective.

Stacking Core Courses

Our stacking heuristic rules for core courses are as follows, in order of priority:

1. Avoid stacking the same course within the same block.
2. In a two-section per year demand regime, attempt to pair any stacked courses. In a three section per year demand regime, use core diagonal offsets.
3. Mirror semesters where possible.
4. Spread sections evenly across block-terms.
5. Load balance by relevant characteristics.
6. If instructor capacity prevents mirrored stacking, maintain commensurate stack sizes.

Note that prerequisite considerations are covered as a distinct discussion in subsequent sections. Both the order of prerequisites in the timetable and prerequisite chain evaluation (in program design) are important considerations in the SSI UTCP.

The harm from stacking the same course in the same block is illustrated in Figures 3 and 4 by the single path of Variants 2 and 5. The benefit of pairing and mirroring is illustrated by comparing Variant 1 to Variant 3.

When demand exceeds two sections per course per year, a combination of mirrored stacking and core diagonal offsets yields improved feasible path scores. Consider Figure 5 below and compare Variant 4 to Variant 7. Variant 4 mirrors the first two section offerings of Variant 1 (for example, in Block 1 and Block 5, the first two sections offered are CC1 and CC5) and adds the third section sequentially shifted (by one block) to avoid stacking the same course in the same block.

If instead of shifting, the timetabler reverses the sequence (Variant 7, third row), then the number of feasible schedule paths increases from 33 to 81. This effect is seen whether the offset is in complete reverse order or two shorter sequences (see Variant 8). We are referring to this as a core diagonal offset because a line drawn across the section matrix from the middle row of Block 8 to the third row in Block 1 is a diagonal transposition (illustrated via double-headed arrows in Figure 5).

Figure 5. *SSI-4S Stacking Variants for Three Sections per Block*

Variant 1

16 paths

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8

Variant 4

33 paths

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8
CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC1

Variant 7

81 paths

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8
CC8	CC7	CC6	CC5	CC4	CC3	CC2	CC1

Variant 8

81 paths

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC3	CC4	CC1	CC2	CC3	CC4
CC5	CC6	CC7	CC8	CC5	CC6	CC7	CC8
CC4	CC3	CC2	CC1	CC8	CC7	CC6	CC5

Mirroring semesters and diagonal transposition of the third section has the added benefit of making it easier for advisors and students to remember schedule rules and may act to prevent mistakes. Mirroring has the additional effect of making it more difficult to accidentally create prerequisite chain problems. As long as the timetable is well-designed in one semester, replicating it exactly in the second semester reduces the cognitive burden on timetablers, advisors, and advisees, and thereby reduces the odds of mistakes in long-term schedule planning.

Stacking Electives

We use the term *required* elective when a specific course is a requirement of a particular minor. We use the term *free* elective when a course is not a specific requirement, but can satisfy an overall credit-hour requirement for a minor.

Our heuristic rules for electives are:

1. Treat a required elective as a core course if it is required for the majority of minors.
2. For other required electives, treat the elective as core, but ensure that the mirrored pair has options for those that are not in the elective's sub-group.
3. Spread free electives evenly across blocks and use them to balance the load per block.
4. Load balance by relevant characteristic.

A required elective acts as a cohort core class, but for a smaller subgroup of the cohort. At UMW, the current junior year cohort progression recommendation is five required core courses, one to two required electives, and one to two free electives. Two schedule variants for junior cohort courses are presented below in Figure 6 to illustrate how stacking can have a significant effect on student flexiblity.

Figure 6. *SSI-4S Elective Stacking Examples*

Variant Alpha

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC4	CC5	CC1	CC2	CC4	CC5
	CC3				CC3		
REL1		REL3	REL2				REL2
FEL1	FEL2			FEL3	FEL4	FEL5	

Variant Beta

FALL				SPRING			
B1	B2	B3	B4	B5	B6	B7	B8
CC1	CC2	CC4	CC5	CC1	CC2	CC4	CC5
	CC3				CC3		
	REL2		REL3		REL2		REL1
FEL1	FEL2			FEL3	FEL4		FEL5

In this example, CC indicates a core course, REL is a required elective (REL1/REL3 are required for 33% of the cohort; REL2 is required for 67%), and FEL indicates a free elective.

Variant Alpha follows our heuristics for both core and elective courses, and Variant Beta breaks several of the heuristic rules.

