Guiding Course Selection via Degree Evaluation Optimization

A Major Qualifying Project (MQP) Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in

Mathematical Sciences,
Industrial Engineering

By:

Lindsey Fletcher

Advisor:

Andrew Trapp, PhD

Abstract

This project introduces two new integer linear programs (ILP) to assist students with course selection. These ILPs use novel mathematical constructs, such as "collections" of courses that may always be interchanged. The first integer linear program minimizes the additional credits needed for degree completion, and the second increases flexibility. Additionally, degree completion rules are modeled with a framework that captures the "real-world" complexity and is sufficiently standardized to be adapted to a user interface. Computational experiments demonstrate the potential of the model to identify efficient schedules and perform "what-if" analysis. The modeling proposed by this project has the potential to be extended to a visual tool that assists students with choosing effective schedules.

Acknowledgements

I would like to thank my advisor, Professor Andrew Trapp, for his continued mentorship and support throughout my time at Worcester Polytechnic Institute. I would also like to express my appreciation for the immense support from my friends, roommates, and Ty.

Contents

List of Tables

List of Figures

1 Introduction

The average six-year graduation rate at four-year colleges and universities in the US is only 64% [1]. With tuition rates at an all-time high and increasing, delayed graduation can be very expensive for students. Students who need more time to complete college must pay more in tuition and face the opportunity cost of delayed entry into the workforce. The total cost of delaying graduation from four to five years has been estimated to be at least $88,000 [2]. Increased graduation time has also been linked to lower mid-career earnings, even after controlling for demographic and institutional factors [3]. Alarmingly, undergraduate enrollment more than doubled between 1970 and 2009, but the completion rate has been "virtually unchanged" [4]. Planning efficient schedules is challenging for students due to the complexity of university requirements and variability in offerings. Many students also graduate with more credits than required. According to "Complete College America", students who receive a bachelor's degree take an average of 14 credits more than necessary; nearly half of the extra credits are due to inefficient planning and course selection [4]. Frequently, information on graduation requirements is not consolidated and students must find the information themselves [4]. Although academic advisors assist students with planning, they may lack the resources and tools to compare different scenarios in detail.

Worcester Polytechnic Institute (WPI) features a unique academic calendar that creates additional challenges. A typical undergraduate academic year at WPI includes four 7-week terms, referred to as A, B, C, and D terms. This does not include summer, and a typical undergraduate student will take 3 classes per term for a total of 12 courses per year. The four-term system significantly increases the possible offering patterns for courses: many are offered only one or two terms per year, while others are offered every term or occasionally every term except one. Some courses are even offered only once every two years. Therefore, there are 16 possible permutations of course offering patterns. WPI also offers much more flexibility than many engineering programs with regard to which courses may be used for various requirements: for many majors, relatively few courses are unequivocally required, and few electives are completely unrestricted. This creates a large number of schedules that satisfy degree requirements, but the complicated course offering patterns make many of these infeasible or impractical. Conflicts between courses that are offered only once per year can force students to overload or even delay graduation an entire year if not resolved. Therefore, early planning is critical.

This project develops a tool to assist students with selecting schedules that are efficient but flexible. It can also be utilized as an alternative degree audit tool. Although WPI does offer a degree audit tool through Workday, it does not clearly display requirement rules, nor does it always place completed courses effectively. For example, courses are sometimes counted toward free electives when they are able to fulfill a more specific requirement. It also lacks the ability to perform "what-if" analysis or provide guidance on course selection. The tool developed in this project seeks to minimize the number of additional credits to graduate, which guarantees efficient placement of already completed courses. It can also be easily extended to allow students to test different scenarios by selecting a different major or entering intended courses as "completed".

In this report, a framework for degree evaluation and planning optimization is developed. After a review of related work, a structure for requirements will be introduced and defined. Furthermore, a method to simplify the problem size is described. This system of "collections" also results in a single "solution" actually representing a set of solutions. This model is then used to create a two-stage series of integer linear programs. The first stage minimizes credits necessary, and the second stage is intended to capture a broader array of

solutions. Various computational experiments are then utilized to build a plan for completing a double major in the minimum amount of credits necessary. Additionally, the impact of specific course planning decisions is evaluated. This process also illuminates multiple potential "problem areas" where students may struggle to identify and fulfill requirements. Finally, a number of potential directions for further work are discussed.

2 Literature Review

In this section, we review prior work on similar projects and describe the factors that were considered in the design of this model.

2.1 Challenges of Constraint Complexity

The primary goal of most course planning tools is to find "ideal" solutions that satisfy a large number of complicated rules. Many models cannot accommodate requirements beyond selecting some minimum number of courses from a list. This is often insufficient for real-life graduation requirements, which may involve many other rules. Unfortunately, many degree requirements involve dependencies. Consider a program that requires three science courses but at least two must be from different areas of science. For a student who has taken two science courses and is choosing a third, whether or not it can fulfill the requirement depends on the two they have already taken. This complicates answers to questions as simple as if a student should take a physics course.

One of the most detailed and complex examples found in this review is Avi Dechter's paper "Model-Based Student Academic Planning" [5]. Dechter modeled the degree requirements for a B.A. in Child and Adolescent Development (CADV) at California State University, Northridge (CSUN). This included breadth requirements, a minimum number of upper-level courses, and constraints on how many credit-hours must be achieved before or after a course is taken. Two versions were implemented, one with constraint programming (CP) and the other with integer linear programming (ILP). Constraint programming has the advantage of not requiring linearity, but performance on the ILP was clearly superior. On the detailed test cases, the CP version failed to find an optimal solution within a "reasonable amount of time" and sometimes could not even find a feasible solution. The ILP ran much faster, but still required several minutes using CPLEX in order to generate only one optimal solution when more almost certainly exist. However, it should be noted that this paper was published in 2007 and the same problem would likely run much faster today. An improved runtime still may limit the practicality of "what-if" analysis, particularly since each run only returns a single optimal solution. Additionally, Dechter notes that the test case was still simpler than the actual requirements. For the CADV program at CSUN, some courses may sometimes be allowed to fulfill multiple requirements.

Another common issue is a lack of standardization of types of requirements. Some level of standardization is necessary for implementing an administrative user interface, which is critical for longevity. If requirements cannot be changed without editing the code itself, this greatly limits the long-term maintainability. In "Dependency Evaluation and Visualization Tool for Systems Represented by a Directed Acyclic Graph" [6], the authors describe a standardized data structure for handling complex requirements as a combination of simpler requirements. Their proposed structure allows a requirement to apply to a set of courses or a set of other requirements. Although their paper focuses primarily on graph-based visualization of prerequisites, structuring requirements such that they may run over other requirements significantly increases the complexity that can be handled.

2.2 Double Majors

Another common cause of complex requirements with dependencies is the case of double majors. Typically a student may count a course once per major, but not more than once within the same major. Identifying the courses with maximal overlap between majors is an inherent part of planning a double major, but relatively little has been published on applying optimization. One of the most direct works on double major optimization is Olabumuyi's "Application of Optimization Methods to Bachelor's Degree Planning" [7]. Olabumuyi models this with an ILP to minimize the total credit-hours needed to complete all degree requirements. Some requirement-sharing rules are implemented in this model: some courses may be also used as an upper-level course or as an elective, but not both. This is handled by adding an extra decision variable for the assignment of the course to an elective and/or upper-level requirement as applicable, the sum of which must be at most one. There do not appear to be any other constraints that limit how many times a course can be applied to a requirement. Whether or not a course satisfies a requirement is determined by a binary variable given as an input, which limits the ability to handle dependencies. Over-assignment seems to be prevented mostly implicitly, because there is little overlap between the courses that can satisfy each requirement. A notable exception is the science requirement, which requires at least one course from "Group 1", one from groups 1 or 2, and one from groups 1, 2, or 3. Presumably a course cannot count toward more than one of these requirements, or a single group 1 course would fill all three. This can still be implemented without restricting the number of assignments by considering the sum of courses from each group: at least two from groups 1-2, and at least three from groups 1-3. However, the actual implementation is unclear from the manuscript. One of the sample solutions given shows a solution with only one course from Group 1, none from Group 2, and two from Group 3 (Table 3-8, Olabumuyi). One of the Group 3 courses may be invalid because it is listed as required in addition to the group 1-3 requirements (Figure 2-1, Olabumuyi).

2.3 Student Choice and Flexibility

One of the most obvious applications of degree planning tools is helping students to identify and choose between options. Additionally, there are often multiple secondary objectives that may be considered. Some work has been done on identifying multiple solutions, such as “A decision support model for long-term course planning” [8], which introduces a “novel diversification technique” in order to generate multiple “structurally different” solutions. Other works, including Olabumuyi, allow students to assign preference weights to courses. It’s trivial for most models to restrict if specific courses are taken or not, but this may not return a feasible solution. Still, introducing secondary objectives requires determining appropriate weighting. Due to the potential size of the set of optimal solutions, it may be more useful to identify “ingredients” of optimal solutions.

One approach to comparing solutions is emphasizing schedule resiliency. Among a set of optimal schedules, which are most likely to remain optimal in the event of common changes? Where are the “pinch points” in a given degree plan? The paper “An Optimal Slack-Based Course Scheduling Algorithm for Personalised Study Plans” [9] addresses this question. The authors’ method revolves around maximizing slackness: “a slack-based algorithm simplifies which course must be scheduled first and which courses can be delayed until a later date”. The authors define “slack” as the difference between the earliest a course can be taken (while satisfying prerequisites) and the latest a course can be taken (without delaying graduation).

The primary limiting factor in their paper is “chains” of prerequisite requirements. Much of the prior work on course planning optimization has pertained to prerequisites, although not all explicitly sought to maximize “slack”. Many variants are essentially a form of critical path analysis, although the details differ. Some focus on prioritization [10], while others take a graphical approach [6]. Alternatively, "Personalizing Education With Algorithmic Course Selection" [11] introduces a calculation of "opportunity cost", based on the number of post-requisites that are "unlocked" by a course and its post-requisites.

In "TAROT: A course advising system for the future" [12], the authors describe a scheduling system that is resilience-based but does not emphasize prerequisites. Their model is also one of the few to include double majors. The authors describe the questions that students and advisors may want to answer as questions about "the present", "possible futures", "all possible futures", and "the rules". The authors note that most other works focus only on the former two, which aligns with the findings of this review. TAROT is also implemented in constraint programming, which allows complicated and nonlinear constraints to be applied more easily. Although Dechter [5] found that constraint programming was significantly slower than integer linear programming and often unable to find any solution, TAROT can solve even its most complex inquiries within about 10 minutes. Some of this difference may be due to implementation differences, but it is unclear if TAROT was tested on the very complex requirements that Dechter used. Based on the examples given, TAROT may have been tested on scenarios with very complex constraints on when a course can be taken, but simpler constraints on the graduation requirements. Dechter also scheduled many more courses; TAROT excluded general education courses. Although excluding general education can be sensible, some universities permit courses to be double-counted toward major-specific requirements and general education requirements. Maximizing this overlap could be another target for optimization, particularly for degree programs outside of STEM.

For a basic inquiry, TAROT develops a single valid schedule that satisfies input parameters or determines that one does not exist. This basic inquiry can typically be completed within 0.10 seconds if any solution exists, and within about 2.0 seconds if none exists. The authors state that a double major is "as simple as finding a schedule for the first major, then starting with this schedule [...] and filling in the requirements for the second major". However, this does not guarantee that the resulting double major schedule has minimized the number of courses needed. Different valid schedules for the first major may not have the same amount of overlap with the second major. For example, consider a double major of a math degree and computer science degree at Stetson University (which TAROT was tested on). A math degree requires one of CSCI 141 and CSCI 261, and a computer science degree requires CSCI 141. Clearly the student should take CSCI 141 for the math requirement to minimize the total courses needed. If the math degree is scheduled first with CSCI 261, it seems that TAROT may add CSCI 141 without dropping CSCI 261. This also means that results may vary depending on which major is scheduled first. The authors do reference a feature to compare the overlap between majors by computing all schedules for each one and counting the courses in common.

Generating all schedules that fulfill given criteria is another interesting feature of TAROT. The schedules can then be compared for different scenarios, which can be applied to assess schedule resilience. The authors give the example of a student who wishes to study abroad and wants to decide which semester to go abroad. Assuming that only general education courses are available while abroad, TAROT determines that there are 51,549 possible schedules if the student studies abroad in Fall 2019, but only 28,026 if the student studies abroad in Spring 2022. Therefore, the student will have much more scheduling flexibility if they go abroad in Fall 2019. Similarly, TAROT can evaluate the impact of a particular course's assignment on the rest of the schedule.

2.4 Summary

Papers on long term course planning often do not include complex requirements like Dechter, and instead assume that all requirements are in the form of selecting a certain number of courses from a set of courses. Real-life constraints can depend on one another, which can make linearization challenging and can significantly increase the number of constraints and variables needed. Additionally, the input of complex constraints must be standardized so that the tool can continue to be used as degree requirements evolve over time. A standardized input for complex rules is essential to the longevity of a degree tool. University administrators should be able to modify rules and requirements without making any modifications to the code of the tool. This is especially critical for a tool that utilizes integer linear programming, such as the one implemented in this project. Complex requirements often need to be linearized, which is challenging to automate in the absence of sufficiently standardized inputs. Any tool that can only be modified by individuals with integer linear programming experience will be severely limited in practical use.

