A Process for Systematically Collecting Plan of Study Data for Curricular Analytics

Abstract

This theory paper describes challenges and opportunities with analyzing engineering curricula using the *Curricular Analytics* framework by offering a data collection framework for systematically collecting engineering plans of study at scale. Introduced by Heileman and colleagues in 2017, the Curricular Analytics framework enables researchers and practitioners to quantify the interconnectedness of their prerequisite structures to unveil gatekeeper courses and forecast the impact of curricular policies or changes using network-analytic metrics. These metrics can be calculated using all available data; all one needs to do is transform a plan of study into a list of courses, prerequisites, and corequisites. However, larger projects that examine institutional, disciplinary, and temporal differences will likely face difficulties when wrangling with the details of diverse organizational contexts. This paper outlines the data entry processes developed by drawing from the research group's Microsoft Teams communications for a National Science Foundation sponsored project to explore trends in curricular complexity across institutions in the Multi-Institution Database for Engineering Longitudinal Development (MIDFIELD) for five disciplines of engineering across ten years. We anticipate these suggestions will streamline data collection for similar large scale projects in the future that employ Curricular Analytics as their analytical approach.

Background

Curricular Analytics involves the quantification of a curriculum to correlate the associated metrics with proxies for student success, often degree completion rates. To accomplish the quantification, we represent a plan of study outlining the coursework requirements a student must complete in order to earn a degree as a network. In the network, courses are represented as vertices (or nodes) and the prerequisite relationships among them are given by directed edges (arrows). This data type allows us to calculate a suite of metrics drawn from the pool of techniques developed in other fields like social network analysis that can help us capture "complexity" in a meaningful way. First appearing in its most recognizable form in work by Wigdahl as the idea of "curricular efficiency" [1], Heileman et al. [2] provide a thorough treatment of the possible quantities that form Curricular Analytics. At its core, Curricular Analytics outlines a framework for conceptualizing and measuring *curricular complexity.*

Curricular complexity is divided into two components: instructional complexity and structural complexity [2]. Instructional complexity attempts to capture the latent factors of the curriculum, such as course difficulty and instructional quality, but is currently only proxied by the pass rate of a course. Explicit advancements in expanding the idea of instructional complexity are almost non-existent with the exception of Waller, who reframed course difficulty using the concept of grade anomalies and found it to be more robust of a metric than individual course DFW rates in his study of organizational factors' impact on student success [3]. Structural complexity has been explored to a greater extent, likely because of the ability to readily access public data and construct simulations with relative ease.

Among the metrics proposed for structural complexity, two have become central to how complexity is calculated. These are visualized in Figure 1.

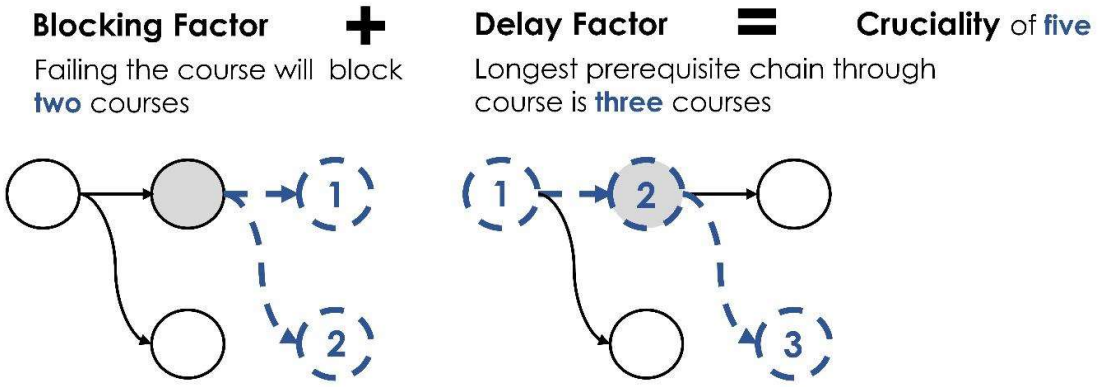

Figure 1. Calculating course cruciality using the blocking and delay factor of the gray course

The first metric is the blocking factor, which is found by counting the number of courses inaccessible to a student if the course is failed. The second metric is the delay factor, the length of the longest prerequisite chain flowing through the course. Adding these two values together yields the cruciality, a local measurement of how entangled a course is in the prerequisite structures of the plan of study and how essential it is to complete. The cruciality can be used to find potential bottlenecks, or gatekeeper courses, in the curriculum [4], [5]. For a global measurement, simply add all the crucialities together – this value is called the *structural complexity* of the curriculum. These global measurements can be used to compare curriculum within and across different strata, such as majors within other colleges [6] or disciplines of engineering [7]. Other efforts have also emerged to apply to connect curricular complexity with topic-level dependencies between courses [8], incorporate a probabilistic student flow approach [9], and extend the framework to be sensitive to transfer student issues [10], [11].

Although the data requirements are minimal, there are distinct challenges to employing this framework across disciplines and institutions – especially if longitudinal analyses are planned. For example, curricula are not always completely defined, leaving space for students to select electives with varying degrees of flexibility. Without specifying these electives, the curriculum's complexity may be underestimated. Moreover, there are deeper data entry considerations; prerequisites can often be a complicated series of ANDs and ORs, with language like "at least" or "X of the following." These configurations do not lead to obvious network representations. Finally, finding accurate plan of study information, even for recent years, can be challenging. Prerequisite and course information can be inaccurate or may not have been appropriately maintained by the appropriate institutional office. As Curricular Analytics is applied more broadly, it is valuable to reflect on current practices and look ahead for how this framework can be expanded to capture more nuanced curricular representations.