Consider the difference in how stacking affects student choice. In Variant Beta, a student ends up not being able to schedule a class for either Block 3 or Block 7, because there is no other choice than CC4, which results in a student only being able to schedule seven courses, putting them behind on program progression. Further, because CC2, CC3, and REL2 are stacked in the second block of each semester, they will only be able to complete two of those three courses, again putting them behind on their program progression. In fact, Variant Beta makes it impossible for a student to complete their recommended junior year program progession, and may make it impossible to take their desired free elective (e.g., FEL2 and FEL4 become exclusive, even if taken out of cohort).

Alternatively, an examination of Variant Alpha makes several things clear. A student who needs to take REL1 for their minor can take REL1 in Block 1, and CC1 in Block 5. A student who does not need to take REL1 (different minor), can take CC1 in Block 1, and FEL3 in Block 5. This not only avoids the exclusion problem but is flexible enough so that the choice of minor does not end up making it impossible to find an as-yet-not-taken course.

Finally, it can sometimes be overlooked that instructors may face the exclusion problem as well. If our aim is to create a stable schedule from year to year, then we must consider what instructors are available, and avoid stacking their subject matter in the same block. If only one instructor is qualified to teach CC2 and FEL2, the timetabler should avoid stacking those courses in the same block, as instructors may not teach two SSI sections per block. Balancing subject matter load across blocks also makes it easier for students to create a schedule that pursues their own intellectual interests using free electives.

It is not always possible to perfectly follow heuristics because instructors take sabbaticals, temporary schedule constraints arise, or other situations occur that make it difficult to perfectly follow the heuristic. In these circumstances, careful study of the possible paths is indicated in order to avoid creating serious progression problems for students.

Load Balancing by Characteristic

By spreading sections evenly across blocks, the chances of a student finding themselves without a course to take is decreased, with an added benefit of smoothing loading for rooms, instructors, and subject matter. By measuring the load balance by all relevant characteristics—block-term, cohort, instructor, room, subject matter, and type—we have eliminated a category of timetable process errors that often create unresolvable conflicts and necessitate last minute schedule changes. These last minute schedule changes can be particularly chaotic and disruptive to student schedule planning.

Prerequisite Chain Design and Heuristics

Addressing the prerequisite challenge is not limited to applying timetabling heuristics that address the ordering problem. Recognizing that the ordering problem can be reduced in complexity by reducing the number of prerequisites and the length of prerequisite chains, we have critically analyzed our chains and tried to eliminate prerequisites that are not absolutely necessary.

The UCTP analysis related to prerequisites has fostered a discussion and review of just what prerequisites still are necessary. For example, some of our prerequisite chains were in place to address administrative issues, such as preventing cohort-crossing or establishing pathways to facilitate assessment points. By reducing prerequisites to only those necessitated by course content preparation, we reduce the ordering problem and increase potential schedule pathways.

We also have reviewed the course content covered for each course in our program curriculum, and made small tweaks to reduce the complexity of prerequisite chains. We have also been rigorous about challenging previous ideas about prerequisites. For example, we had a Microsoft Office course as a prerequisite to Financial Accounting. With further consideration of the pedagogical approach to Financial Accounting, the faculty determined that Financial Accounting does not require Microsoft Office as a prerequisite.

Each institution and each program will have a different experience with respect to prerequisite design. The more specialized and in-depth the program, we expect that the more likely it is that rigor requires longer prerequisite chains. STEM disciplines or technical programs are likely to require longer specific pathways.

For programs with unavoidably long prerequisite chains, we advise modifying the path counting algorithm to score and evaluate different timetable options. An alternative is to formulate an integer programming model, and employ related optimization techniques.

We believe that the net marginal benefit to our institution of developing and maintaining a linear programming model is negative when compared to our current heuristic approach, and so to address the prerequisite issues we use the following guidelines:

1. Load prerequisite chains sequentially across block-terms. The first prerequisite should be at the beginning of the academic year (or semester if mirroring), and the last should be at the end of the academic year (or semester if mirroring).
2. Follow core course heuristics.

3. If courses early in the prerequisite chain have more demand than the courses farther down the chain, it may be acceptable, instructor capacity permitting, to stack the early prerequisite and break the core heuristic of avoiding stacking sections of the same course.