Many other works also do not consider the double major case. In the literature reviewed, only TAROT and Olabumuyi address double major optimization. Olabumuyi does not significantly address identifying multiple solutions and TAROT identifies multiple solutions through enumeration. Although TAROT uses constraint programming in Prolog to strategically enumerate possibilities, it takes about 4 minutes to compute all paths for a double major. In this paper, we introduce a standardized structure for requirements and a system to group courses with shared attributes. By establishing the circumstances in which courses can be interchanged, this allows for the solution to assign categories of courses instead of a singular course, when possible.

3 Methodology

This section will discuss the mathematical modeling utilized in this project.

3.1 Requirements

For this project, a regular degree requirement is defined as a requirement that a specified number of credits be taken from a given set of courses. If a course is applied to a regular requirement, it may not be applied to any other regular requirements within the same major. For double majors and minors, a course may be applied up to two times as long as each applied to a different program. However, many degree requirements at WPI and other universities cannot be expressed solely as regular requirements. For example, many programs require a certain number of upper division courses, yet do not specify which upper division courses. The rules around how courses may be assigned to requirements will be referred to as super-requirements. Unlike regular requirements, no course(s) are assigned to super-requirements directly.

This project considers two types of super-requirements. The first type, called Type 1, sets a minimum or maximum number of credits from a set of courses that can be applied to a set of requirements. A common example is restrictions on upper and lower-level courses. For example, a math major at WPI may count at most one 1000-level course toward the "Transition Courses" requirement. Type 1 super-requirements may also be applied to prevent multiple mutually-exclusive courses from being taken, although this is not considered in this model. Type 2 super-requirements typically reflect "depth" requirements: a minimum number of credits must be selected from the *same* subset. A Type 2 super-requirement is considered satisfied if at least the minimum number of credits has been completed from any of the subsets. All Type 1 and Type 2 super-requirements can be implemented as linear constraints in an integer linear program; the formulation can be seen in Section 3.3. Although this project focuses on a small number of majors, this structure is intended to be applicable to many others.

Key	**Credits**	**Courses**
MA TRANSIT	12	MA 1033, MA 1971, MA 2073, MA 2631, MA 2211, MA 2251, MA 2271, MA 2273, MA 2431, MA 2631, MA 3631

Table 1: Math transition courses requirement

Key	**Direction**	**Credits**	**Selection**	**Applicable Courses**	**Applicable Reqs**
MA TRANSIT MAX	AT MOST	3	ANY OF	MA 1033, MA 1971	MA TRANSIT

Table 2: Math transition courses super-requirement

All super-requirements have a “direction” and “selection”. The direction field must be “AT LEAST” or “AT MOST” and the selection field must be “ANY OF” or “ONE OF”. The type is determined by the selection field– “ANY OF” a set of courses is a Type 1 super-requirement and “ONE OF” a set of sets of courses is a Type 2 super-requirement. For Type 2 super-requirements, the direction must be “AT LEAST”. The field “Applicable Courses” refers to this set of courses or set of subsets of courses. Finally, “Applicable Reqs” refers to the requirements that the super-requirement applies to. For example, the restriction on 1000-level math courses only applies to the courses applied to the math transition requirement– multiple 1000-level math transition courses can be used toward other requirements. An example input for the math transition requirement and 1000-level maximum super-requirement can be seen in Table 1 and Table 2.

An example of the versatility of this requirement and super-requirement model is the WPI humanities requirement. WPI requires students to choose a concentration and take 9 credits of courses from the concentration, as well as a 3-credit project related to their concentration. Concentrations may be chosen from art, foreign language, writing, history, and philosophy. Students may propose self-designed concentrations, but this requires individual approval and is thus not included in the model. WPI also requires that students take at least one course outside of their concentration (waived for foreign languages), and one "free" humanities course. The concentration requirement can be implemented with a Type 2 super-requirement, and the breadth requirement can be implemented with a series of Type 1 super-requirements that prohibit all courses from being selected from the same area. See Tables 3, 4, and 5 for the corresponding standardized input.

Key	**Credits**	**Courses**
HUA CORE	15	AR DEPT, TH DEPT, MUS DEPT, AB DEPT, CN DEPT, GN DEPT, SP DEPT, EN DEPT, WR DEPT, HI DEPT, HU DEPT, INTL DEPT, PY DEPT, RE DEPT
HUA PROJ	3	HU 3900, HU 3910

Table 3: Humanities requirements

Key	Direction	Credits	Selection	Applicable Courses	Applicable Reqs
HUA DEPTH	AT LEAST	9	ONE OF	[AR DEPT, TH DEPT, MUS DEPT] [AB DEPT, CN DEPT, GN DEPT, SP DEPT] [EN DEPT, WR DEPT] [HI DEPT, HU DEPT, INTL DEPT] [PY DEPT, RE DEPT]	HUA CORE

Table 4: Humanities depth super-requirement

Key	Direction	Credits	Selection	Applicable Courses	Applicable Reqs
HUA ART MAX	AT MOST	12	ANY OF	AR DEPT, TH DEPT, MUS DEPT	HUA CORE
HUA WR MAX	AT MOST	12	ANY OF	EN DEPT, WR DEPT	HUA CORE
HUA HI MAX	AT MOST	12	ANY OF	HI DEPT, HU DEPT, INTL DEPT	HUA CORE
HUA PY MAX	AT MOST	12	ANY OF	PY DEPT, RE DEPT	HUA CORE

Table 5: Humanities breadth super-requirement

3.2 Collections

One of the challenges of course selection optimization is handling the large number of course offerings, which leads to a large number of potential outcomes. Some courses may be grouped together, such as all courses in a given department, which helps to reduce the problem size and complexity. Combining courses into a single group is also desirable because it leads to a "single" solution actually representing a set of solutions. However, courses must be grouped carefully to ensure that the solution space is unchanged by the grouping process. For example, cross-listed courses may be unaccounted for if the grouping process is done naively. Simultaneously, courses may be enumerated unnecessarily, which increases the number of variables and the difficulty of identifying other optimal solutions.

In this project, courses are grouped into collections. All courses in a collection must be equivalent with respect to certain properties. These properties include the requirements and super-requirements that a course can fulfill, and the number of credits each course is worth. Additionally, collections must be non-overlapping; each course may be in only one collection. All courses that may be applied to any requirement must belong to a collection. These requirements are automatically satisfied if each course belongs to its own unique collection; therefore it is always possible to define a valid set of collections.

As an example, consider a simplified version of the requirements for a math major in Table 6; assume all courses are 3 credits. Notice that MA 3631 may be used as a transition course or as an upper-level course, while no other transition courses are upper-level. Therefore, MA 3631 must be in a separate collection from the rest of the transition courses. No other courses can satisfy the upper-level requirement and the transition courses requirement, so MA 3631 is the only member of its collection. Similarly, MA 1033 and 1971 are subject to the 1000 level maximum, and the rest of the transition courses are not. Therefore, they cannot be in the same collection as the rest of the transition courses, but may be in the same collection as each other. We therefore get the following collections and the requirements to which they apply, as seen in Table 7.

Requirements	**Courses**
Transition Courses	MA 1033, 1971, 2073, 2631, 2211, 2251, 2271, 2273, 2431, 2631, 3631
Real Analysis	MA 3831, 3832
Numerical Methods	MA 3257, 3457
Abstract Algebra	MA 3825, 3823
Upper Level Math	Any 3000+ MA
Super Requirements	
Transition 1000-Level Max	MA 1033, 1971

Table 6: Simplified version of math requirements

Collection of Courses	**Requirements and Super Requirements**
MA 1033, 1971	Transition Courses, Transition 1000-Level Max
MA 2073, 2631, 2211, 2251, 2271, 2273, 2431, 2631	Transition Courses
MA 3631	Transition Courses, Upper Level Math
MA 3831, 3832	Real Analysis, Upper Level Math
MA 3257, 3457	Numerical Methods, Upper Level Math
MA 3825, 3823	Abstract Algebra, Upper Level Math
All other 3000+ MA	Upper Level Math

Table 7: Collections for simplified math major

If a double major is being considered, we can apply a similar procedure to determine the collections. Consider a simplified version of the math courses that may be used for industrial engineering (Table 8). By comparing any overlap between the two, we find that MA 1033 can be applied to Calc III but not MA 1971, so these courses must be split into their own collections. Additionally, MA 2631 is the only course to fulfill the math transition requirements and the Statistics and Probability requirement. Finally, the upper-level math courses included in the OIE electives may also be counted toward a math major. The resulting set of collections is shown in Table 9.

Requirements	**Courses**
Calc III	MA 1023, 1033
Calc IV	MA 1024, 1034
Statistics	MA 2611
Statistics and Probability	MA 2612, 2621, 2631
OIE Electives	MA 3231, 3233, 3627, 4235, 4237, 4631, 4632; MIS 3720, 4084, 4720, 4741; OIE 3405, 3600, 4410, 4430, 4460
Super Requirements	
None	

Table 8: Simplified version of IE requirements

Collection of Courses	**Requirements and Super Requirements**
MA 1971	Transition Courses, Transition 1000-Level Max
MA 1033	Transition Courses, Transition 1000-Level Max, Calc III
MA 1023	Calc III
MA 1024, 1034	Calc IV
MA 2073, 2211, 2251, 2271, 2273, 2431	Transition Courses
MA 2611	Statistics
MA 2612, 2621	Statistics and Probability
MA 2631	Transition Courses, Statistics and Probability
MA 3631	Transition Courses, Upper Level Math, OIE Electives
MA 3831, 3832	Real Analysis, Upper Level Math
MA 3257, 3457	Numerical Methods, Upper Level Math
MA 3825, 3823	Abstract Algebra, Upper Level Math
MA 3231, 3233, 3627, 4235, 4237, 4631, 4632	OIE Electives, Upper Level Math
MIS 3720, 4084, 4720, 4741 OIE 3405, 3600, 4410, 4430, 4460	OIE Electives
All other 3000+ MA	Upper Level Math

Table 9: Collections for math and IE double major

3.3 Stage I Model

In this section, we introduce our mathematical model for minimizing the number of additional credits that must be taken for degree completion. This project uses a two-stage ILP; this section will describe the first and the next section describes the second. Our approach to this problem involves assigning courses to requirements, where the primary decision variable are *how many* courses from a collection to assign to each requirement. This section provides a brief overview of the model; additional details may be seen in Appendix F. The conditions that must be met for a collection to be valid are also outlines in Appendix F.

Sets	
P	Programs (majors) p being evaluated
R_p	Set of requirements r needed for program p
M	Set of all collections m
m	A collection, which is a set of one or more courses such that certain conditions are met for all courses in the collection
M_r	Set of collections m that may be applied to requirement r
R_m	Set of requirements r such that any course in m may be applied to r
U_{max}	Set of Type 1 super-requirements u with an upper bound
U_{min}	Set of Type 1 super-requirements u with a lower bound
M_u	Set of collections m to which u applies
R_u	Set of requirements to which u applies
W	Set of Type 2 super-requirements w
V_w	Set of sets v such that w is satisfied if sufficient credits have been taken from a single $v \in V_w$
M_v	Set of all m such that $m \subseteq v$
R_w	Set of requirements to which w applies

Constants	
$\alpha(m)$	Credits for any course in C_m
$\beta(r)$	Credits associated with requirement r
$\mu(u)$	Credits associated with u (upper or lower bound)
$\mu(w)$	Minimum number of credits that must be selected from the same set of courses for Type 2 super-requirement w
$T(m)$	The number of courses that have already been taken from C_m
$\gamma(m)$	The number of courses contained in C_m

Variables	
$x_{m,r}$	Integer variable representing the number of courses from C_m assigned to requirement r
y_m	Integer variable representing the total number of courses from C_m assigned to any requirement, excluding courses that have already been taken
$q_{v,w}$	Binary variable taking the value of 1 if sufficient credits have been selected from v for super-requirement w and 0, otherwise

The primary decision variable in the integer linear program is $x_{m,r}$, which represents the number of courses from C_m assigned to requirement r. Because the objective is to minimize the number of *additional* credits that need to be taken, our objective function should not include courses that have already been taken. Our objective function is therefore:

$$\min \sum_{m \in M} \alpha(m) y_m$$

First, the number of courses assigned from a collection to each program must not exceed the total number of courses in the collection. This also bounds y_m from below.

$$\sum_{r \in R_m \cap R_p} x_{m,r} \leq y_m + T(m), \quad \forall p \in P, \quad \forall m \in M$$

Next, the constraint that each requirement is satisfied:

$$\sum_{m \in M_r} \alpha(m) x_{m,r} \geq \beta(r), \quad \forall p \in P, \quad \forall r \in R_p$$

Constraints for Type 1 super-requirements:

$$\sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \alpha(m) x_{m,r} \leq \mu(u), \quad \forall u \in U_{max}$$

$$\sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \alpha(m) x_{m,r} \geq \mu(u), \quad \forall u \in U_{min}$$

For Type 2 super-requirements w, we add a binary indicator variable for whether sufficient credits are taken from each sublist, and a constraint that sufficient credits are taken from at least one:

$$q_{w,v} \leq \frac{1}{\mu(w)} \left(\sum_{m \in M_v} \sum_{r \in R_w \cap R_m} \alpha(m) x_{m,r} \right), \quad \forall w \in W, \quad \forall v \in V_w$$

$$\sum_{v \in V_w} q_{w,v} \geq 1, \quad \forall w \in W$$

The final ILP appears in (1) – (10). In summary, (1) minimizes the number of additional credits that must be completed. Constraint (2) properly bounds y_m. Constraint (3) stipulates that all requirements are met. Constraints (4) and (5) enforce Type 1 super-requirements. Constraint (6) sets the value of the variables that indicate whether a set of courses can be used as a depth for Type 2 super-requirements. Finally, constraint (7) requires that at least one depth is met for each Type 2 super-requirement.