Research Aim

The purpose of this paper is to overview the different obstacles encountered during the data collection process and the standardized procedures and conventions we developed for a project employing Curricular Analytics. We outline these procedures not only for transparency, but to assist other researchers and practitioners who want to use the Curricular Analytics framework at scale. Given the lack of formal guidance on how to handle plan of study data for broader projects, we contend this work can become a resource to fill the current gap in standard practices for proper data entry to analyze curricula.

Drawing Insights from a Broader Longitudinal Project on Curricular Complexity

This work is derived from a larger project focused on quantifying plans of study for five engineering disciplines (Civil Engineering, Electrical Engineering, Mechanical Engineering, Chemical Engineering, and Industrial Engineering) to compare the complexity of such programs across the United States. The sampling frame, in this case, was the Multi-Institution Database for Engineering Longitudinal Development (MIDFIELD) [12]. Data collection for the larger project was completed in January 2023. In Fall 2022, five undergraduate research assistants in the College of Engineering and Applied Science and a PhD student in engineering education were tasked with entering data from course catalogs over the course of a decade for thirteen schools in MIDFIELD. To facilitate team communication, we used a channel in Microsoft Teams. The students were encouraged to talk with one another and ask questions if they ran into issues during data collection.

One of the major outputs of the NSF project is an R package that will allow users to interact with the dataset we created, but also use more customized functions to explore different dimensions of curricular complexity. We chose to write our package in R because of the existing packages for analyzing data from MIDFIELD, namely midfieldr [13] and midfielddata [14]. The first of which contains ready-to-go functions for properly processing data at the student level, and the second package is a stratified random sample from MIDFIELD for users to practice on and explore. The data produced from this project will be made available in a similar format. We anticipate the output synergizing with the broader goal of expanding access to and participation in MIDFIELD's development [15].

Data Collection

Although originally intended for project communication alone and standardization to ensure process reliability [16], we found our Microsoft Teams chats to be a valuable source of practical issues that other teams might encounter when conducting a similar project. The data source for this work was the set of chats from our team's communication regarding data entry within the platform, Microsoft Teams. Due to licensing policies with the university-sponsored Microsoft Teams account, it was not feasible to export all the chats in the channel through a request to the university IT department. Instead, after expanding all chats to ensure longer messages and pictures were not cut off, the Chrome plugin GoFullPage created a pdf of the channel using the web-browser version of Teams. One small inconvenience of this method is that the resulting pdf was not searchable, meaning that all relevant questions needed to be labeled by hand and referenced by number using a spreadsheet. After filtering out non-data entry questions (e.g., questions about submitting timesheets or meeting time updates), we were left with 88 questions.

Analysis

We employed descriptive coding to split the questions into categories where similar or repeated questions could be grouped [17]. To create more generalized questions that removed specific institutional context, questions within each category were processed using the constant comparative method [18] to consolidate similar inquiries into one unique question. The research assistants, both undergraduate and graduate, who entered the data in Fall 2022 were involved in the synthesis of these generalized questions, and their perspectives shared in weekly meetings are a form of peer debriefing [19]. Among these questions, we selected the questions related to general data entry and created a flowchart to summarize a process for entering curricular data in similar projects.

A General Process for Collecting and Extracting Plan of Study Data

Through our data collection processes in Fall 2022, including the discussions and questions posed during data entry, we have assembled the following considerations for collecting plans of study for use with the Curricular Analytics framework. These considerations include methodological issues and general process inconsistencies that can emerge.

Longitudinal Studies of Curricular Complexity

The main study this work is derived is the first longitudinal study of which we are aware that uses Curricular Analytics. During our process of collecting data across multiple years, especially trending into the early 2010s and 2000s, we found that data entry can be fraught with challenges. In particular, several questions were posed regarding the availability of plans of study and catalogs:

- [INSTITUTION]'s archived catalogs only go back till 2014. If there are more, how do I find them? Or did it join late or something like that?
- There doesn't appear to be a catalog for 13-14. Did anyone find it?
- It seems like Industrial Engineering doesn't exist as a concentration prior to 2017-2018.

To go back in time and retrieve plans of study dating earlier than those immediately available through the current institutional websites, we used the Wayback Machine, an archive of webpages from across the Internet. Especially for searches farther back in time, we recommend starting from the most general page that can be retrieved, which was often the department homepage or institutional homepage, and navigating to the appropriate pages for the registrar and major. Sometimes retrieving older plans of study is not possible because the Wayback Machine does not archive all webpages. Unfortunately, there is nothing that can be done without contacting the registrar at the institution to request more information. Keeping record of the plans of study and the catalog from the years under study and noting where discrepancies occur is highly recommended.

Analytical Considerations for Curricular Complexity Studies. Preliminary results from our project suggest that complexity is not necessarily stable from year to year. Small curricular changes can have a noticeable impact on structural complexity. Consider one of the MIDFIELD institutions in Figure 2. Here we can see comparisons of the form "X is more structurally complex than Y" does not hold uniformly from year to year. In this case, mechanical engineering eventually switched places with chemical engineering from 2019-2020 to 2020-2021. Thus, we suggest other researchers consider the longitudinal behavior of complexity in their studies. In particular, looking back a few years to check how stable the prerequisite structures are would be useful as a quality and validity check.