As an illustrative example, our Bachelor of Science (BS) in Business Administration requires a prerequisite sequence of Math (FY) -> Accounting 1 (SO) -> Accounting 2 (SO) -> Finance (JR) -> Capstone (SR). Variant Mu in Figure 7 is our solution to sequencing this prerequisite chain. We offer ACTG 201 in Blocks 1, 2, 5, and 6. We offer ACTG 202 in Blocks 3 and 7, and we offer Finance (BFIN 322) in Blocks 4 and 8.

Note that our design (Variant Mu) facilitates cohort crossing, even though we discourage the practice. The reality is that cohort crossing happens, and rather than make it more difficult for students who have made mistakes in scheduling (or for transfer students or students changing their major) to complete a prerequisite chain, we have determined that the benefits of making it easier for students to get back on track outweighs the benefits of building a sequence that makes it harder for students to cohort cross.

We face another consideration outside of the ordering problem for business majors when timetabling this sequence. Accounting 1 (ACTG 201, or Financial Accounting) is a requirement of other programs as well as for our Bachelor's of Science in Business Administration program, and typically has a meaningful portion of non-business majors enrolled. The full prerequisite chain (ACTG 201 through Capstone) is completed only by business students, and not these non-business majors. Hence, demand for ACTG 201 is greater than demand for ACTG 202.

In such a case, Variant Nu in Figure 7 may be acceptable (even though it breaks the heuristic of avoiding stacking core courses) depending upon the proportion of demand that comes from students that will not complete the full prerequisite chain. In our case, Variant Mu is our preferred solution when considering other departments' schedules, but we note that circumstances may vary across institutions. When a course is a requirement of multiple programs and part of a prerequisite chain, careful evaluation and interdepartmental communication is indicated.

Figure 7. *SSI-4S Prerequisite Chain Examples*

Variant Mu	FALL				SPRING			
	B1	B2	B3	B4	B5	B6	B7	B8
SOPH	ACTG201	ACTG201	ACTG202		ACTG201	ACTG201	ACTG202	
JUNIOR				BFIN322				BFIN322
SENIOR	CAPSTONE STRINGER				CAPSTONE STRINGER			

Variant Nu	FALL				SPRING			
	B1	B2	B3	B4	B5	B6	B7	B8
SOPH	ACTG201		ACTG202		ACTG201		ACTG202	
	ACTG201				ACTG201			
JUNIOR				BFIN322				BFIN322
SENIOR	CAPSTONE STRINGER				CAPSTONE STRINGER			

PREREQUISITE SEQUENCE:

ACTG 201--> ACTG 202 --> BFIN 322 --> CAPSTONE

Prerequisite chains add complexity to the block timetabling process by introducing an ordering problem. While some of this problem can be ameliorated by strict adherence to only academically necessary prerequisites, prerequisites are an inherent feature of programs, and thus require consideration when solving the UCTP.

While we have primarily discussed challenges unique to SSI, the block scheduling approach has a unique advantage over traditional semester scheduling when it comes to prerequisite chains. Students taking courses in the semester model can only take two courses in a chain per academic year, whereas theoretically, students in an SSI-8Y model could complete an eight course prerequisite chain in a single academic year.

Program Requirements Design

Our university currently offers a Bachelor's degree in Business Administration with a required minor chosen from a set of specific discipline minors. Across institutions and disciplines, program requirements often are defined in terms of a one-year general education requirement, a roughly two-year program of core requirements for the major, and then a roughly one-year program with a set of specialized courses for the narrower discipline (e.g. minor).

A minor program with a separate set of required, non-core, elective courses creates its own small cohort. It reduces schedule flexibility because it results in a smaller sub-group cohort. If demand exceeds capacity, then students will automatically cohort-cross, making demand for core program courses more variable through time and potentially making it harder for students to complete their program in the recommended progression.

A minor program with a smaller set of required electives mitigates this issue. Consider a department with three minors, each with equally sized cohorts that each require one section per year to meet demand. These three minors include: (a) a minor with eight unique required courses, (b) a minor that requires REL1, REL2, and REL3 and then requires the student to select their remaining five electives from a pool of free elective courses, and (c) a minor with REL2 as a required elective while requiring the student to select their remaining seven electives from a pool of free elective courses.