Final ILP statement:

$$\min z = \sum_{m \in M} \alpha(m) y_m \tag{1}$$

subject to

$$\sum_{r \in R_m \cap R_p} x_{m,r} \leq y_m + T(m), \qquad \forall p \in P, \quad \forall m \in M \tag{2}$$

$$\sum_{m \in M_r} \alpha(m) x_{m,r} \geq \beta(r), \qquad \forall p \in P, \quad \forall r \in R_p \tag{3}$$

$$\sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \alpha(m) x_{m,r} \leq \mu(u), \qquad \forall u \in U_{max} \tag{4}$$

$$\sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \alpha(m) x_{m,r} \geq \mu(u), \qquad \forall u \in U_{min} \tag{5}$$

$$\sum_{m \in M_v} \sum_{r \in R_w \cap R_m} \alpha(m) x_{m,r} \geq \mu(w) q_{w,v}, \qquad \forall v \in V_w, \quad \forall w \in W \tag{6}$$

$$\sum_{v \in V_w} q_{w,v} \geq 1, \qquad \forall w \in W \tag{7}$$

$$x_{m,r} \in \{0, 1, ..., \gamma(m)\} \qquad \forall m \in M, \quad \forall r \in R \tag{8}$$

$$y_m \in \{0, 1, ..., \gamma(m) - T(m)\} \qquad \forall m \in M \tag{9}$$

$$q_{w,v} \in \{0, 1\} \qquad \forall v \in V_w, \quad \forall w \in W \tag{10}$$

3.4 Stage II Model

The first ILP minimizes total new credits, but may select a more narrow choice than is necessary. For example, a particular upper-level course may be selected when *any* upper-level course would suffice. In the second stage, we maximize *choice* without exceeding the minimum number of credits determined in the first stage. Each collection is assigned a choice weight $\Gamma(m)$ and the choice weights of all assigned courses are summed.

By default, $\Gamma(m) = \gamma(m)$. For collections that represent any course in a given department, $\gamma(m)$ excludes all courses in that department that already belong to other collections. However, the choice weights include all courses in a department *including* ones that already belong to other collections. This forces the model to preferentially assign collections that represent entire departments when possible. The same calculation is also used for collections that represent any course of a given level in a given department.

The choice weights can also be used to express preference. Although student preference is not currently taken into account, future extensions could allow courses preferred by the student to be assigned a higher choice weight. Conversely, courses that students do *not* want to take could be assigned negative scores of sufficiently large magnitude. This will still allow for non-preferred courses to be assigned if necessary. Lower choice weights can also be assigned to collections with multiple very similar courses.

The final ILP for Stage II can be seen below. The new objective function (11) maximizes the choice weights. Constraint (12) prevents the minimum number of credits needed from being exceeded. See the previous section for descriptions of constraints (2) – (10).

Constants	
z^*	The minimum number of credits necessary to complete all requirements
$\Gamma(m)$	The *choice weight* of any course in collection m

$$\max \sum_{m \in M} \Gamma(m) y_m \tag{11}$$

subject to

(2) - (10)

$$\sum_{m \in M} \alpha(m) y_m \leq z^* \tag{12}$$

4 Results and Analysis

In this section, we give an overview of the implementation of the model and some results from computational experiments.

4.1 Implementation

For this project, the model was tested on the mathematical sciences and industrial engineering majors at WPI. Data on graduation requirements was retrieved from the WPI tracking sheets and based on the requirements for the class entering in 2022-2023. All requirements and super-requirements were hand-input into a spreadsheet in a standardized format as described in Section 3.1 and can be seen in Appendix B. Physical education, free electives, and the major qualifying project (MQP) were excluded from the input and handled in post-processing. The "Additional Courses" math requirement was considered additional elective slots. Courses were also partitioned into "collections" as described in the previous section. All collections were manually formed by hand, and partial code was created that could be extended in future work to automate this process. The result can be seen in Appendix C.

Both spreadsheets were read and processed using Python. After processing, the resulting Python dictionaries were written to a JSON file. This allows for improved run-times due to much of the data processing not needing to be repeated unless the graduation rules themselves are changed. The ILP was created and solved using PuLP [13], which is open-source. A small correction factor of 0.1 was added to the left-hand side of constraint (6) to prevent negatives due to rounding. Additionally, a constraint was added to avoid overfilling the "related courses" requirement for mathematical sciences. Otherwise, the ILP formulation matched the formulation described in the methodology.

The program input required the user to select a major or set of majors (industrial engineering, mathematical sciences, or both) and to input a list of courses that have already been taken. Each course input as "taken" is compared to the courses listed under "contents" in the collections input. If a match is found, this is added to the count of courses already taken from the collection. Some courses are matched based on the department and level; no validity check is performed to check if courses entered are real courses. If no exact match is found, the course is assumed to be 3 credits.

The program was first tested with an empty list of "taken" courses, which models an incoming student with no prior credit. This was run for a math major, industrial engineering major, and for both. Table 11 shows some attributes of the ILP and Table 10 show the run times. Note that "Total" time includes setting up the model and reading the solution, therefore it is longer than the sum of the solve times for each stage. Overall credits were broken down into the following categories: general education, major-specific, MQP, and free electives. General education consists of the humanities requirement, social science requirement, physical education requirement, and IQP. "Major-specific" covers all other requirements that are input into the model, and any remaining credits are considered free electives. As previously noted, the "additional courses" for mathematics are also considered free electives.

	Math	**IE**	**Double**
Stage I Solve	0.0513	0.0459	0.0566
Stage II Solve	0.0500	0.0509	0.0588
Start to Finish	0.1154	0.1123	0.1443

Table 10: Solve time (seconds), assuming no courses have been taken yet

Problem and Stage	**Rows**	**Columns**	**Nonzero Elements**
Math, stage I	52	75	148
Math, stage II	53	75	173
IE, stage I	69	94	167
IE, stage II	70	94	203
Double, stage I	143	201	449
Double, stage II	144	201	502

Table 11: Comparison of ILPs

The distribution of credits for the three scenarios is shown in Table 12. A double major in math and industrial engineering requires a minimum of 147 credits, which is 12 more credits than the minimum for a single-major. Only 9 of these are additional coursework, while the other 3 are added to the MQP. This also shows that a maximum of 57 credits from the major-specific category may be overlapped. However, the minimum number of credits for graduation can only be achieved if 57 credits that can be double-counted are taken. Figure 1 shows the output of the initial double major run.

Credits	**Math**	**IE**	**Double**
HUA, SS, IQP, PE	36	36	36
MQP	9	9	12
Major-specific	75	81	99
Free	15	9	0
Total	135	135	147

Table 12: Distribution of credits for different programs

Industrial Engineering

INDUSTRIAL ENGINEERING CORE (27 credits)

OIE 3405 or OIE 4430
OIE 2081
BUS 3020
CS 2119, 2102, or 2103
OIE 2850
OIE 3410
OIE 3420
OIE 3460
OIE 3510

INDUSTRIAL ENGINEERING ELECTIVES (9 credits)

MA 3631
MA IE Elective
MA IE Elective

TECHNICAL ELECTIVES (9 credits)

ES
MA 3257 or 3457
CS

MATHEMATICS & BASIC SCIENCE (36 credits)
Calculus and Statistics

MA 1021
MA 1022
MA 1033
MA 1024 or 1034
MA 2051
MA 2611
MA 2631

Physics/Chemistry Sequence

CH
PH
PH

Math/Science Electives

MA 3831
MA 3832

Mathematical Sciences

CORE COURSES (12 credits)

MA 3831
MA 3832
MA 3823 or 3825
MA 3257 or 3457

TRANSITION COURSES (12 credits)
Choose from: MA 1033, MA 1971, MA 2073, MA 2211, MA 2251, MA 2271, MA 2273, MA 2431, MA 2631, MA 3631. At most one of 1033 and 1971 may be applied.

MA 1033
MA 2631
MA 3631
Math Transition Course

ADDITIONAL 3000 OR HIGHER MATH (9 credits)

MA IE Elective
MA IE Elective
3000+ MA

INTRODUCTORY AND OTHER MATH COURSES (24 credits)

MA 1021
MA 1022
MA 1024 or 1034
MA 2051
MA 2611
MA
MA
MA

RELATED COURSES (18 credits)
Two science courses (BB, CH, ES, GE, PH), two CS/DS courses, and two other courses from science, engineering, computer science or business (except FIN 1250). At most one from DS 1010, CS 2022, and CS 3043.

PH
PH
CS
CS 2119, 2102, or 2103
BUS 3020
OIE 2081

Figure 1: Course placement from first run

4.2 Identifying Critical Courses

We can now perform sensitivity analysis to determine the conditions necessary to complete a double major in only 147 credits. First, we consider "fixed" courses– if a requirement can be filled only by a single collection, then at least one course must be taken from that collection. The fixed courses for math and IE can be seen in Table 13. Note that Calculus III is not fixed because MA 1033 can be applied as a math transition course and MA 1023 cannot. Running the list of fixed courses as "taken" against each single-major can identify the double-counting potential of the fixed courses. When the fixed courses are run against a math major, 21 of the 60 credits cannot be applied to any major-specific requirements for a math major. When run against industrial engineering, 3 of the 60 credits cannot be applied. The remaining 36 credits can be applied to both. We can then compute the distribution of credits that apply to a single major or to both, which is shown in Table 14. This yields a remaining 36 credits for math and 24 credits for IE, with a total of 39 additional credits. Therefore, 21 of the remaining credits must double-count.

IE Core	**IE Math and Science**	**Math Core**
OIE 3405 or 4430	Calculus I	MA 3831
OIE 2081	Calculus II	MA 3832
BUS 3020	Calculus IV	Numerical Methods
OIE 2850	MA 2051	Abstract Algebra
OIE 3410	MA 2611	
OIE 3420	CH	
OIE 3460	PH	
OIE 3510	ES	

Table 13: Fixed courses

Consider other areas where courses may double-count. For the IE programming requirement, either OIE 3600, CS 2102, CS 2103, or CS 2119 may be used. The initial solution selects the collection of CS courses. If the program is run with OIE 3600 added to taken, the minimum total credits increases to 150. Therefore, selecting OIE 3600 prevents any optimal solution from being reached. The CS courses may all be applied to the math CS requirement, so this is a guaranteed double-count.

	Math	**IE**	**Combined**
Only IE	0	21	21
Only math	3	0	3
Both	36	36	36
Remaining	36	24	39
Total	75	81	99

Table 14: Distribution of credits for fixed courses

The set of math courses that may be applied as industrial engineering electives are another opportunity to double-count. Another way to evaluate the necessity of courses is to adjust the choice weight. If the choice weight is set to -1000 for all courses in the aforementioned set, this still results in 3 of them being assigned. Therefore, the minimum number of credits cannot be reached without at least 9 credits of these courses applied.

Another consideration is the third physics/chemistry course. This was assigned as a physics course in the initial run (Figure 1), but switching it to a chemistry course and re-running does not increase the minimum credits needed. However, adding either to fixed courses and running as a math major returns an additional elective course. Therefore, physics or chemistry may be used for the IE science requirement but neither can be double-counted in addition to the courses already double-counted.

After applying all of the changes, the new distribution table is shown at Table 15.

This table indicates that the rest of the courses counted toward IE must also double-count toward MA. Table 16 shows the requirements that must be met by the remaining 24 credits. This is now small enough that it could be enumerated if desired. Even without enumerating, we can easily see the results of simple scenarios. It can be readily seen that the remaining courses must include 21 credits of mathematics and 3 credits of CS or DS courses. Therefore, the final IE technical elective must be from math or CS/DS. If the program is run with even one technical elective course outside of these areas, the minimum credits for a double major increases to 150.

	Math	**IE**	**Combined**
Only IE	0	24	24
Only math	3	0	3
Both	48	48	48
Remaining	24	9	24
Total	75	81	99

Table 15: Updated distribution of credits

Consider if a student takes a 1000-level CS course in preparation for the required 2000-level CS courses. This course may be applied to the math CS/DS requirement, but 1000-level CS courses are excluded from IE technical electives. The IE technical elective must then be filled by a math course. This results in a much smaller subset of the technical electives that may be selected from in order to finish with the minimum number of credits.

Math	**IE**
12 credits from math transition courses	3 credits from MA 1033 or 1023
3 credits from CS or DS	3 credits from MA 2612, 2621, or 2631
6 credits of other math courses	3 credits from technical electives

Table 16: Remaining requirements

Figure 2 shows which courses must be taken and how they must be applied for an optimal double major. Asterisks indicate which courses may be moved to a different requirement and which courses must double count. Based on this analysis, any other changes will prohibit the double major from being completed in only 147 credits.