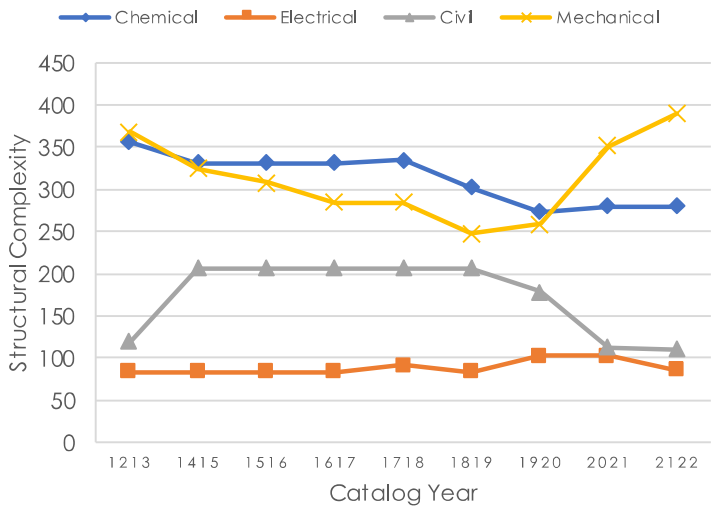

Figure 2. Trajectories of four engineering disicplines at one MIDFIELD institution demonstrating that structural complexity is not necessarily stable year to year; note that the catalog for 2013-2014 was unable to be found

Non-Trivial Prerequisites

Although many courses will have simple prerequisite structures that do not require special treatment, there are more complicated relationships that can be decomposed such that they can be represented in the network. These questions were frequent in our discussions, a selection of which are below:

- This has nested "Or" options is there a good way to show this in the excel files?;"
- How should we show prerequisites like this given that there are "Or" + "And"? Should I just list them all and separate them with a comma?
- Also, I just saw that there was a prereq where you could take two classes from a "this or that". How should I work that into the prereq chain? [Note: "this or that" refers to a configuration where a student can pick one of two courses to satisfy a requirement.]
- MATH 160 is the first math course with MATH 161 and 261 coming in later, and some courses have prereqs like this: MATH 160 or MATH 161 or MATH 261, In this case, what do we write down as prereqs?
- For this it says 6 of the 7 courses, what should I do about this?
- This senior design class has a bunch of prereqs, and some of the are in a this or that in the POS. For example, GEEN 1400 is a prereq for the senior design class but you can choose between GEEN 1400 and ASEN 14000 or ECEN 1400 in semester 1. Should I keep the this or that and put the elective course in the prereq section or just choose GEEN 1400 as the only class you take?

Curricular Analytics implicitly assumes that courses have a simple "AND" configuration to describe prerequisites; for example, MATH 101 AND MATH 102 are required for ENGR 105. However, for certain courses, these configurations can involve "ORs" that are also nested within "ANDs," such as (MATH 101 OR MATH 101H) AND PHYS 101. These are often combined with grade cut-offs, such as "Min Grade C-." Last, some courses have prerequisite structures that offer a list of courses and require the students have passed a subset of them using language like "at least" or "X of the following." None of these situations are addressable in Curricular Analytics as currently defined, except that it is acknowledged on the FAQ page of the tool's website that OR relationships tend to be irrelevant [20].

To help explore whether these are indeed factors needed in curricular complexity, we introduce the following notation in Table 1 to enter the data as they are shown in course catalogs. As the framework evolves, we foresee the estimates of structural complexity to be an average of different scenarios. Our next steps involve incorporating the functionality in our R package to calculate the structural complexity for different configurations.

Table 1. Notation for prerequisite structures

Prerequisite Type	Notation	Example
AND	When a list of courses must be completed, each one is separated by a comma.	MATH 101, MATH 102
OR	When the option between prerequisites is given, we use the + symbol.	MATH 101 + MATH 102
Both Logical Connectives	Combine the comma and + notation to represent the prerequisite structure. Use parentheses to group.	(MATH 101 + MATH 102), MATH 103
Subset of Course	Use the keyword FROM and list the courses followed by the number of courses to consider in brackets.	FROM(MATH 101, MATH 102, MATH 103)[2]
Minimum Grade	Use the keyword MINGRADE, name the course and insert the minimum grade in brackets	MINGRADE(MATH 101)[C-]

We are currently experimenting with this notation, so to transform back to the original version – using purely AND connectives, we would convert all "+" to ",", remove all parentheses, and extract the courses inside the FROM and MINGRADE notation. Next, any courses that do not appear would be removed from the list of requisites. This approach would match how a researcher would conventionally enter the data. Starter R script is provided below.

```
#This function takes in the columns for the pre- and co-requisites
#and removes the supplementary information in our suggested notation
#such that it will work with the conventional functions for
#Curricular Analytics
ConvertToOriginalNotation <- function(RequisiteStructure) {
  #Replace all ORs (+) with ANDs (,)
  RequisiteStructure <- gsub("\\+", ",",RequisiteStructure)
  #Remove the parentheses
  RequisiteStructure <- gsub("[()]", "", RequisiteStructure)
  #Begins removing the notation for MINGRADE and FROM.
  RequisiteStructure <- gsub("FROM|MINGRADE", "", RequisiteStructure)
  #Remove notation for MINGRADE and FROM.
  RequisiteStructure <- gsub("\\[.*?\\]", "", RequisiteStructure)
  #Return the courses with the original notation.
  return(RequisiteStructure)
}

#Apply the function to each of the courses in the prerequisite and
#corequisite columns.
ConvertedRequisites <- sapply(RequisiteStructures, ConvertToOriginalNotation)
```

Incorporating Minimum Grades. The one prerequisite type that is not immediately clear how to best incorporate yet is the minimum grade condition, but it is certainly a structural barrier that students must overcome. A weight could be placed on such courses to account for the additional barrier of increased expectations before moving into the next course(s) and should be explored further. The minimum grade is most closely associated with the blocking factor because it is an additional barrier to progress into the next set of courses, so it would be sensible to incorporate it into the metric. Alternatively, because the minimum grade is a condition of the *edge* (i.e., the prerequisite) and not the vertex, a different metric might be necessary. A common term for these minimum grades is the "C-Wall," referring to imposing a minimum C grade for introductory coursework to enter second year offerings or graduate altogether. For example, one potential metric (which we can call "C-Wall" density, CWD) could be weighted sum of overall edges in the network (i.e., $e \in E$) where we assign some value to each edge based on a function Q, as depicted in the following equation:

$$CWD = \sum_{e \in E} Q(e)$$

For example, Q could take on the value zero for any prerequisite without a minimum grade requirement. For a D- minimum, Q can assign a value of 1/3 and increment up in steps of 1/3 (to account for the +/- system of grades).