The minor with eight required courses can only offer a single cohort schedule per year with one path. But for the other two minors, two of the courses (REL1, REL3) are offered in a single section, one of the courses (REL2) is offered in two sections, and then a department can offer a rotating schedule of electives to allow students to complete their program requirements. In addition to student flexibility, this program design also impacts capacity planning, as it relaxes the instructor exclusivity constraint. It also facilitates, if desired by students or instructors, a much wider topical set of electives to be offered.

Interdepartmental Coordination

Two impactful inter-departmental coordination issues that can affect student progression are the general education requirement and non-department majors sharing department courses. As previously illustrated, the former can benefit from thoughtful coordination as load and subject matter balancing can benefit student choice and general education completion in their first year. Non-department majors sharing department courses will vary among departments, as the primary issue is either (a) psuedo-shared (across departments) core, or (b) demand for minors.

At UMW, several other departments have determined that our department's courses are relevant for their programs, and so we have meaningful demand for these courses from students who are not business majors nor business minors. This demand will be seen only in these shared-core courses and not the rest of our department's core courses, and so these courses (which are also core requirements for business majors) will always have a higher demand than our junior core courses.

Working with these other departments is vital: If other programs require one of our courses as core, then they must treat it as a core course when timetabling their program, and our department must consider how to coordinate their program timetable with our timetable. If we fail to consider inter-departmental impacts, we risk reducing flexibility for these non-business students and potentially making recommended program progression impossible.

Scoring Mechanisms

We have developed a suite of management metrics to help us manage our timetabling process. We measure path counts, load balance (by cohort, by subject matter, by instructor, by block-term), expected class size (via demand forecasting), and instructor preference scoring. These measurements are, for the most part, calculated by our automation efforts described below.

Path counting and load balances are our primary metric to evaluate student schedule flexibility, and are automatically recalculated each time we iterate the timetable. We evaluate instructor preference using a simple and efficient approach that automatically recalculates scoring metrics as we iterate the timetable. Instructor preference score is calculated by asking each faculty to score time-slot (a.m. or p.m.), block-term, format (online or face-to-face), and course. The scores are normalized, aggregated, and reported for each timetable iteration. This transparency to all faculty in the department not only helps optimize faculty preference, it helps to ensure that all faculty feel that they have agency, choice, and that the timetabling process is fair.

We have automated the timetabling scoring process via an Excel spreadsheet. Well-structured (defined fields and defined primary keys) data tables for the course catalogue, course section timetable, instructor master, and instructor preferences are designed, created, and populated. By entering the appropriate rows and values in these tables, the various load charts and load metrics are automatically created from one set of inputs. When the timetable is final and ready to send to the registrar, using well-structured data tables enhances efficiency and reduces errors in upload file preparation.

Discussion

We view exclusivity, proper stacking, and cohort size as the prime considerations for SSI timetablers and have presented what we view as the most impactful solutions in the previous section. As we have worked to continually improve our timetabling approach, we have identified additional issues and solutions that we view as less impactful than our prime considerations, but that practitioners are likely to find useful. In this section, we discuss some of these additional considerations and solutions, our view of the limitations of this paper, and our suggestions for future research.

Subject Matter Load Balancing

Instructor schedules share the same exclusivity problem as student schedules, but instead of a schedule constraint based upon what course has not yet been taken and prerequisite requirements, instructor assignment is constrained by subject matter expertise and availability.

If we think of subject matter area expertise as a single course, it is straightforward to see that how the timetabler stacks subject matter is impactful in the same way stacking sections of the same course is impactful. If we want to maintain a flexible schedule, both for students and instructors, then we should be thoughtful about how we stack subject matter expertise.

Given that we are a relatively small department in a small community, it is difficult to find, on short notice, replacement instructional capacity, especially for narrow subject matter areas. Our work on course stacking has illustrated to us that subject matter stacking can have a similar and meaningful impact on instructional capacity flexibility. If we stack subject matter A, taught by two instructors in the same block, if one instructor becomes unavailable and a replacement for subject matter A cannot be found, we must cancel one of the sections.

If instead of stacking one subject matter, we stack two different subject matters, with overlap of the instructors' skill sets, we can now look for a replacement instructor for subject matter A or subject matter B, because if both existing instructors can teach either A or B, we can rotate one to replace the other. This has informed our staffing decisions by emphasizing the value of the generalist when recruiting new faculty.