Industrial Engineering

INDUSTRIAL ENGINEERING CORE (27 credits)

OIE 3405 or OIE 4430
OIE 2081
BUS 3020
CS 2119, 2102, or 2103
OIE 2850
OIE 3410
OIE 3420
OIE 3460
OIE 3510

INDUSTRIAL ENGINEERING ELECTIVES (9 credits)

MA IE Elective
MA IE Elective
MA IE Elective

TECHNICAL ELECTIVES (9 credits)

ES
MA 3257 or 3457
TECH ELECT: MA or CS**

MATHEMATICS & BASIC SCIENCE (36 credits)
Calculus and Statistics

Calc I
Calc II
MA 1023 or MA 1033
Calc IV
MA 2051
MA 2611
MA 2612, 2621, or 2631

Physics/Chemistry Sequence

CH
PH
CH or PH

Math/Science Electives

MA 3831
MA 3832

Mathematical Sciences

CORE COURSES (12 credits)

MA 3831
MA 3832
MA 3823 or 3825
MA 3257 or 3457

TRANSITION COURSES (12 credits)
Choose from: MA 1033, MA 1971, MA 2073, MA 2211, MA 2251, MA 2271, MA 2273, MA 2431, MA 2631, MA 3631. At most one of 1033 and 1971 may be applied.

ADDITIONAL 3000 OR HIGHER MATH (9 credits)

MA IE Elective
MA IE Elective
MA IE Elective

INTRODUCTORY AND OTHER MATH COURSES (24 credits)

Calc I
Calc II
MA 1023 or 1033*
Calc IV
MA 2051
MA 2611
MA 2612, 2621, or 2631*

RELATED COURSES (18 credits)
Two science courses (BB, CH, ES, GE, PH), two CS/DS courses, and two other courses from science, engineering, computer science or business (except FIN 1250). At most one from DS 1010, CS 2022, and CS 3043.

CH
PH
CS or DS
CS 2119, 2102, or 2103
ES
OIE 2081

*May be moved to a different requirement
**Must be double-counted

Figure 2: Completed tracking sheets for major-specific requirements for a double major

4.3 Delay Cost of Switching

The next analysis conducted compares the impact of switching or adding a major at different points in time. Three sample student schedules were generated for testing the model. The first two semesters of each were based upon the WPI recommendations for first-year students (Appendix E) for students majoring in mathematics, IE, or who are undecided. For students majoring in mathematics or IE, a sample second year schedule was created. These were selected based upon course offerings and the tracking sheets and were intended to be relatively well-distributed among different subjects, although the high number of technical courses required is still reflected. The math and IE students were assumed to have taken MA 1021 and 1022 equivalents prior to WPI, as is the case for many WPI students, while the undecided student was assumed to begin with no prior college credits. Sample schedules were determined prior to determining the optimal double major path in the previous section.

	Original major		
Semesters	**IE**	**Math**	**Undecided**
1	147	147	147
2	147	147	150
3	150	147	-
4	156	153	-

Table 17: Total credits needed complete double major from various starting points

Table 17 shows the minimum number of total credits needed to complete a double major in mathematics and industrial engineering, assuming a student stays on the sample path for the given number of semesters. Students who begin as math or IE majors and follow the WPI recommendations for their first year are still able to complete the double major in the minimum number of credits. Therefore, achieving a minimum-credit double major may be achievable even if the second major is not added until the end of the first year. However, course availability may prevent being able to take all necessary double-counting courses.

An undecided freshman who follows the sample "undecided" schedule shown in Table 18 can still complete a double major in the minimum number of credits after the first semester. After the second semester, an additional 3 credits are required. This is consistent with the plan described in the previous section. The first semester complies with the plan, but the second semester contains ME 1800, which cannot be double-counted for non-elective credit.

Freshman			
A	**B**	**C**	**D**
MA 1021	MA 1022	MA 1023	MA 1024
PH 1110	PH 1120	CH 1010	ME 1800
HUA	Soc Sci	ES 1310	HUA

Table 18: Sample undecided freshman schedule

If a student begins on the sample math schedule shown in Table 19 and adds an industrial engineering major, the first three semesters align with the optimal double major plan. The course CS 2022 may be counted as a technical elective for IE and as an introductory math course for a math major, so it still fits within the plan. In the fourth semester, BB 1002 does not fit within the double major plan and adds an additional 3 credits. MA 3431 is also unable to be additionally double-counted because all "general and introductory" math slots have already been filled but it cannot fill any specific requirements for IE. Additionally, the student has already taken two social science courses and double majoring does not allow for any free electives, which means that taking ID 2050 will also increase the minimum by another 3 credits.

Although almost all IE requirements outside of the IE core can be double-counted

Freshman			
A	**B**	**C**	**D**
MA 1023	MA 1024	MA 2051	MA 1971
PH 1110	PH 1120	MA 2611	MA 2071
CS 1004	HUA	CS 2022	HUA
Sophomore			
A	**B**	**C**	**D**
MA 2631	MA 3457	MA 2073	MA 2273
MA 3231	ES 1310	BB 1002	MA 3431
ENV 1000	HUA	HUA	PSY 1400

Table 19: Sample freshman and sophomore math schedule

Freshman			
A	**B**	**C**	**D**
MA 1023	MA 1024	MA 2051	MA 2611
PH 1110	PH 1120	CH 1010	ES 1310
HUA	Soc Sci	HUA	OIE 2850
Sophomore			
A	**B**	**C**	**D**
OIE 3420	OIE 3600	OIE 3410	OIE 3405
BUS 3020	MA 2612	OIE 2081	ME 1800
MA 2071	HUA	HUA	BB 1025

Table 20: Sample freshman and sophomore IE schedule

toward a math degree, the sample IE schedule in Table 20 produced a steeper increase in the minimum possible credits to add a double major. Comparison to the double major plan in the previous section indicates that OIE 3600 will add a minimum of 3 credits to the plan, while the other courses in the first 3 semesters fit within the framework from the previous section. The fourth semester adds 6 more credits. This aligns with the forecast in the previous section that additional science or engineering courses (BB 1025, ME 1800) will increase the minimum number of credits needed.

Although these small and hand-picked examples do not necessarily represent the "average" student, the timing of increases to the total number of credits supports the validity of the double major planning sheet developed in the previous section. However, there is no guarantee that course offerings and students' particular schedules will permit completing the most efficient courses within the typical graduation time. This continues to apply even if the original optimal path is no longer possible, so a logical next step is to consider course availability.

5 Conclusion

In this section the implications, limitations, and extensions of this project will be discussed.

5.1 Discussion

This project proposed and implemented a framework for evaluating and optimizing student course selection. The framework successfully accommodated the real, complex requirements for two different majors at WPI. Run time was very fast (all runs completed within 1 second), and a single solution actually refers to a larger set of options due to the method of collections. Although a user interface remains to be implemented, the framework should allow for one to be added without making fundamental changes to the underlying code. For rules too complex to be applied with this framework, it may be worth considering if the rules can be simplified. However, the mathematical model applied in this paper is most effective as a tool for determining paths to complete double majors in the minimum number of credits. This could be extended to accommodate minors and BS/MS students, but is likely unnecessarily complex for the single major case. The degree planning problem has many facets, and different tools are better suited to different aspects.

Adding a user interface to allow students to perform their own what-if analysis and automatically generate a tracking sheet would likely be of great utility to the student body. The current tool in Workday has many limitations and does not appear to have the capability to compare various scenarios. Even simple scenarios, such as the impact of changing one's major, can be beneficial for students. For double majors, the "cost of switching" analysis (Section 4.3) indicated that relatively late additions of a second major (after the first year) may not increase the minimum credits needed by much if at all. However, even if it is theoretically possible to complete the second major without any delay compared to students who added the second major sooner, course offering schedules may make this impractical. A key component of the degree planning problem at WPI is the term offering patterns; although this was not able to be addressed in this project, some potential strategies are described in the extensions section.

Throughout this project, a few potential areas of confusion or delay for students were noted. All of the WPI first-year recommendations (Appendix E) include a social science course as one of the recommended options for each term, but only two social science courses are required and one may be ID 2050, which is mandatory for students who complete IQP abroad. Although it's understandable to encourage freshmen to take a balance of technical and non-technical courses, taking excess social science courses in the first year decreases the "slack" of free elective courses remaining. In addition, students who complete much of their humanities and social science requirements early on will have greater difficulty balancing technical and non-technical courses as upperclassmen. If WPI would like students to take more social science courses, adjusting the requirements to allow students to earn more non-elective credit for social sciences may help. Similarly, some majors do not allow GPS courses to count toward any non-elective credit. Consideration should be given to the effects of students filling multiple free elective slots as freshmen.

Another possible area of difficulty is "prerequisites". Although WPI does not have formal prerequisites, many courses would be inadvisable to take without sufficient background. The lack of required prerequisites can make it difficult for students to identify which courses need to be taken sooner. One remedy may be to identify "foundational courses" in different departments or areas of study. Students could still be permitted the freedom to enroll in any course, but with more clear guidelines for courses that should be taken sooner rather than later to prepare for upper-level coursework. Additional analysis would also likely reveal that certain "not required" courses are implicitly required if a student abides by even more of the "recommended" course guidelines. Relatedly, the IE program gives the students the choice of a scripting course or an object-oriented programming course. However, it's recommended that students take a 1000-level computer science course prior to object-oriented programming, but the IE program does not allow any credit to be given for the former except for free electives. In the example in Section 4.2, specifying a couple of standard courses such as introductory computer science or linear algebra significantly narrows the possibilities to complete the double major without exceeding the minimum required credit.

5.2 Limitations

Although care was taken to preserve the complexity of degree-planning, some factors were not considered in this project. Graduate courses were excluded, and requirements such as the residency minimum and limits on applying AP courses were also not considered.. Mutually exclusive courses were also not enforced. Although this can be handled with Type 1 super-requirements, no exportable data source on which courses are mutually exclusive was found. Course descriptions were the only identified source of this, but a substantial number of courses were listed as mutually exclusive with courses that do not appear to have been offered in the last five years. The model was also only tested on two majors, though the formulation was developed with the intention of being viable for other WPI majors. Because WPI does not have required prerequisites (only "suggested" and "recommended" courses), this was not considered in the model. This model also does not provide any recommendations regarding when to take a course.

Additionally, the process of partitioning courses into collections was performed manually and therefore may be subject to errors. Some code was written to automate this process but it was unable to be completed in time. The data can be manually altered and reloaded, but this still requires specific formatting rules to be adhered to. There is also no current way for student preference to be considered. Although the choice weights could be adjusted manually, it would likely make more sense for student weights to be multipliers on static choice weights. other than manually adjusting choice weights. The current heuristic for assigning choice weights is also unable to account for dependencies. For example, selecting three courses from a collection of seven will have a higher choice weight than selecting two from a collection of seven and one from a collection of six, although the number of permutations is higher for the latter.

The model will also typically not return the entire set of optimal solutions. The use of collections allows a "single" solution to represent a set, but in some scenarios courses will be selected randomly. Frequently, a requirement could be fulfilled by any course from a union of collections, but because collections by definition have some different properties from each other, this cannot be assumed. For example, a depth may be automatically chosen for a student if there are multiple options. However, it cannot be assumed that all depth options are legitimate. Similarly, care must be taken to avoid violating Type 1 super-requirements. The Stage II ILP was originally intended to address this, but allowing overlapping groups of collections requires substantial modeling changes. Some other possible methods of addressing this are described in further detail in the next section.

5.3 Future Work

One of the most immediate applications of this tool would be to design and implement a user interface. The requirements structure is intended to allow for a seamless connection with an administrative UI, in order for such a tool to be maintainable. The tool could create filled-in tracking sheets for students and perform what-if analysis for various scenarios, such as switching their major. Other considerations may include generating double major tracking sheets and considering student preferences. Students could also be permitted to assign their own weights for the "choice scores" described in Section 3.4. However, one barrier to implementing a tool based on this model is the need to generate the collections. This is not currently automated, and no algorithm has been finalized for ensuring that the fewest possible number of collections is used (which is similar to the set covering problem).

One clear extension to this project is to include minors and combined BS/MS programs. This introduces additional challenges because triple-counting must be prohibited, and a minimum number of courses must not be double counted at all. Additionally, some courses that are cross-listed as graduate and undergraduate courses while others are not cross-listed but are mutually exclusive. Graduate courses also have firmer prerequisites than undergraduate courses, and are typically offered on a semester basis rather than a term basis. For a BS/MS student, minimizing credits may be less important than minimizing the time to completion. Therefore, a critical part of the planning problem becomes not only which courses to take, but when.

Many other works on degree planning focus on when to take a course and prerequisites, but WPI's four-term system greatly increases the complexity of deciding when to take a course. Additionally, long-term planning that considers the timing of individual course sections is often impractical due to year-to-year variation; the terms in which a course is offered tends to be more consistent. Unfortunately, the unique four-term system at WPI can significantly increase the number of combinations of schedules, which may be prohibitive to translate into an ILP and solve in a reasonable amount of time. Although a possible ILP formulation may be seen in Appendix G, multiple areas of this project may be better suited to an algorithmic approach. In particular, the two-stage ILP may be more efficient as a combination of an algorithm and a single ILP.

This project uses a second ILP to return "broader" solutions when possible, but this will still require the solver to choose between "equivalent" solutions. One option may be to use an ILP to determine the minimum number of credits and then algorithmically determine which "swaps" are permitted to expand the solution set without losing optimality. Alternatively, an algorithm could be used to generate initial assignments and then an ILP could be used to perform sensitivity analysis, similar to the process used in Section 4.2.

One concept that was considered but unable to be implemented in this project is the "density" of course requirements. For each collection of courses, vectors representing the distribution of each course in the collection can be summed to calculate an "expected" number of courses that the student will take each term. An average of these can then be computed, weighted by the expected number of courses to be taken from each collection as determined by the ILP. This can assist in identifying terms that are most "dense" and most likely to create a bottleneck or force an overload. A similar analysis could also be done to evaluate current course offering schedules. For example, a simulation could be conducted to forecast student demand for various courses and identify areas that could most benefit from adding additional course sections. Various methods could also be developed to identify the most critical courses.