Non-Calculus-Ready Pathways. Building from our previous observations regarding non-trivial prerequisite structures, these most often occur with courses in the first year. Such prerequisite structures result from attempts to account for diverse offerings of mathematics courses to reflect differences in students' academic preparation. Some requirements span beyond coursework, including minimum SAT and math place scores – which we observed in our sample. When constructing the plan of study, the standard operating procedure was to only consider courses in the plan of study. However, to consider student's diverse pathways into the curriculum, researchers may consider incorporating the mathematics prerequisite structures that form the pre-calculus block of offerings. Data from Pirkey and Santiago [21] show retention statistics for students in engineering at West Virginia University, revealing that 29% percent of students were not Calculus-ready and an overwhelming majority of students in their second semester were enrolled in Trigonometry. The prevalence of non-Calculus-ready students raises additional questions about how we represent the situation in Curricular Analytics, especially considering being Calculus-ready in the first semester is a significant predictor of completing an engineering degree [22]. There are opportunities to analyze not only the expected pathway for the average first-time-in-college student but also for students who do not have the necessary math background to start in the intended mathematics course.

The notation in Table 1 can help facilitate such analyses if that is the researcher's intent. Otherwise, the notation can be converted to the standard form and analyzed normally. Only courses that appear in the list would be incorporated into the construction of the prerequisite structure in our R package.

Illogical Arrangements and Inconsistencies. Occasionally, there will be instances where a configuration in the plan of study results in an illogical arrangement of prerequisites or corequisites, as evidenced in the following sample of questions:

- ***Corequisites in different terms:*** There is a course that is calling for a coreq but the course it wants as a coreq is listed under a previous year. Should I place it as a prereq instead?
- ***Prerequisites not in the plan of study because of course changes:*** For 14-15, there is as course ME 221 that prereqs ME 210: Statics of Mechanic. But this course isn't the POS,

instead there is EM 214: Statics. For 16-17, EM 215 was the prereq for ME 221. Do I assume this as a typo?

- ***Missing courses/credits:*** For the same POS, the total sum of the credits is given as 128, but is actually 129. And, there are 3 credits missing.

Though not mentioned in our chats, upon inspecting plans of study for logical arrangements, an error that occurred was accidently listing a corequistite as a prerequisite – which would create a theoretically impossible structure. Other issues can appear as well, so it is critical to note where potential errors have occurred. We contend these inconsistencies emerged because of our longitudinal scope, introducing situations where courses become defunct and are replaced with a new offering or sets of offerings, but this change does not become fully represented in the plan of study or catalog. During data entry or post-processing, it is suggested that researchers check for illogical arrangements, particularly with prerequisites in the same term and corequisites in different terms, against the original plan of study. If it is indeed the intended arrangement, add a note on the same line in the spreadsheet. If not, make the necessary adjustment.

Hidden Laboratory Coursework

Engineering curricula include laboratory-focused courses to allow students to apply principles from lecture to physical systems, and their realization in the curriculum can vary. We observed that plans of study would either have a specific laboratory course with a corequisite relationship to the lecture course or describe an integrated laboratory experience that is mentioned by course name and number or no distinct course at all. Questions would emerge related to how these could be handled, for example:

- How should I show labs for a class that has the labs "baked in" with the class?
- If there is a recitation class, should I include it even if it doesn't show as a coreq and it has no credits?
- Some classes have recitation some don't, I wasn't sure if we wanted to call that out. The way labs for EE classes are listed is the same as above. No course name or number and no mention in the POS other than what comes up when you go to look at the prereqs and coreqs.

We found it helpful to untangle these hidden laboratory requirements by creating a course with zero credit hours with a corequisite relationship with the lecture course to unveil the additional burden on students that they create. In our view, bundling the lecture and laboratory can underestimate the true structural complexity because credit hours are not factored into the calculations at all. This convention we adopted placed all courses with labs on the same footing.

Assumptions about Corequisite Relationships

The measures in Curricular Analytics rely on the idea of a *directed acyclic graph,* a type of network where none of the directed edges form a cycle. In other words, by following the edges from vertex to vertex, you will never be able to revisit a vertex. These curricular graphs are directed acyclic graphs by design because prerequisite and corequisite relationships naturally build on one another. It would not make sure for a course taken later in the curriculum to be a prerequisite for a course in a previous term; thus, there is no reasonable potential for a cycle to form. However, there is one exception - mutual corequisite relationships. For example, we have seen cases where a course and its lab list one another as corequisites. The courses then form a cycle, which makes calculating the delay factor impossible. But the redundancy is not needed, so it is more sensible to only have one edge connecting the two.

Deciding the direction of the corequisite relationship is a nontrivial decision, which came about when the validity of the plan of study data was being evaluated. To illustrate why, consider the simplest possible configuration where the distinction is evident: Course A is a corequisite with Course B and Course B is a prerequisite for Course C. If we have the corequisite relationship defined as Course B points to Course A, we can calculate the structural complexity to be 8. Next, by simply changing the direction of the corequisite relationship, we see the structural complexity rise to 12 – an increase of 50%! This is troubling because we did not change anything fundamental about the prerequisite relationships; the interpretation is the exact same – Courses A and B are intended to be taken together. However, the direction of the relationship has a significant impact on the course crucialities and the overall structural complexity as a result.

Why does this occur? The issue lies in the calculation of both the blocking factor and delay factor. When we change the direction of the corequisite relationship, we create a different path for requisite relationships to flow. As a result, the delay factors change for all three courses. In the first configuration, Course A does not have any paths out of the vertex but is connected to Course B, so its delay factor is 2; it also does not block any courses, so its blocking factor is 0. However, once we switch the direction of the edge, suddenly Course A becomes part of the prerequisite chain formed by Course B and Course C. This changes all the delay factors for A, B, and C to 3 and increases Course A's blocking factor to 2.