The same benefits that accrue to instructional capacity flexibility benefit students as well. When subject matter areas are stacked, student choice in pursuit of their individual intellectual curiousity becomes limited. For example, if all of the elective courses on quantitative methods are stacked in a single block, then students who want to investigate a variety of quantitiative methods are limited to a single course, and they may be forced into other, less appealing, electives in order to complete their program on the recommended timeline.

Advising Best Practices

We place importance on helping students design four year plans. While no plan survives contact with the enemy, encouraging students to plan helps to avoid dead-ends in their schedules, and teaches them important lessons about the benefit of long term planning.

We encourage students to take the recommended sequence of study for their cohort. Reducing cohort-crossing makes forecasting demand more accurate, and if the demand forecast is more accurate, we are less likely to see schedule exclusion problems.

Online Stringers

As we seek to continue to innovate and serve students, we are currently exploring offering select courses as *online stringers*, which are 16 week online class sections. The Montana University System (MUS) is currently experimenting with a platform that allows any student in the 16 universities of the MUS to register for any online course offered at any MUS institution, and our experimentation complements this strategic effort. We anticipate that offering online stringers will not replace our standard block schedule, but rather will help students remedy gaps or progression problems with respect to their program requirements.

Path Counting Algorithm

Prior to development of our colleague Dr. Seacrest's algorithm, we identified the number of feasible paths by hand via manual combinatorics analysis using tree diagrams. This process was subject to human error. Dr. Seacrest's short Java program (approximately 120 lines of code) is a brute-force algorithm. The general structure of the algorithm is as follows:

- Declare variables: core courses, elective courses, and block-terms.
- Generate a set of all possible timetable combinations.
- Step through each row, evaluate feasibility, and flag as feasible or not feasible.
- If feasible, increment a counter.
- When the possible set loop is complete, report the value of the counter to console.

Limitations and Opportunities for Future Research

We have identified the following issues as limitations in our work presented herein:

- We have uncovered no research on the UCTP applied specifically to block scheduling. As an undeveloped field, there may be unexplored critical considerations.
- It is likely that other institutions utilizing block scheduling have identified additional unique issues as well as have identified solutions that differ from those offered in this paper, and those issues and solutions are not covered herein.
- Our work focuses on the administrative and operational aspects of the UCTP and does not investigate linkages to student success, pedagogy, or the academic role of timetabling.
- We have not exhausted all possible heuristics. Our heuristic development was informed by first identifying the problem, then reactively identifying a solution to the problem, and finally evaluating the impact of that solution.
- We have not validated our heuristics with an integer programming approach or evaluated other formal analytical optimization approaches.
- The generalizability of our heuristics and other solutions are unknown until they are tested and studied at other institutions with different characteristics.
- Personal beliefs, experiences, or expectations may influence the way we interpret our solutions. The potential for researcher bias can compromise the objectivity of our analysis.

We believe that opportunities for future research on the block scheduling UCTP include:

- Evaluation of the exclusion and stacking problem for additional cohort sizes (number of sections) would be a useful extension of the work herein. In this paper we have focused on one to three section cohort sizes, but other institutions may have different demand levels. More in-depth analysis and codification of the interactions among exclusivity, stacking heuristics, and cohort size would be a meaningful addition to the literature.
- Development of an SSI specific integer programming optimization approach, or similar method, would link the UCTP optimization literature to SSI timetabling. There is a vast existing literature on UCTP optimization as applied to parallel semester timetabling that could be adapted to the SSI UCTP.

- A comparative study of multiple institutions' and departments' UCTP solutions would help identify common challenges and alternative solutions to SSI timetabling. Konjarski et al. (2023) examined five institutions committed to the SSI model on other (non-timetabling) dimensions; this may provide a model for further study of how our UCTP problems and solutions affect student outcomes.
- Gathering student perceptions through qualitative and quantitative data collection and analysis could provide valuable insights on current and future UCTP solutions. Swain (2016) and Klein et al. (2019) found a preference for block scheduling versus traditional semester scheduling. Future research could employ similar evaluation models of student perspective related to the importance of schedule flexibility, or student perceptions of impacts to their success when they cannot finish a recommended sequence in a recommended time period.
- An analysis of the relationship between student schedule flexibility and measures of student success, such as graduation rates and retention, could quantify the importance of careful timetabling in block models. Future work could build on the benefits of centring student success via the block model, which have already been studied at Victoria University (Tangalakis et al., 2024).
- Future work could also potentially address the relevance of these findings to pedagogical approaches, timetabling's importance in block models, and the academic role of timetabling for institutions using SSI scheduling.