Appendices

A Reflection

In this project, we proposed a system for standardizing degree requirements and grouping courses with shared characteristics. Course planning is a complicated problem that impacts every college student. According to "Complete College America" [4], the number of degrees earned has not increased in parallel with the number of students enrolled in a degree program. WPI currently utilizes Workday for academic administration. Although Workday has a "degree progress" tool, it's unintuitive and does not permit any what-if analysis. Furthermore, the tool sometimes places courses inefficiently or outright fails to match courses to any requirement even when a match exists.

The engineering design process was applied to this problem. One of the primary objectives was creating a process that can be adapted to evolving circumstances and remain in operation without the involvement of the original creators. An otherwise impeccably designed tool may not be useful if it no longer works when information changes or if it cannot be maintained without its creator. Software development principles were considered in the implementation of this project, particularly in terms of maintaining modular code that can be adapted. A standardized spreadsheet input format was created, and code was written so that a single function could reload all of the data. The standardized input was intended to balance simplicity and readability with the complexity of actual degree requirements. The final input is structured simply enough that it should be doable to replace the spreadsheet with a user interface later. Care was taken to avoid opportunities to enter conflicting data; data that needed to be available in multiple locations was distributed during processing. After processing, the data was stored in a JSON. For this type of project, the amount of data processing is high but changes to the data are relatively infrequent. Therefore, it made sense to “front-load” the processing.

The tasks and workflow for this project differed substantially from typical, non-project-based coursework. As a long-term project, dynamically adjusting to circumstances was essential. Additionally, the original scope of the project was very broad and ultimately unrealistic for the intended timeline. Project management skills were critical to the completion of the project. One of the biggest challenges was balancing the need to be meticulous and detail-oriented with realistic goals and timeframes. Despite these challenges, this project was a valuable experience for gaining insight into the engineering design process.

B Requirements Input

Program Key	Req Key	Credits	Req Description	Courses that fill req
MATH_MAJOR	MA_INTRO	24	Intro and Elective Math Courses	["MA_DEPT"]
MATH_MAJOR	MA_TRANSIT	12	Transition Courses	["MA_1033", "MA_1971", "MA_2073", "MA_2631", "MA_2211", "MA_2251", "MA_2271", "MA_2273", "MA_2431", "MA_2631", "MA_3631"]
MATH_MAJOR	MA_REAL	6	Real Analysis	["MA_3831", "MA_3832"]
MATH_MAJOR	MA_NMTHD	3	Numerical Methods	["MA_3257", "MA_3457"]
MATH_MAJOR	MA_ABSTR	3	Abstract Algebra	["MA_3823", "MA_3825"]
MATH_MAJOR	MA_UPRLVL	9	Upper Level Math	["MA_DEPT_3000_L", "MA_DEPT_4000_L"]
MATH_MAJOR	MA_GEN_SCI	6	General Science	["PH_DEPT", "CH_DEPT", "ES_DEPT", "BB_DEPT", "GE_DEPT"]
MATH_MAJOR	MA_CS_DS	6	Computer and Data Science	["CS_DEPT", "DS_DEPT"]
MATH_MAJOR	MA_REL_CR	6	Related Courses	["BUS_DEPT", "ETR_DEPT", "MKT_DEPT", "OIE_DEPT", "MIS_DEPT", "OBC_DEPT", "FIN_DEPT", "CS_DEPT", "DS_DEPT", "PH_DEPT", "CH_DEPT", "ES_DEPT", "BB_DEPT", "GE_DEPT"]
OIE_MAJOR	OIE_CALC_I	3	Calculus Sequence	["MA_1021"]
OIE_MAJOR	OIE_CALC_II	3	Calculus Sequence	["MA_1022"]
OIE_MAJOR	OIE_CALC_III	3	Calculus Sequence	["MA_1023", "MA_1033"]
OIE_MAJOR	OIE_CALC_IV	3	Calculus Sequence	["MA_1024", "MA_1034"]
OIE_MAJOR	OIE_DIFEQ	3	Calculus Sequence	["MA_2051"]
OIE_MAJOR	OIE_STATS_I	3	Statistics	["MA_2611"]
OIE_MAJOR	OIE_STATS_II	3	Statistics	["MA_2612", "MA_2621", "MA_2631"]
OIE_MAJOR	OIE_SCI	9	Physics and Chemistry	["PH_DEPT", "CH_DEPT"]
OIE_MAJOR	OIE_MA_SCI	6	Math and Science Electives	["MA_DEPT", "CH_DEPT", "PH_DEPT", "BB_DEPT", "GE_DEPT"]
OIE_MAJOR	OIE_CORE	21	OIE Core	["OIE 2081", "BUS 3020", "OIE 2850", "OIE 3410", "OIE 3420", "OIE 3460", "OIE 3510"]
OIE_MAJOR	OIE_CS	3	OIE Core	["OIE_3600", "CS_2119", "CS_2102", "CS_2119"]
OIE_MAJOR	OIE_44_34	3	OIE Core	["OIE_3405", "OIE_4430"]
OIE_MAJOR	OIE_ELECT	9	OIE Electives	["OIE_3405", "OIE_3600", "OIE_4410", "OIE_4430", "OIE_4460", "MIS_3720", "MIS_4084", "MIS_4720", "MIS_4741", "MA_3231", "MA_3233", "MA_3627", "MA_3631", "MA_4235", "MA_4237", "MA_4631", "MA_4632"]
OIE_MAJOR	OIE_TECH_ELECT	6	Technical Electives	Any designated CE (except CE 3022), CHE, CS (except CS 1004, 1101, 1102, 3043), ECE, ES (except ES 1000, 3323), ME, OIE, RBE, as well as any IE Elective (see above
OIE_MAJOR	OIE_ES	3	Technical Electives	["ES_DEPT"]
ALL_MAJORS	HUA_CORE	15	Humanities	["AR_DEPT", "TH_DEPT", "MUS_DEPT", "AB_DEPT", "CN_DEPT", "GN_DEPT", "SP_DEPT", "EN_DEPT", "WR_DEPT", "HI_DEPT", "HU_DEPT", "INTL_DEPT", "PY_DEPT", "RE_DEPT"]
ALL_MAJORS	HUA_PROJ	3	Humanities	["HU_3900", "HU_3910"]
ALL_MAJORS	IQP	9	IQP	["IQP_A", "IQP_B", "IQP_C", "IQP_D", "IQP_OFF_CAMPUS", "IQP_ON_CAMPUS"]
ALL_MAJORS	SOC_SCI_REQ	6	Social Science	["ID_2050", "ECON_DEPT", "ENV_DEPT", "GOV_DEPT", "PSY_DEPT", "SD_DEPT", "SOC_DEPT", "SS_DEPT", "STS_DEPT", "DEV_DEPT"]

Figure 3: Requirements input sheet

Program Key	Sreq Key	Direction	Credits	Selection Type	Applicable Courses	Applicable Reqs	Sreq Type	Sublists Count	Sreq Description
MATH_MAJOR	MA_TRANSIT_MAX	AT MOST	3	ANY OF	["MA_1971", "MA_1033"]	["MA_TRANSIT"]	1b	0	At most one of MA 1033 and MA 1971 may count toward Transition Courses
MATH_MAJOR	MA_CSDS_MAX	AT MOST	3	ANY OF	["DS_1010", "CS_2022", "CS_3043"]	["MA_CS_DS", "MA_REL_CR"]	1b	0	At most one of DS 1010, CS 2022, and CS 3043 may be used for the related courses
MATH_MAJOR	MA_REL_CR_MAX	AT MOST	7.5	ANY OF	["BUS_DEPT", "ETR_DEPT", "MKT_DEPT", "OIE_DEPT", "MIS_DEPT", "OBC_DEPT", "FIN_DEPT", "CS_DEPT", "DS_DEPT", "PH_DEPT", "CH_DEPT", "ES_DEPT", "BB_DEPT", "GE_DEPT"]	["MA_REL_CR"]	1b	0	Avoid over-filling related courses
OIE_MAJOR	OIE_CH	AT LEAST	3	ANY OF	["CH_DEPT"]	["OIE_SCI"]	1a	0	At least one CH course
OIE_MAJOR	OIE_PH	AT LEAST	3	ANY OF	["PH_DEPT"]	["OIE_SCI"]	1a	0	At least one PH course
OIE_MAJOR	OIE_ELECT_RESTR	AT MOST	0	ANY OF	["CE_3022", "CS_1004", "CS_1101", "CS_1102", "CS_3043"]	["OIE_TECH_ELECT"]	1b	0	May not be used for OIE technical electives
OIE_MAJOR	OIE_ES_RESTR	AT MOST	0	ANY OF	["ES_3323"]	["OIE_TECH_ELECT", "OIE_ES"]	1b	0	May not be used for OIE technical electives
ALL_MAJORS	HUA_DEPTH	AT LEAST	9	ONE OF	[["AR_DEPT", "TH_DEPT", "MUS_DEPT"], ["AB_DEPT", "CN_DEPT", "GN_DEPT", "SP_DEPT"], ["EN_DEPT", "WR_DEPT"], ["HI_DEPT", "HU_DEPT", "INTL_DEPT"], ["PY_DEPT", "RE_DEPT"]]	["HUA_CORE"]	2	5	Depth requirement
ALL_MAJORS	HUA_ART_MAX	AT MOST	12	ANY OF	["AR_DEPT", "TH_DEPT", "MUS_DEPT"]	["HUA_CORE"]	1b	0	Breadth requirement
ALL_MAJORS	HUA_WR_MAX	AT MOST	12	ANY OF	["EN_DEPT", "WR_DEPT"]	["HUA_CORE"]	1b	0	Breadth requirement
ALL_MAJORS	HUA_HI_MAX	AT MOST	12	ANY OF	["HI_DEPT", "HU_DEPT", "INTL_DEPT"]	["HUA_CORE"]	1b	0	Breadth requirement
ALL_MAJORS	HUA_PY_MAX	AT MOST	12	ANY OF	["PY_DEPT", "RE_DEPT"]	["HUA_CORE"]	1b	0	Breadth requirement
ALL_MAJORS	ON_OFF_CAMPUS	AT LEAST	3	ANY OF	["ID_2050", "IQP_ON_CAMPUS"]	["IQP", "SOC_SCI_REQ"]	1a	0	Must take ID 2050 for off campus IQP