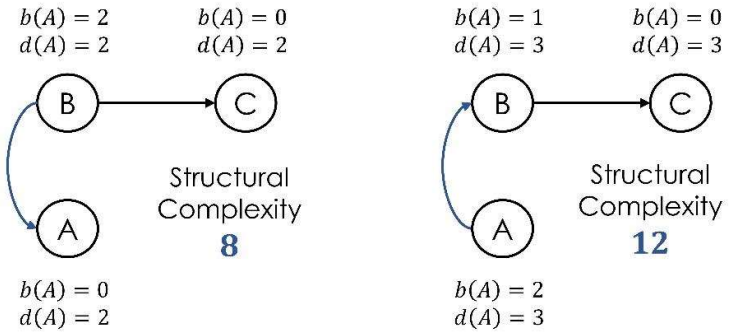

Figure 2. Illustrating the impact of changing the direction of a corequisite relationship (in blue); one change in directionality changes structural complexity by 50%; b(.) is the blocking factor and d(.) is the delay factor

The researcher must be consistent with defining the corequisite relationships to ensure the complexity values are internally consistent. If a convention for directionality is not universally applied in a dataset, then the comparability among plans of study is questionable. We could argue for either configuration, and we believe it is immaterial which orientation is chosen; it is critical that the assumption made about directionality should be reported, at least.

Another way to circumvent this decision is to ignore corequisite relationships when calculating the delay and blocking factors; one could argue that corequisite relationships are not as strict as prerequisites and providing them with equal weight might misrepresent the complexity of a given plan of study. The advantage of ignoring the corequisites when calculating the blocking and delay factors is that the solution will be unique, but that uniqueness will come at the cost of model completeness.

Course Timing Conventions

Course timing as an idea emerged in three different ways during our data collection procedures: the value add of term information in structural complexity, issues when plans of study were not specified at the term level, and additional terms that were not Fall or Spring.

Value Add in Term-Weighting Structural Complexity. In these models, term numbers do not add additional information into the *typical* calculations. Instead, they are used to create a more informative and readable visualization. However, as this research strand continues to develop, knowing when courses occur will become more relevant for alternative metrics. For example, if one would like to analyze a curriculum from a transfer student's perspective, knowing which term they enroll would be a critical piece of information, which would be matched up with the relevant courses. Moreover, DeRocchis et al. introduced the idea of *term-weighted structural complexity* [23] The concept involves multiplying the cruciality of a course by the term it occupies, which punishes courses that are part of dense prerequisite structures later in the curriculum. Researchers should consider exploring the weighted and unweighted structural complexity to parse the value add of the term information.

When Plans of Study Are Not Organized by Term. Occasionally, a plan of study will not specify when a course is taken and may only provide the year or simply a list of courses. Questions about this occurred more frequently early in data collection:

- The sections are according to years, not semesters. What do I do about this?
- There's an extra summer semester in which they take 12 credits of electives. What do I do about this?
- And I've also made quite a lot of such assumptions for the 3 years of [INSTITUTION], they didn't have a semester-by-semester breakdown too. Where do I document every assumption, I made? In the same row as the course where I assumed?

For this, we would suggest first entering all the required courses and prerequisite information, then organizing the courses term by term based on the prerequisite relationships. To form coherent plans of study, using credit hour loads to check the reasonability of a configuration is advised. If the number of credits is too small, then the student will not have full-time status. If the number of credits is too large, then the student would possibly incur overload charges. Move around less connected courses before adjusting the timing of courses in denser prerequisite structures. Some catalogs provide information on when the courses are offered, which can clue the research in to where the student is most likely to take the course in question. If not, searching the course on a website like Coursicle provides the recent semesters it was offered.

Summer and Winter Terms. There might also be situations where a course is specified to be taken during a Summer or Winter term. A simple solution is to treat the extra term just like a Fall or Spring session and count the terms normally. Alternatively, and what is used in the R package in development, is the convention of 1 = Fall, 2 = Spring, 3 = Summer, 4 = Fall, 5 = Spring, 6 = Summer and so on. Similarly, for Winter terms, 1 = Fall, 2 = Winter, and 3 = Spring. This convention makes it more visually obvious that a Summer term is incorporated and allows us to calculate metrics that incorporate limited offerings into our calculations [10], [11]. However, by labelling the terms using the explicit Summer term, we inevitably cause issues when calculating term-weighted metrics. To accommodate the labelling convention, the package automatically adjusts for the empty terms by discarding them and renumbering the terms.

To record plan of study level assumptions, notes about Summer courses like what is discussed in this section were entered into a dataset-level Word document. Upon completion of data collection, this Word document serves as a manual for the dataset allowing other researchers to understand the way in which the individual plans of study were created – increasing research transparency.

Representing Electives

When entering courses into a plan of study, different types of electives are likely required, such as general education courses and major-specific courses. Questions about different kinds of electives were common in our chats and became more intricate as more specialized arrangmenets were observed:

- For [INSTITUTION] in the second semester of senior year there are many courses with no credit hours, how should I enter these?
- When it gives the student the option between choosing classes, how should we show this? [INSTITUTION] for a major in Electrical Engineering gives students the option to choose a concentration.
- I already have an option for taking 2 classes and was wondering what I should do with that.
- For a base course with no other options but varying course credits, what should we do?

Unfortunately, applications of curricular complexity do not explicitly address how one would incorporate student choice at such a level in the Curricular Analytics model. The most common way of representing electives is to simply leave the course unspecified and name it according to the elective type, such as "Gen Ed" or "Tech Elective." Because the course is not specified, there is no prerequisite or corequisite information entered as well. Thus, the plan of study captures the *base* complexity as envisioned by the faculty, which does not align with how a researcher may casually discuss the complexity of the curriculum using this framework.