Conclusions

Our key findings are that exclusivity, stacking, and cohort size for instutions using an SSI block scheduling approach presents unique and impactful challenges to the UCTP. Student schedule flexibility, as measured by counting feasible schedule paths, is inherently reduced under an SSI approach compared to a traditional scheduling approach. We have developed solutions to these key challenges, and other challenges unique to SSI scheduling, and begun a conversation on how block scheduling affects the UCTP.

Institutions using block scheduling approaches may benefit from applying the heuristic solutions we have devised in response to the challenges, or they may find our exposition of the challenges a path towards further innovation of their own with respect to the UCTP.

Disclosures

No conflict of interest, financial or otherwise, are declared by the authors.

Acknowledgments

Dr. Tyler Seacrest (University of Montana Western Math Department) was extremely helpful in developing the Java code used to count the feasible schedule path possibility set. Readers who are interested in obtaining a copy of the code can contact the authors.

References

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, *86*, 43-59. https://doi.org/10.1016/j.cie.2014.11.010

Chen, M. C., Goh, S. L., Sabar, N. R., & Kendall, G. (2021). A survey of university course timetabling problem: perspectives, trends and opportunities. *IEEE Access*, *9*, 106515-106529. https://doi.org/10.1109/ACCESS.2021.3100613

Drucker, P. (1985) *Innovation and entrepreneurship: Practices and principles*. Harper & Row.

Jenkins, E., Queen, A., & Algozzine, B. (2002). To block or not to block: That's not the question. *The Journal of Educational Research*, *95*(4), 196–202. https://doi.org/10.1080/00220670209596592

Kingston, J. H. (2022, August 30-September 2). Timetabling research: A progress report [Paper presentation]. 13th International Conference on the Practice and Theory of Automated Timetabling, Leuven, Belgium. https://patatconference.org/patat2022/proceedings/PATAT_2022_paper_23.pdf

Klein, R., Kelly, K., Sinnayah, P., & Winchester, M. (2019). The VU way: The effect of intensive block mode teaching on repeating students. *International Journal of Innovation in Science and Mathematics Education*, *27*(9), 47–59. https://doi.org/10.30722/IJISME.27.09.004

Konjarski, L., Weldon, J., Ashley, S., Freeman, T., Shanata, J., Yamanishi, M., Lotz, E., Gilde, C., & Ganzel, A. (2023). The block: A catalyst for ongoing innovation. *Journal of University Teaching and Learning Practice*, *20*(4), https://doi.org/10.53761/1.20.4.13

Lawrence, W. W., & McPherson, D. D. (2000). A comparative study of block scheduling and traditional scheduling on academic achievement. *Journal of Instructional Psychology*, *27*(3), 178.

Muscat, A., & Thomas, M. B. (2023). Teaching on the block: An exploration of university educators' experiences of block teaching in higher education contexts. *Journal of Block and Intensive Learning and Teaching*, *1*(2), 32–48. https://doi.org/10.15209/jbilt.1301

Queen, J. A. (2008). *The block scheduling handbook* (2nd ed.). Corwin Press.

Ravaglia, R. (2023, October 25). Aquinas College boldly embraces block scheduling: One class at a time. *Forbes*. https://www.forbes.com/sites/rayravaglia/2023/10/25/aquinas-college-boldly-embraces-block-schedulingone-class-at-a-time/

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, *13*(2), 87–127. https://doi.org/10.1023/A:1006576209967

Spalding University. (2024). *Block Schedule*. https://spalding.edu/block-schedule/

Swain, M. (2016). Block teaching and the three A's: Attendance, attainment, and attitudes. *Innovations in Practice*, *10*(1), 33–38. https://doi.org/10.24377/LJMU.iip.vol10iss1article57

Tangalakis, K., Hoang, C. H., Knight, E., Hurley, P., Jackson, J., & Hildebrandt, M. (2024). Differentiation through innovation in the contemporary higher education environment: The case of the 'Victoria University Block Model®.' *Innovations in Education and Teaching International*, *61*(6), 1305–1319. https://doi.org/10.1080/14703297.2024.2329192

Zepeda, S. J., & Mayers, R. S. (2006). An analysis of research on block scheduling. *Review of Educational Research*, *76*(1), 137–170. https://doi.org/10.3102/00346543076001137