Figure 4: Super-requirements input sheet

C Collections Input

Collection Key	Collection Size	Choice Weight	Credits Each	Description	Contents	Req and Sreq Keys
MA_ANY	33	52	3	MA DEPT	["MA_DEPT", "BCB_4004", "DS_4635"]	["OIE_MA_SCI"]
OIE_MA_ELECT	8	8	3	OIE MATH	["MA_3231", "MA_3233", "MA_3627", "MA_3631", "MA_4235", "MA_4237", "MA_4631", "MA_4632"]	["OIE_ELECT", "OIE_MA_SCI", "OIE_TECH_ELECT"]
CALC_I	1	1	3	MA 1021	["MA_1021"]	["OIE_CALC_I"]
CALC_II	1	1	3	MA 1022	["MA_1022"]	["OIE_CALC_II"]
CALC_III	2	2	3	CALC III	["MA_1023", "MA_1033"]	["OIE_CALC_III"]
CALC_IV	2	2	3	CALC IV	["MA_1024", "MA_1034"]	["OIE_CALC_IV"]
DIF_EQ	1	1	3	MA 2051	["MA_2051"]	["OIE_DIFEQ"]
STATS_1	1	1	3	MA 2611	["MA_2611"]	["OIE_STATS_I"]
PROB_STATS	3	3	3	MA 2612, 2621, 2631	["MA_2612", "MA_2621", "MA_2631"]	["OIE_STATS_II", "OIE_MA_SCI"]
CS_ANY	31	41	3	CS DEPT	["CS_DEPT", "BCB_4002", "BCB_4003"]	["OIE_TECH_ELECT"]
MA_CS	3	3	3	CS 4032, 4033, 2022	["CS_4032", "CS_4033", "CS_2022"]	["OIE_MA_SCI", "OIE_TECH_ELECT"]
OBJ_ORT	3	3	3	CS 2119, 2102, 2103	["CS_2119", "CS_2102", "CS_2103"]	["OIE_CS"]
OIE_ENG_RESTR	5	5	3	CE 3022, CS 1004, CS 1101, CS 1102, CS 3043	["CE_3022", "CS_1004", "CS_1101", "CS_1102", "CS_3043"]	["OIE_ELECT_RESTR"]
OIE_CORE	7	7	3	OIE Core	["OIE_2081", "BUS_3020", "OIE_2850", "OIE_3410", "OIE_3420", "OIE_3460", "OIE_3510"]	["OIE_CORE"]
OIE_3600	1	1	3	OIE 3600	["OIE_3600"]	["OIE_CS", "OIE_ELECT", "OIE_TECH_ELECT"]
OIE_OTHER	2	2	3	OIE 4410, 4460	["OIE_4410", "OIE_4460"]	["OIE_ELECT", "OIE_TECH_ELECT"]
OIE_CHOICE	2	2	3	OIE 3405, 4430	["OIE_3405", "OIE_4430"]	["OIE_44_34", "OIE_ELECT", "OIE_TECH_ELECT"]
OIE_MIS_ELECT	4	4	3	OIE MIS	["MIS_3720", "MIS_4084", "MIS_4720", "MIS_4741"]	["OIE_ELECT", "OIE_TECH_ELECT"]
ES_ANY	13	15	3	ES DEPT	["ES_DEPT"]	["OIE_ES", "OIE_TECH_ELECT"]
OIE_ES_RESTR	1	1	3	ES 3323	["ES_3323"]	["OIE_ES_RESTR", "OIE_ES", "OIE_TECH_ELECT"]
OIE_ENG_ELECT	100	100	3	Engineering Elective	["CE_DEPT", "CHE_DEPT", "ECE_DEPT", "ME_DEPT", "RBE_DEPT", "BME_4504", "BME_4606", "BME_4814", "CS_4801"]	["OIE_TECH_ELECT"]
BB_4190	1	1	3	BB 4190	["BB_4190"]	["OIE_SCI", "OIE_MA_SCI", "OIE_CH"]
CH_ANY	27	27	3	CH DEPT	["CH_DEPT"]	["OIE_SCI", "OIE_MA_SCI", "OIE_CH"]
ECE_2112	1	1	3	ECE 2112	["ECE_2112"]	["OIE_SCI", "OIE_MA_SCI", "OIE_PH", "OIE_TECH_ELECT"]
PH_ANY	28	28	3	PH DEPT	["PH_DEPT", "ECE_2112", "AE_2550"]	["OIE_SCI", "OIE_MA_SCI", "OIE_PH"]
OTHR_SCI	43	43	3	BB DEPT, GE DEPT	["BB_DEPT", "GE_DEPT"]	["OIE_MA_SCI"]
IQP_ON_CM	1	25	9	IQP on campus	["IQP_ON_CAMPUS"]	["IQP", "ON_OFF_CAMPUS"]
IQP_OFF_CM	4	100	9	IQP off campus	["IQP_OFF_CAMPUS", "IQP_A", "IQP_B", "IQP_C", "IQP_D"]	["IQP"]
ID_2050	1	1	3	ID 2050	["ID_2050"]	["SOC_SCI_REQ", "ON_OFF_CAMPUS"]
ART_CON	20	20	3	Humanities	["AR_DEPT", "TH_DEPT", "MUS_DEPT", "IMGD_341X"]	["HUA_CORE", "HUA_DEPTH_SL_0", "HUA_ART_MAX"]
FLG_CON	20	20	3	Humanities	["CN_DEPT", "GN_DEPT", "SP_DEPT", "AB_DEPT", "ID_3525", "ID_3527", "ID_3529", "ID_3530", "ID_3531"]	["HUA_CORE", "HUA_DEPTH_SL_4"]
WR_CON	20	20	3	Humanities	["EN_DEPT", "WR_DEPT", "IMGD_2400", "IMGD_3400", "EN_241X"]	["HUA_CORE", "HUA_DEPTH_SL_1", "HUA_WR_MAX"]
HI_CON	20	20	3	Humanities	["HI_DEPT", "HU_DEPT", "INTL_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_2", "HUA_HI_MAX"]
PY_CON	20	20	3	Humanities	["PY_DEPT", "RE_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_3", "HUA_PY_MAX"]
ALL_SOC_SCI	60	60	3	Social Science	["ECON_DEPT", "ENV_DEPT", "GOV_DEPT", "PSY_DEPT", "SD_DEPT", "SOC_DEPT", "SS_DEPT", "STS_DEPT", "DEV_DEPT", "MU_2501", "ENV_2500"]	["SOC_SCI_REQ"]
HUA_PROJ	2	1	3	HU 3900 or 3910	["HU_3900", "HU_3910"]	["HUA_PROJ"]

Figure 5: Collections input for IE single major

Collection Key	Collection Size	Choice Weight	Credits Each	Description	Contents	Req and Sreq Keys
MA_3000P	22	26	3	MA 3000+	["MA_DEPT_3000_L", "MA_DEPT_4000_L", "BCB_4004"]	["MA_UPRLVL", "MA_INTRO"]
MA_3631	1	1	3	MA 3631	["MA_3631"]	["MA_TRANSIT", "MA_UPRLVL", "MA_INTRO"]
REAL_ALYS	2	2	3	MA 3831, 3832	["MA_3831", "MA_3832"]	["MA_REAL", "MA_UPRLVL", "MA_INTRO"]
ABS_ALG	2	2	3	MA 3823, 3825	["MA_3823", "MA_3825"]	["MA_ABSTR", "MA_UPRLVL", "MA_INTRO"]
MA_ANY	16	50	3	MA DEPT	["MA_DEPT"]	["MA_INTRO"]
TRANSIT_R	7	7	3	Transition Courses	["MA_2073", "MA_2631", "MA_2211", "MA_2251", "MA_2271", "MA_2273", "MA_2431"]	["MA_TRANSIT", "MA_INTRO"]
TRANSIT_LL	2	2	3	MA 1033, 1971	["MA_1033", "MA_1971"]	["MA_TRANSIT", "MA_TRANSIT_MAX", "MA_INTRO"]
CS_DS_DEPTS	40	45	3	CS DEPT, DS DEPT	["CS_DEPT", "DS_DEPT", "BCB_4002", "BCB_4003"]	["MA_CS_DS", "MA_REL_CR", "MA_REL_CR_MAX"]
CS_3043	1	1	3	CS 3043	["CS_3043"]	["MA_CS_DS", "MA_REL_CR", "MA_CSDS_MAX", "MA_REL_CR_MAX"]
CS_2022	1	1	3	CS 2022	["CS_2022"]	["MA_CS_DS", "MA_REL_CR", "MA_CSDS_MAX", "MA_INTRO", "MA_REL_CR_MAX"]
NUM_METHD	2	2	3	Numerical Methods	["CS_4032", "CS_4033", "MA_3257", "MA_3457"]	["MA_NMTHD", "MA_UPRLVL", "MA_INTRO"]
DS_4635	1	1	3	DS 4635	["DS_4635"]	["MA_UPRLVL", "MA_INTRO", "MA_CS_DS", "MA_REL_CR", "MA_REL_CR_MAX"]
REG_SCI_ES	100	100	3	Science (PH, CH, ES, BB, GE)	["PH_DEPT", "CH_DEPT", "ES_DEPT", "BB_DEPT", "GE_DEPT", "ECE_2112", "AE_2550", "BB_4190"]	["MA_GEN_SCI", "MA_REL_CR", "MA_REL_CR_MAX"]
BUS_SCH_ELECT	100	100	3	Business School (any)	["BUS_DEPT", "ETR_DEPT", "MKT_DEPT", "OIE_DEPT", "MIS_DEPT", "OBC_DEPT", "FIN_DEPT"]	["MA_REL_CR", "MA_REL_CR_MAX"]
ECON_2910	1	1	3	ECON 2910	["ECON_2910"]	["MA_REL_CR" , "SOC_SCI_REQ", "MA_REL_CR_MAX"]
IQP_ON_CM	1	25	9	IQP on campus	["IQP_ON_CAMPUS"]	["IQP", "ON_OFF_CAMPUS"]
IQP_OFF_CM	4	100	9	IQP off campus	["IQP_OFF_CAMPUS", "IQP_A", "IQP_B", "IQP_C", "IQP_D"]	["IQP"]
ID_2050	1	1	3	ID 2050	["ID_2050"]	["SOC_SCI_REQ", "ON_OFF_CAMPUS"]
ART_CON	20	20	3	Humanities	["AR_DEPT", "TH_DEPT", "MUS_DEPT", "IMGD_341X"]	["HUA_CORE", "HUA_DEPTH_SL_0", "HUA_ART_MAX"]
FLG_CON	20	20	3	Humanities	["CN_DEPT", "GN_DEPT", "SP_DEPT", "AB_DEPT", "ID_3525", "ID_3527", "ID_3529", "ID_3530", "ID_3531"]	["HUA_CORE", "HUA_DEPTH_SL_4"]
WR_CON	20	20	3	Humanities	["EN_DEPT", "WR_DEPT", "IMGD_2400", "IMGD_3400", "EN_241X"]	["HUA_CORE", "HUA_DEPTH_SL_1", "HUA_WR_MAX"]
HI_CON	20	20	3	Humanities	["HI_DEPT", "HU_DEPT", "INTL_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_2", "HUA_HI_MAX"]
PY_CON	20	20	3	Humanities	["PY_DEPT", "RE_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_3", "HUA_PY_MAX"]
ALL_SOC_SCI	60	60	3	Social Science	["ECON_DEPT", "ENV_DEPT", "GOV_DEPT", "PSY_DEPT", "SD_DEPT", "SOC_DEPT", "SS_DEPT", "STS_DEPT", "DEV_DEPT", "MU_2501", "ENV_2500"]	["SOC_SCI_REQ"]
HUA_PROJ	2	1	3	HU 3900 or 3910	["HU_3900", "HU_3910"]	["HUA_PROJ"]