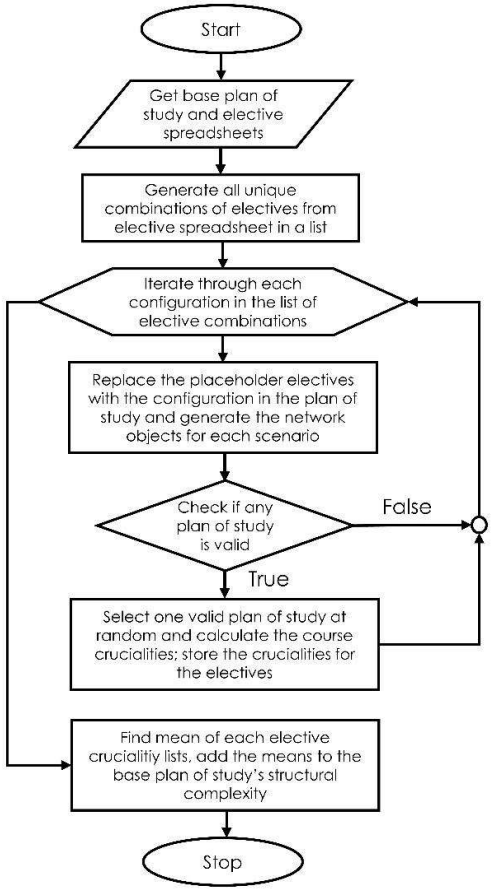

Figure 3. An algorithm for estimating the structural complexity to incorporate student choice in electives

A more complete picture of a plan of study's complexity can be obtained by splitting the data into two pieces, one "base" spreadsheet and one "elective" spreadsheet. The base spreadsheet captures exactly what a researcher would enter when applying the Curricular Analytics framework. When giving an elective a name, ensure it reflects the type of elective. For example, suppose there are two types of technical electives that students need to choose called Depth electives and Breadth electives. In the elective spreadsheet, we would copy the list of courses under these electives and create a new column that associates the courses with their elective group. The elective spreadsheet will serve as a pool of possible courses to pull from to estimate the expected structural complexity when we factor in student choice with electives. The algorithm for estimating the cruciality for each elective is given in Figure 3.

Using the base plan of study and elective list, we create all possible permutations of courses and store them in a list. In R, the data can be stored like so:

```
configurations <- list(c('ENGR 100','ENGR 101','ME 100'),c('ENGR 100', 'ENG
R 101', 'ME 191')) #only first two entries shown
```

Note that there will be repeated entries as individual courses are swapped out to form a new configuration, but repeated entries where a course simply switches spots with another are discarded. Once the list is generated, use a for loop to run through all the configurations. Check if the configuration is valid, meaning that prerequisite and corequisite relationships are not violated. If it is not valid, rearrange the courses into different elective slots on the base plan of study. If the plan of study is valid, then the cruciality of each elective is calculated. Store those in an atomic vector or data frame to keep track of the values for each configuration. Once a passable arrangement is found, we will move on to the next configuration. At the end of the loop, take the mean of the individual elective crucialities to obtain an estimate of the expected cruciality. To yield the expected structural complexity, add the estimates to the base structural complexity.

The resulting adjusted value then incorporates the ability for students to choose a subset of their courses. We are planning to incorporate this functionality into our R package to perform the necessary analysis.

Webscraping to Automate Data Collection

During our data collection processes, we considered different ways that we could automate data entry at least partially. Because the course catalogs also had descriptions of all the different courses that were available at the institution for that catalog year, a web scraper in Python could be used to scrape these descriptions, including the course code, full course name, number of credit hours, and the requisite paragraphs. The BeautifulSoup library was used in conjunction with Pandas to arrange the data into the standardized format. From there, only the course codes and the terms columns needed to be entered, and Python could be used to import the plan of study file, convert it into a data frame, analyze which courses were taken, import the descriptions of those courses from the scraped catalog data, and fill in the full course names, the number of credit hours and the requisite structure, and update the plan of study file with this information. Then, the researcher would only have to look at the requisite structure, determine which courses should be entered into prerequisites and corequisites, and enter that information. For institutions with predictable and standardized HTML formatted catalogs, this process enabled us to move much more quickly than manual data entry.

Issues with Automating Data Collection. However, when conducting a study with multiple institutions across multiple year, automating the data collection in the way we described was not feasible. The complete data entry, especially as described in Table 1, was generally not time efficient from a programming perspective because of the various ways institutions would organize their course descriptions. Instead, whereever possible, the pre- and corequisite information was scrapped into files for the remaining cleaning to be done manually. For example, consider a course taken in term 2, PHYS 2001, has prerequisites, MATH 2001, or MATH 2002. In this case, the prerequisite entered into the spreadsheet would depend on which MATH course was taken in term 1, MATH 2001 or MATH 2002. So, it was a difficult task to extract the requisite structure in a format that could make automation simple given the different formats different institutions used for their catalogs,

and even the difference in formats within an institution. Some institutions had a direct approach and listed out the requisites explicitly, whereas others had a descriptive paragraph to explain the structure. Within an institution, most lower-level courses had simple structures, but higher-level courses tended to have a more complicated structure. Therefore, it was difficult to write a "one size fits all" code, so each institution would have to be handled differently. For other researchers considering a similar approach, consider the costs and benefits of creating a web scraper to handle the data entry – especially when exploring diverse institutions.