Figure 6: Collections input for math single major

Collection Key	Collection Size	Choice Weight	Credits Each	Description	Contents	Req and Sreq Keys
MA_3000P	14	27	3	MA 3000+	["MA_DEPT_3000_L", "MA_DEPT_4000_L", "BCB_4004"]	["MA_INTRO", "MA_UPRLVL", "OIE_MA_SCI"]
OIE_MA_ELECT	7	7	3	OIE MATH	["MA_3231", "MA_3233", "MA_3627", "MA_4235", "MA_4237", "MA_4631", "MA_4632"]	["MA_INTRO", "MA_UPRLVL", "OIE_ELECT", "OIE_MA_SCI", "OIE_TECH_ELECT"]
MA_3631	1	1	3	MA 3631	["MA_3631"]	["MA_INTRO", "MA_TRANSIT", "MA_UPRLVL", "OIE_MA_SCI", "OIE_ELECT", "OIE_TECH_ELECT"]
REAL_ALYS	2	2	3	MA 3831, 3832	["MA_3831", "MA_3832"]	["MA_INTRO", "MA_TRANSIT", "MA_REAL", "MA_UPRLVL", "OIE_MA_SCI"]
ABS_ALG	2	2	3	MA 3823, 3825	["MA_3823", "MA_3825"]	["MA_ABSTR", "MA_INTRO", "MA_TRANSIT", "MA_UPRLVL", "OIE_MA_SCI"]
MA_ANY	7	52	3	MA DEPT	["MA_DEPT"]	["MA_INTRO", "OIE_MA_SCI"]
CALC_I	1	1	3	MA 1021	["MA_1021"]	["MA_INTRO", "OIE_CALC_I"]
CALC_II	1	1	3	MA 1022	["MA_1022"]	["MA_INTRO", "OIE_CALC_II"]
CALC_III	1	1	3	MA 1023	["MA_1023"]	["MA_INTRO", "OIE_CALC_III"]
CALC_IV	2	2	3	Calc IV	["MA_1024", "MA_1034"]	["MA_INTRO", "OIE_CALC_IV"]
MA_1033	1	1	3	MA 1033	["MA_1033"]	["MA_INTRO", "MA_TRANSIT", "MA_TRANSIT_MAX", "OIE_CALC_III"]
TRANSIT_LL	1	1	3	MA 1971	["MA_1971"]	["MA_INTRO", "MA_TRANSIT", "MA_TRANSIT_MAX", "OIE_MA_SCI"]
DIF_EQ	1	1	3	MA 2051	["MA_2051"]	["MA_INTRO", "OIE_DIFEQ"]
STATS_1	1	1	3	MA 2611	["MA_2611"]	["MA_INTRO", "OIE_STATS_I"]
PROB_STATS	2	2	3	MA 2612, 2621	["MA_2612", "MA_2621"]	["MA_INTRO", "OIE_STATS_II", "OIE_MA_SCI"]
MA_2631	1	1	3	MA 2631	["MA_2631"]	["MA_INTRO", "MA_TRANSIT", "OIE_STATS_II", "OIE_MA_SCI"]
TRANSIT_R	6	6	3	Math Transition Course	["MA_2073", "MA_2211", "MA_2251", "MA_2271", "MA_2273", "MA_2431"]	["MA_INTRO", "MA_TRANSIT", "OIE_MA_SCI"]
CS_DEPT	31	41	3	CS DEPT	["CS_DEPT", "BCB_4002", "BCB_4003"]	["MA_CS_DS", "MA_REL_CR", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
CS_21XX	3	3	3	CS 2119, 2102, 2103	["CS_2119", "CS_2102", "CS_2103"]	["MA_CS_DS", "MA_REL_CR", "OIE_TECH_ELECT", "OIE_CS", "MA_REL_CR_MAX"]
CS_3043	1	1	3	CS 3043	["CS_3043"]	["MA_CS_DS", "MA_CSDS_MAX", "MA_REL_CR", "OIE_TECH_ELECT", "OIE_ELECT_RESTR", "MA_REL_CR_MAX"]
CS_2022	1	1	3	CS 2022	["CS_2022"]	["MA_INTRO", "MA_CS_DS", "MA_CSDS_MAX", "MA_REL_CR", "OIE_MA_SCI", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
NUM_METHD	2	2	3	Numerical Methods	["CS_4032", "CS_4033", "MA_3257", "MA_3457"]	["MA_INTRO", "MA_NMTHD", "MA_UPRLVL", "OIE_MA_SCI", "OIE_TECH_ELECT"]
OIE_ENG_RESTR	3	3	3	CS 1004, 1101, 1102	["CS_1004", "CS_1101", "CS_1102"]	["MA_CS_DS", "MA_REL_CR", "OIE_ELECT_RESTR", "MA_REL_CR_MAX"]
DS_DEPT	3	4	3	DS DEPT	["DS_DEPT"]	["MA_CS_DS", "MA_REL_CR", "MA_REL_CR_MAX"]
DS_4635	1	1	3	DS 4635	["DS_4635", "MA_4635"]	["MA_INTRO", "MA_CS_DS", "MA_REL_CR", "MA_UPRLVL", "OIE_MA_SCI", "MA_REL_CR_MAX"]
OIE_CORE	7	7	3	OIE Core	["OIE_2081", "BUS_3020", "OIE_2850", "OIE_3410", "OIE_3420", "OIE_3460", "OIE_3510"]	["MA_REL_CR", "OIE_CORE", "MA_REL_CR_MAX"]
OIE_3600	1	1	3	OIE 3600	["OIE_3600"]	["MA_REL_CR", "OIE_CS", "OIE_ELECT", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
OIE_OTHER	2	2	3	OIE 4410, 4460	["OIE_4410", "OIE_4460"]	["MA_REL_CR", "OIE_ELECT", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
OIE_CHOICE	2	2	3	OIE 3405, 4430	["OIE_3405", "OIE_4430"]	["MA_REL_CR", "OIE_44_34", "OIE_ELECT", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
OTHR_SCI	30	30	3	BB DEPT, GE DEPT	["BB_DEPT", "GE_DEPT"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_MA_SCI", "MA_REL_CR_MAX"]
BB_4190	1	1	3	BB 4190	["BB_4190"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_CH", "OIE_MA_SCI", "OIE_SCI", "MA_REL_CR_MAX"]
REG_SCI_ES	1	1	3	AE 2550	["AE_2550"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_PH", "OIE_MA_SCI", "OIE_SCI", "MA_REL_CR_MAX"]
PH_ANY	30	30	3	PH DEPT	["PH_DEPT"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_MA_SCI", "OIE_SCI", "OIE_PH", "MA_REL_CR_MAX"]
CH_ANY	30	30	3	CH DEPT	["CH_DEPT"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_SCI", "OIE_MA_SCI", "OIE_CH", "MA_REL_CR_MAX"]
CE_3022	1	1	3	CE 3022	["CE_3022"]	["OIE_ELECT_RESTR"]
OIE_ES_RESTR	1	1	3	ES 3323	["ES_3323"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_ES_RESTR", "OIE_ES", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
ES_ANY	30	30	3	ES DEPT	["ES_DEPT"]	["MA_GEN_SCI", "MA_REL_CR", "OIE_ES", "OIE_TECH_ELECT", "MA_REL_CR_MAX"]
ECE_2112	1	1	3	ECE 2112	["ECE_2112"]	["OIE_SCI", "OIE_MA_SCI", "OIE_PH", "OIE_TECH_ELECT"]
BCB_4004	1	1	3	BCB 4004	["BCB_4004"]	["OIE_MA_SCI"]
BUS_SCH_ELECT	50	50	3	Business School, any	["BUS_DEPT", "ETR_DEPT", "MKT_DEPT", "MIS_DEPT", "OBC_DEPT", "FIN_DEPT"]	["MA_REL_CR", "MA_REL_CR_MAX"]
ECON_2910	1	1	3	ECON 2910	["ECON_2910"]	["MA_REL_CR", "SOC_SCI_REQ", "MA_REL_CR_MAX"]
OIE_MIS_ELECT	4	4	3	OIE MIS	["MIS_3720", "MIS_4084", "MIS_4720", "MIS_4741"]	["OIE_ELECT", "OIE_TECH_ELECT"]
OIE_ENG_ELECT	50	50	3	Engineering elective	["CE_DEPT", "CHE_DEPT", "ECE_DEPT", "ME_DEPT", "RBE_DEPT", "BME_4504", "BME_4606", "BME_4814"]	["OIE_TECH_ELECT"]
IQP_ON_CM	1	25	9	IQP on campus	["IQP_ON_CAMPUS"]	["IQP", "ON_OFF_CAMPUS"]
IQP_OFF_CM	4	100	9	IQP off campus	["IQP_OFF_CAMPUS", "IQP_A", "IQP_B", "IQP_C", "IQP_D"]	["IQP"]
ID_2050	1	1	3	ID 2050	["ID_2050"]	["SOC_SCI_REQ", "ON_OFF_CAMPUS"]
ART_CON	20	20	3	Humanities	["AR_DEPT", "TH_DEPT", "MUS_DEPT", "IMGD_341X"]	["HUA_CORE", "HUA_DEPTH_SL_0", "HUA_ART_MAX"]
WR_CON	20	20	3	Humanities	["EN_DEPT", "WR_DEPT", "IMGD_2400", "IMGD_3400", "EN_241X"]	["HUA_CORE", "HUA_DEPTH_SL_1", "HUA_WR_MAX"]
HI_CON	20	20	3	Humanities	["HI_DEPT", "HU_DEPT", "INTL_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_2", "HUA_HI_MAX"]
PY_CON	20	20	3	Humanities	["PY_DEPT", "RE_DEPT"]	["HUA_CORE", "HUA_DEPTH_SL_3", "HUA_PY_MAX"]
FLG_CON	20	20	3	Humanities	["CN_DEPT", "GN_DEPT", "SP_DEPT", "AB_DEPT", "ID_3525", "ID_3527", "ID_3529", "ID_3530", "ID_3531"]	["HUA_CORE", "HUA_DEPTH_SL_4"]
ALL_SOC_SCI	60	60	3	Social Science	["ECON_DEPT", "ENV_DEPT", "GOV_DEPT", "PSY_DEPT", "SD_DEPT", "SOC_DEPT", "SS_DEPT", "STS_DEPT", "DEV_DEPT", "MU_2501", "ENV_2500"]	["SOC_SCI_REQ"]
HUA_PROJ	2	1	3	HU 3900 or 3910	["HU_3900", "HU_3910"]	["HUA_PROJ"]

Figure 7: Collections input for math-IE double major

D Tracking Sheets

MATHEMATICAL SCIENCES MAJOR
Program Tracking Sheet
Effective for students entering AY 2022-2023

Name:	Class Year:
Advisor:	2nd Major:

NOTES: Minimum total academic credit = 15 units
Residency Req.: Min. of 8 units must be completed at WPI

HUMANITIES AND ARTS (6/3 units)
All 5 HUA courses must be completed before beginning the Inquiry Seminar or Practicum.

Depth Component
Students must complete at least three thematically-related courses prior to the culminating Inquiry Seminar or Practicum in the same thematic area. At least one of the three courses should be at the 2000-level or above.

	Course	Term	Grade	Units
1				1/3
2				1/3
3				1/3
4	HU 3900 or HU 3910			1/3

Breadth Component
Students must take at least one course outside the grouping in which they complete their depth component. To identify breadth, courses are grouped in the following manner.
i. art/art history, drama/theatre, and music (AR, EN/TH, MU);
ii. foreign languages (AB, CN, EN, GN, SP);
iii. literature and writing rhetoric (EN, WR, RH);
iv. history and international studies (HI, HU, INTL);
v. philosophy and religion (PY, RE).
Exception: May take all six courses in a foreign language

5				1/3

Humanities Elective

6				1/3

PHYSICAL EDUCATION (4 PE classes = 1/3 unit)

7				1/12
				1/12
				1/12
				1/12

SOCIAL SCIENCE (2/3 unit) ECON, ENV, GOV, PSY, SD, SOC, SS, STS, DEV, and ID2050

8				1/3
9				1/3

THE INTERACTIVE QUALIFYING PROJECT (1 unit)

10				1/3
11				1/3
12				1/3

FREE ELECTIVES (1 unit)

13				1/3
14				1/3
15				1/3
16				1/3
17				1/3
18				1/3
19				

MATHEMATICS (22/3 units)
May not include both MA 2631 and MA 2621.
May not include both MA 2071 and MA 2072.
At least 7/3 units must consist of MA courses at the 3000 level or above.

TRANSITION COURSES (4/3 units)
Must include at least four of the following: MA 1033 or MA 1971 (one only), MA 2073, MA 2211, MA 2251, MA 2271*, MA 2273*, MA 2431, MA 2631, MA 3631, or their equivalents (MA 2621 cannot be counted as a transition course)

CORE COURSES (4/3 units)

20	MA 3831			1/3
21	MA 3832			1/3
22	MA 3257 or MA 3457			1/3
23	MA 3823* or MA 3825*			1/3

ADDITIONAL COURSES AT 3000 LEVEL OR HIGHER (1 unit)

24				1/3
25				1/3
26				1/3

INTRODUCTORY AND OTHER MATH COURSES (8/3 units)

27				1/3
28				1/3
29				1/3
30				1/3
31				1/3
32				1/3
33				1/3
34				1/3

MAJOR QUALIFYING PROJECT (3/3 unit)

35				1/3
36				1/3
37				1/3

MATHEMATICAL PROGRAM - RELATED COURSES (2 units)
Courses from other departments that are related to the student's mathematical program. At least 2/3 units of science must be included. At least 2/3 unit in computer science or data science must be included; the remaining courses are to be selected from science, engineering, computer science or business (except FIN 1250). Science courses may be chosen from the following disciplines: BB, CH, ES, GE, PH. CS/DS courses may include only one of DS 1010, CS 2022 and CS 3043.

38	SCI			1/3
39	SCI			1/3
40	CS or DS			1/3
41	CS or DS			1/3
42				1/3
43				1/3

ADDITIONAL COURSES (2/3 units)
Additional courses or independent studies (except AS, MS, PE courses, and other degree requirements) from any area.

44				1/3
45				1/3

*Category II classes. Only offered every other year.

Figure 8: Mathematical Sciences Tracking Sheet

INDUSTRIAL ENGINEERING MAJOR
Program Tracking Sheet
Effective for students entering AY 2023 (2022-2023)

Name:	Class Year: 2026	Date tracking sheet completed:
Student ID #:	Advisor:	2nd Major/Minor:

NOTES: Minimum academic credit = 15 units
Residency Req.: Min. of 8 units must be completed at WPI.
No course should appear more than once on this sheet, with the exception of PE courses.

HUMANITIES AND ARTS REQUIREMENT (6/3 units)

--Breadth Component
Students must take **at least one** course outside the grouping in which the depth component falls:
a. Art/art history, drama/theatre, music (AR, EN/TH, MU)
b. Foreign languages (AB, CN, EN, GN, SP)
c. Literature, writing, rhetoric (EN, WR, RH)
d. History, international studies (HI, HU, INTL)
e. Philosophy, religion (PY, RE)

Course	Term	Grade	Units
1.			1/3
2.			1/3

--Depth Component
Students must complete **at least three** thematically-related courses prior to the culminating Inquiry Seminar or Practicum in the same thematic area. At least one must be at the 2000 level or above. Exception: May take all six courses in German or Spanish.

3.			1/3
4.			1/3
5.			1/3
6. HU 3900/3910 (**taken last**)			1/3

PHYSICAL EDUCATION (4 PE classes = 1/3 unit)

7.1			1/12
7.2			1/12
7.3			1/12
7.4			1/12

MATHEMATICS & BASIC SCIENCE (12/3 units)

Calculus Sequence

8. MA 1021			1/3
9. MA 1022			1/3
10. MA 1023			1/3
11. MA 1024			1/3
12. MA 2051			1/3

Statistics Sequence

13. MA 2611			1/3
14. MA 2612 or 2621			1/3

Physics/Chemistry Sequence

15. CH			1/3
16. PH			1/3
17. CH or PH			1/3

Math/Science Electives (one of each is recommended)
Recommended in Math: MA 2071, probability & stats., numerical analysis
Science: BB, CH, GE, PH

18.			1/3
19.			1/3

SOCIAL SCIENCES (2/3 units)
Two from: DEV, ECON, ENV, GOV, ID 2050/SS 2050, PSY, SD, SOC, SS, STS

20. ID 2050/SS 2050	AY '25		1/3
21.			1/3

THE INTERACTIVE QUALIFYING PROJECT (3/3 units)

22.	AY '25		1/3
23.	AY '25		1/3
24.	AY '25		1/3

INDUSTRIAL ENGINEERING CORE (9/3 units)

25. OIE 3405 or OIE 4430			1/3
26. OIE 2081			1/3
27. BUS 3020			1/3
28. OIE3600 or CS2119 or 2102/2103**			1/3
29. OIE 2850			1/3
30. OIE 3410			1/3
31. OIE 3420			1/3
32. OIE 3460			1/3
33. OIE 3510			1/3

**Take CS 1004 prior to CS 2119. Take CS 1101/1102 prior to CS 2102/2103.

INDUSTRIAL ENGINEERING ELECTIVES-Operations Research (3/3 units)
Choose three: OIE 3405*, 3600*, 4410, 4430*, 4460; MIS 3720, 4084, 4720, 4741; MA 3231, 3233, 3627, 3631, 4235, 4237, 4631, 4632.
*Only if not taken in IE Core.