A Flow Diagram Synthesizing Our Data Collection Framework

To synthesize our frequently asked questions and processes developed during our data collection process, we offer a suggested process for reducing ambiguities while completing data entry for Curricular Analytics at scale. We divided it into three components: pre-processing (Figure 4), processing (Figure 5), and post-processing (Figure 6). In pre-processing, we made a pdf of the source material, such as the website or catalog page using the naming convention: *InstitutionName_CatalogYear_DisciplineName_POS.pdf*. For the data file, we used the convention *InstitutionName_CatalogYear_DisciplineName_Base.csv* for the specified courses and *InstitutionName_CatalogYear_DisciplineName_Elective.csv* for elective. The CatalogYear will be whatever year range is on the title of the catalog. For example, if the catalog year is 2021-2022, then the CatalogYear will be 2122. The pre-processing step takes into account cases where summer terms appear (or winter terms, equivalently) and if courses are not assigned to terms or semesters (such as by year). The processes in Figure 5 handle data processing, including all of the conventions we established earlier for non-trivial prerequisite structures, embedded labs, and additional enrolment requirements. In Figure 6, we clean up any loose ends by checking for mutual corequisites and sorting courses without term assignments because courses were not specified by semester by rebalancing courses free to move until credit hour distributions fall between the minimum to be enrolled and below the maximum.