34.			1/3
35.			1/3
36.			1/3

TECHNICAL ELECTIVES (3/3 units)
Any designated CE (*except* CE 3022), CHE, CS (*except* CS 1004, 1101, 1102, 3043), ECE, ES (*except* ES 1000, 3323), ME, OIE, RBE, as well as any IE Elective (see above). Suggested courses include:
CS 2011, CS 4032/MA 3257, ECE 2010, ES 1310, ES 2001, ES 2800, ES 3001, ME 1800, ME 2820. *GPS course credits do not qualify.*

37. **One ES course required**			1/3
38.			1/3
39.			1/3

THE MAJOR QUALIFYING PROJECT (3/3 units)

40.	A '25		1/3
41.	B '25		1/3
42.	C '26		1/3

FREE ELECTIVES (3/3 units)

43.			1/3
44.			1/3
45.			1/3

NOTES

Revised 05/18/2021

Figure 9: Industrial Engineering Tracking Sheet

E WPI First-Year Recommended Courses

A	**B**
MA Science/Econ/Additional MA GPS/HUA/CS	MA Science/Econ/Additional MA GPS/HUA/CS
C	**D**
MA Science/Econ/Additional MA GPS/HUA/CS	MA Science/Econ/Additional MA HUA/CS

Table 21: First-year recommendations for math majors

A	**B**
MA CH or PH GPS/HUA/SS	MA CH or PH GPS/HUA/SS
C	**D**
MA CH/PH GPS/HUA/SS	MA Engineering Science HUA/SS/OIE 2850

Table 22: First-year recommendations for IE students

A	**B**
CH 1010 or PH 1110/1111 MA GPS/HUA/SS	CH 1020 or PH 1120/1121 MA GPS/HUA/SS
C	**D**
CH 1010 or PH 1110/1111 MA GPS/HUA/SS/Intro Course	CH 1020 or PH 1120/1121 MA GPS/HUA/SS/Intro Course

Table 23: First-year recommendations for undecided students

Source: [14]

F Model Derivation

In this section, we will begin with the assumption that we wish to assign individual courses to requirements. The necessary attributes of collections will be demonstrated and defined.

Sets	
P	Programs (majors) p being evaluated
R_p	Set of requirements needed for program p
C	Set of all courses c
C_r	Set of all courses that can be applied to requirement r
R_c	Set of all requirements that course c can be applied to
U_{max}	Set of Type 1 super-requirements u with an upper bound
U_{min}	Set of Type 1 super-requirements u with a lower bound
C_u	Set of courses that u applies to
R_u	Set of requirements to which u applies
W	Set of Type 2 super-requirements w
$v \in V_w$	Each set of courses that may be selected for Type 2 super-requirement w
R_w	Set of requirements to which w applies

Constants	
$\alpha(c)$	Credits for course c
$\beta(r)$	Credits associated with requirement r
$\mu(u)$	Credits associated with u (upper or lower bound)
$\mu(w)$	Credits associated with w (lower bound)

Variables	
$x_{c,r}$	Binary variable taking the value of 1 if course c is assigned to requirement r; 0 otherwise
y_c	Binary variable taking the value of 1 if course c is assigned at all and has not already been taken; 0 otherwise
$q_{v,w}$	Binary variable taking the value of 1 if sufficient credits have been selected from v for super-requirement w; 0 otherwise

Consider the following constraints that ensure all requirements are met:

$$\sum_{c \in C_r} \alpha(c) x_{c,r} \geq \beta(r), \qquad \forall p \in P, \forall r \in R_p \tag{1}$$

$$\sum_{c \in C_u} \sum_{r \in R_c \cap R_u} \alpha(c) x_{c,r} \leq \mu(u), \qquad \forall u \in U_{max} \tag{2}$$

$$\sum_{c \in C_u} \sum_{r \in R_c \cap R_u} \alpha(c) x_{c,r} \geq \mu(u), \qquad \forall u \in U_{min} \tag{3}$$

$$\sum_{c \in v} \sum_{r \in R_w \cap R_c} \alpha(c) x_{c,r} \geq \mu(w) q_{w,v}, \qquad \forall v \in V_w, \forall w \in W \tag{4}$$

The primary goal of the model transformation is to reduce the number of variables through the following substitution, where m is a *collection* of courses.

$$x_{m,r} = \sum_{c \in m} x_{c,r}$$

In order for this to be possible, the following conditions must hold for all collections $m \in M$ and for any $c_j \in m$:

$$\alpha(c_j) = \alpha(c_k) \qquad \forall c_k \in m$$

$$c_j \in C_r \iff m \subseteq C_r \qquad \forall r \in R_p, \quad \forall p \in P$$

$$c_j \in C_u \iff m \subseteq C_u \qquad \forall u \in U$$

$$c_j \in C_v \iff m \subseteq C_v \qquad \forall v \in V_w, \quad \forall w \in W$$

In simple terms, the set of requirements and super-requirements that a course can apply to must be the same for all courses in a collection. Additionally, the number of credits must be the same for all courses in the same collection. We define the following new sets, where c_j is an arbitrary element of m:

$$
\begin{aligned}
R_m &:= \quad R_{c_j} \\
\alpha(m) &:= \quad \alpha(c_j) \\
M_r &:= \quad \{m|m \subseteq C_r\} \\
M_u &:= \quad \{m|m \subseteq C_u\} \\
M_v &:= \quad \{m|m \subseteq C_v\}
\end{aligned}
$$

We can now begin to transform expressions. First, the LHS of constraint (1):

$$\sum_{c \in C_r} \alpha(c) x_{c,r} \iff \sum_{m \in M_r} \sum_{c \in m} \alpha(c) x_{c,r} \iff \sum_{m \in M_r} \alpha(m) x_{m,r}$$

Next, the LHS of constraints (2) and (3):

$$\sum_{c \in C_u} \sum_{r \in R_c \cap R_u} \alpha(c) x_{c,r} \iff \sum_{m \in M_u} \sum_{c \in m} \sum_{r \in R_m \cap R_u} \alpha(c) x_{c,r}$$

$$\iff \sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \sum_{c \in m} \alpha(c) x_{c,r} \iff \sum_{m \in M_u} \sum_{r \in R_m \cap R_u} \alpha(m) x_{m,r}$$

Then the LHS of constraint (4):

$$\sum_{c \in C_v} \sum_{r \in R_c \cap R_w} \alpha(c) x_{c,r} \iff \sum_{m \in M_v} \sum_{c \in m} \sum_{r \in R_m \cap R_w} \alpha(c) x_{c,r}$$

$$\iff \sum_{m \in M_v} \sum_{r \in R_m \cap R_w} \sum_{c \in m} \alpha(c) x_{c,r} \iff \sum_{m \in M_v} \sum_{r \in R_m \cap R_w} \alpha(m) x_{m,r}$$

G Alternative Formulation

This is one possible formulation for determining when to take courses to minimize time to graduation. This version does not use collections and represents individual course assignments. A simpler definition of requirements is used and courses are not assigned to requirements.

Sets	
R	Set of requirements r that must all be completed
C	Set of considered courses
C_r	Set of courses that can be applied to requirement r
C_{SEM}	Set of semester-length courses
T	Set of all term indices, consecutive integers 1 through n
T_c	Terms when course c may be taken
T_c^*	Terms when course c may be completed (must be even index for semester courses)
C_t	Courses that may be taken during term t
$q \in Q$	Set of prerequisite labels
C_q	Courses that satisfy prerequisite q
Q_c	All prerequisites that must be satisfied before course c
U	Set of all sets of mutually exclusive courses
u	Set of course indices for a set of mutually exclusive courses

Constants	
$\alpha(c)$	Credits for c, in terms of undergrad credits
k_c	Semester constant: 2 if semester and 1 if term
$\beta(r)$	Minimum credits needed to satisfy requirement r
λ	Maximum credits a student is willing to take per term (default 10.5)
n	Number of terms considered in planning horizon

Variables	
$x_{c,t}$	Binary variable; assumes the value of 1 if course c is assigned to term t and 0 otherwise, $\forall c \in C, \forall t \in T_c$
y_c	Binary variable; assumes the value of 1 if course c is taken at all and 0 otherwise, $\forall c \in C$
q_t	Binary variable; assumes the value of 1 if prerequisite q has been met by term t and 0 otherwise, $\forall t \in T, \forall q \in Q$
Z	The latest term where any courses are taken

First, we add a constraint to set the value of y_c. This also forces that a course is taken in only one term, or two for semester courses.

$$\sum_{t \in T_c} x_{c,t} = k_c y_c, \quad \forall c \in C$$

Next, constraint that all requirements are met:

$$\sum_{c \in C_r} \alpha(c) y_c \geq \beta(r), \quad \forall r \in R$$

And the maximum number of credits that may be taken per term:

$$\sum_{c \in C_t} \frac{\alpha(c) x_{c,t}}{k_c} \leq \lambda, \forall t \in T$$

Preventing mutually exclusive courses from both being taken:

$$\sum_{c \in U_u} y_c \leq 1, \quad \forall u \in U$$

Setting a constraint to define the indicator variable for whether q is satisfied by term t:

$$q_t \leq \sum_{c \in C_q} \sum_{j \in T_c^* \cap [1,t-1]} x_{c,j}, \quad t \in [2, n]$$

$$q_1 = 0$$

Next, that a course may only be taken if the prerequisite is satisfied:

$$x_{c,t} \leq q_t, \quad \forall q \in Q_c, \forall c \in C, \forall t \in T_c$$

For a term-based course, the number of the term that course c is assigned to, if any, is given by:

$$\sum_{t \in T_c} t x_{c,t}$$

For a semester-based course, the larger of the two consecutive terms that course c is taken can be given by:

$$\frac{1}{2} + \frac{1}{2} \sum_{t \in T_c} t x_{c,t}$$

Therefore, if we define Z as a nonnegative integer and set the following constraint, Z may be no less than the latest term that contains any courses:

$$Z \geq \frac{1}{k_c} \left(\sum_{t \in T_c} t x_{c,t} \right) \quad \forall c \in C$$

Thus, minimizing Z will identify the minimum number of terms that must be completed for the BS/MS. One final constraint is needed to ensure that semester courses are assigned to consecutive terms, which must include an even numbered term and the one before it, or not at all.

$$\left(\sum_{t \in T_c^*} t x_{c,t}\right) - \left(\sum_{t \in T_c^*} (t-1) x_{c,t-1}\right) = y_c, \quad \forall c \in C_{SEM}$$

If $y_c = 1$, then this constraint can only be satisfied if a course is assigned to an even-numbered term and the term before it. If $y_c = 0$, then the definition of y_c means that the LHS is also 0. constraint is satisfied by definition of y_c

The final ILP is then:

$$\min Z$$

subject to:

$$\sum_{t \in T_c} x_{c,t} = k_c y_c, \qquad \forall c \in C$$

$$\sum_{c \in C_r} \alpha(c) y_c \geq \beta(r), \qquad \forall r \in R$$

$$\sum_{c \in C_t} \frac{\alpha(c) x_{c,t}}{k_c} \leq \lambda, \qquad \forall t \in T$$

$$\sum_{c \in U_u} y_c \leq 1, \qquad \forall u \in U$$

$$\sum_{c \in C_q} \sum_{j \in T_c^* \cap [1,t-1]} x_{c,j} \geq q_t, \qquad t \in [2,n], \forall q \in Q$$

$$q_1 = 0, \qquad \forall q \in Q$$

$$x_{c,t} \leq q_t, \qquad \forall q \in Q_c, \forall c \in C, \forall t \in T_c$$

$$\frac{1}{k_c}\left(\sum_{t \in T_c} t x_{c,t}\right) \leq Z, \qquad \forall c \in C$$

$$\left(\sum_{t \in T_c^*} t x_{c,t}\right) - \left(\sum_{t \in T_c^*} (t-1) x_{c,t-1}\right) = y_c, \qquad \forall c \in C_{SEM}$$

References

[1] National Center for Education Statistics, "Undergraduate retention and graduation rates," 2022. Accessed on March 2, 2023.

[2] S. K. Hayes, "Student employment and the economic cost of delayed college graduation," *Journal of Business & Leadership*, vol. 6, pp. 129 – 140, 2010.

[3] D. Witteveen and P. Attewell, "Delayed time-to-degree and post-college earnings," *Research in Higher Education*, vol. 62, pp. 230 – 257, 2019.

[4] Complete College America, "New rules: Policies to strengthen and scale the game changes," 2016. Accessed on March 2, 2023.

[5] A. Dechter, "Model-based student academic planning," *The International Journal of Applied Management and Technology*, vol. 5, no. 1, pp. 87 – 104, 2007.

[6] S. Samaranayake and A. Gunawardena, "Dependency evaluation and visualization tool for systems represented by a directed acyclic graph," *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020.

[7] T. Olabumuyi, *Application of Optimization Methods to Bachelor's Degree Planning.* PhD thesis, Southern Methodist University, 2015.

[8] A. Mohamed, "A decision support model for long-term course planning," *Decision Support Systems*, vol. 74, 2015.

[9] A. L. Richards and R.-S. Tsay, "An optimal slack-based course scheduling algorithm for personalised study plans," in *Proceedings of the 2020 9th International Conference on Educational and Information Technology*, ICEIT 2020, (New York, NY, USA), p. 1–7, Association for Computing Machinery, 2020.

[10] L. A. Le Blanc, C. T. Rucks, and W. S. Murray, "A decision support system for prescriptive academic advising," *Advanced Topics in End User Computing*, vol. 1, pp. 263 – 284, 2002.

[11] T. Morrow, "Personalizing education with algorithmic course selection," Master's thesis, Missouri University of Science and Technology, 2017.

[12] J. Ekroth and R. Anderson, "TAROT: A course advising system for the future," *Journal of Computing Sciences in Colleges*, vol. 34, no. 3, pp. 108 – 116, 2018.

[13] Computational Infrastructure for Operations Research, "Optimization with PuLP." Accessed on March 8, 2023.

[14] Worcester Polytechnic Institute, "The first-year experience," 2023. Accessed on March 2, 2023.