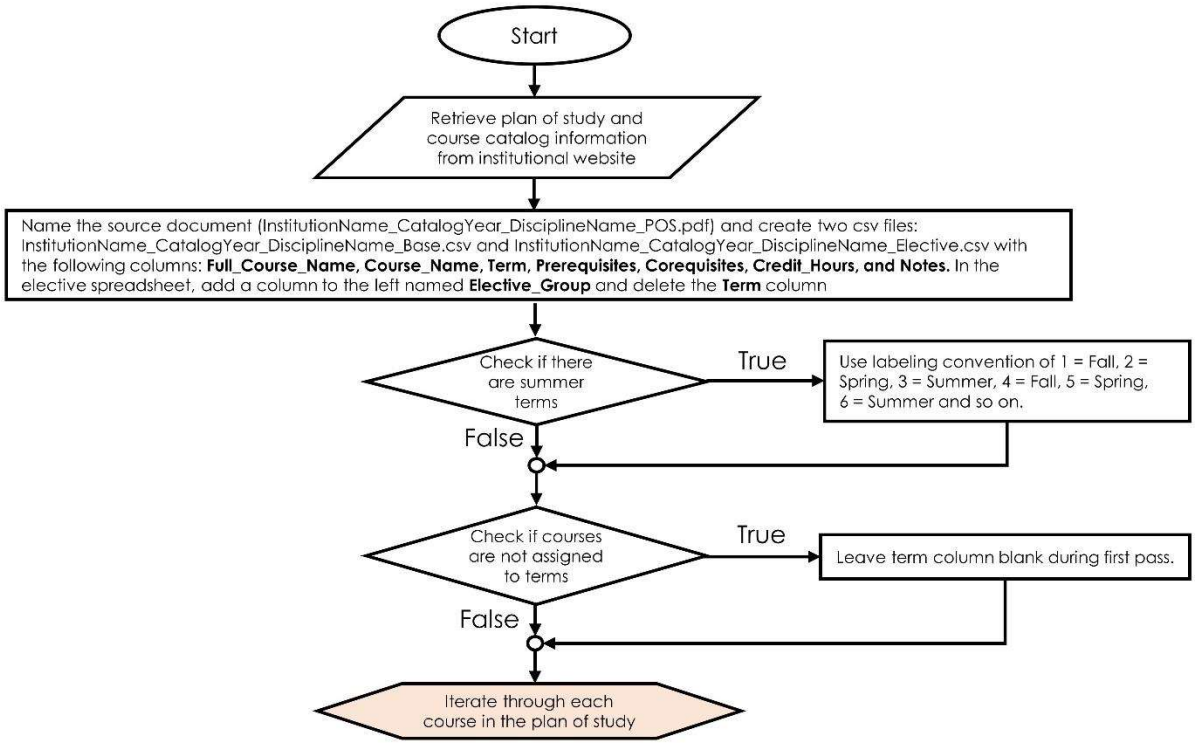

Figure 4. Pre-proccessing before entering data for a plan of study

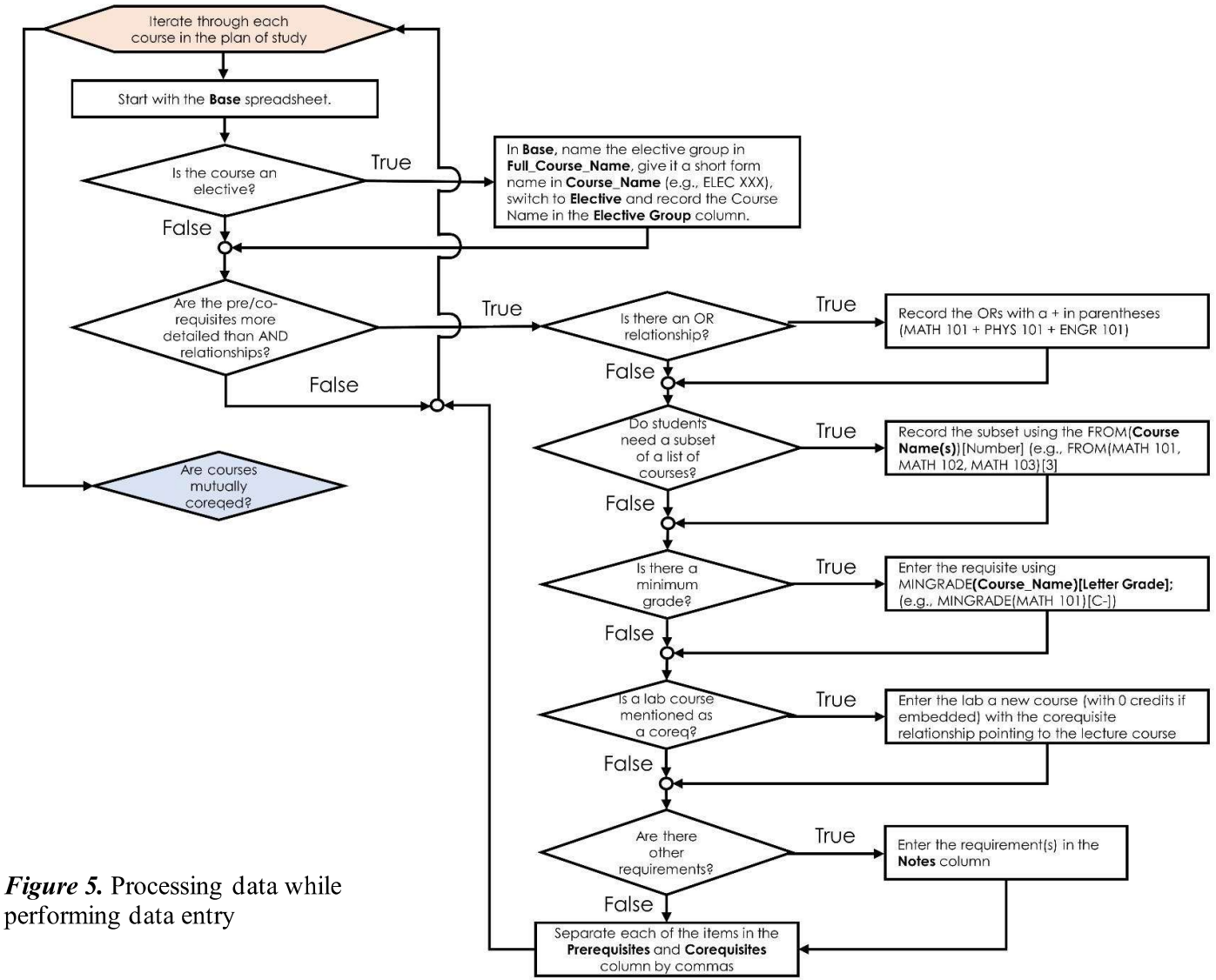

Figure 5. Processing data while performing data entry

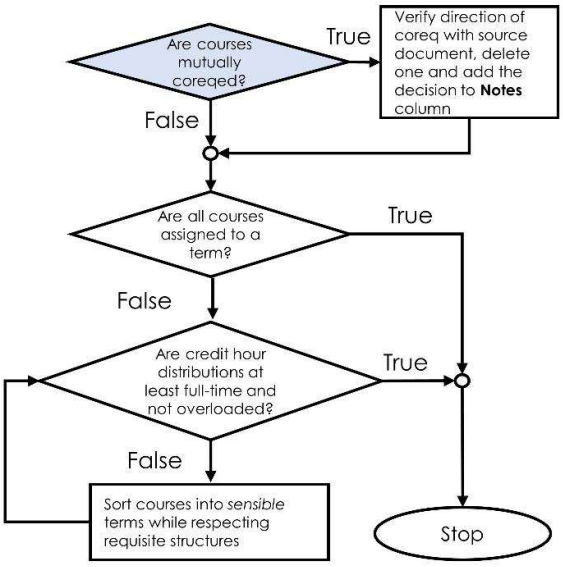

Figure 6. Post-processing after completing entry

Contributions to Theory in Curricular Analytics

We contend that this work advances the theory and application of Curricular Analytics in engineering education research in two distinct areas: (1) establishing a process for systematic data collection and (2) extending how researchers can account for multiple pathways in terms of electives.

As of writing, there is little to no guidance for how a research team would collect data for Curricular Analytics at scale. We have illustrated through our findings in this paper that entering plan of study data across departmental, institutional, and temporal contexts carry heavy implications in terms of assumptions that must be made to achieve the desired formatting. Prerequisite and corequisite relationships are often not logically simple - that is, a set of courses that must all be taken to enroll in the course. Instead, prerequisites can have both AND and OR logical connectives (e.g., (MATH 101 and MATH 102) or (MATH 103H)), subsets of a longer list of approved courses (e.g., three of the following: ....), minimum grade requirements (e.g., MATH 101 with min C-), or a combination of all three.

Second, we advanced how future researchers can think about a broadened perspective on a plan of study's complexity. One potential critique that can be made of how data are typically entered for studies on curricular complexity is the simplification of electives. For general education courses and other requirements that are mostly independent of the major-specific courses, specifying these has little impact on the structural complexity metrics. As such, they are often labelled as "Gen Ed" or some equivalent and left without prerequisites or corequisites.

However, applying the same treatment to major-specific electives is less defensible analytically. Often upper-division courses that serve as electives have prerequisites, which are lost when they are aggregated into a generic "Tech Elective" category. Yet, incorporating the electives into the plan of study directly poses a different problem. We could specify the most common sets of electives students take, but this decision would ignore the students'

agency afforded to them by the premise of these elective courses. On the other extreme, we could create a plan of study for every possible pathway, but this would be a tedious data entry task.

Our suggestion in this theory paper is a two-staged data collection process that balances the desire to calculate the complexity of a curriculum using available data and incorporate student agency into the analyses. When we do not incorporate student agency, meaning we do not specify elective courses or pathways, we are not calculating the true structural complexity of the curriculum. We are calculating the expected *base* structural complexity of the curriculum *as envisioned by the faculty.* By base, we are referring to the set of courses and their prerequisites that are specified as required by the faculty in the plan of study. We add the qualifier "expected" because students have agency over the base structural complexity as well – not just electives – through transfer credit, AP credit, and exceptions. Thus, by reframing how we think and talk about structural complexity, albeit slightly, we can be more precise in its theoretical and practical applications.

Conclusion

We anticipate this work being useful to researchers and practitioners interested in systematic analyses of curricula, especially in combination with student data to explore retention-related issues for first-time-in-college students. The dataset being created will be freely available, and others are encouraged to add their own plans of study. We offer the standard operating procedures in this paper, along with data conventions, to best facilitate the large-scale analysis of this type of network data. As the dataset grows, we anticipate the ability for the community to understand and interrogate the programmatic barriers to student success in engineering across the nation will also expand – leading to a cornucopia of previously unexplored questions at scale.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. XXXXXX. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.