Analyzing Trends in Curricular Complexity and Extracting Common Curricular Design Patterns

Sushant Makarand Padhye, University of Cincinnati

Sushant Padhye is a sophomore majoring in electrical engineering and an Undergraduate Research Assistant at the University of Cincinnati.

Dr. David Reeping, University of Cincinnati

Dr. David Reeping is an Assistant Professor in the Department of Engineering and Computing Education at the University of Cincinnati. He earned his Ph.D. in Engineering Education from Virginia Tech and was a National Science Foundation Graduate Research Fellow. He received his B.S. in Engineering Education with a Mathematics minor from Ohio Northern University. His main research interests include transfer student information asymmetries, threshold concepts, curricular complexity, and advancing quantitative and fully integrated mixed methods.

Nahal Rashedi , University of Cincinnati

PhD Student of Engineering Education

Analyzing Trends in Curricular Complexity and Extracting Common Curricular Design Patterns

Abstract

This research paper explores how curricular design patterns can be extracted from plan of study data systematically. Engineering is a notoriously sequential discipline in its curricular design, with courses organized using a variety of prerequisite and corequisite structures to filter students through specific experiences in order. Considering how these arrangements can influence student progress toward their degree, such as the sequencing of gatekeeper courses like Statics, we posed the following research question: "What are the common and uncommon curricular design patterns in degree programs across the United States for mechanical, electrical, civil, industrial, and chemical engineering?" We leveraged existing data collected as part of an ongoing project about connecting a data-driven framework for evaluating curricula (i.e., Curricular Analytics) to the Multiple Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) to provide a new perspective for analyzing student success outcomes, such as graduation rates. Unlike previous efforts with Curricular Analytics, these data span 13 universities across five disciplines over a decade – comprising 494 plan of study networks. We contend analyzing this dataset and using it in conjunction with student data in MIDFIELD can help the community glean new insights and contextualize previous findings. We employed network analysis and data mining techniques in Python to uncover the curricular design patterns in the dataset for courses related to Introduction to Engineering, Calculus, and Mechanics (e.g., Statics, Dynamics, and Strength of Materials). We also juxtapose results about curricular complexity across this dataset that can be used alongside our network analysis efforts for further research.

Introduction

Since ABET's transition to an outcomes-based philosophy in the accreditation process, engineering faculty have more freedom to structure engineering programs instead of following overly prescribed disciplinary criteria [1]. Thus, engineering programs can exhibit different organizational structures when defining required coursework – which can be influenced by many internal and external factors [2]. For example, faculty at Wright State University, led by Nathan Klingbeil, have published extensively on a model of introductory engineering mathematics courses that circumvents the necessity of the Calculus sequence [3], which has shown considerable success in improving graduation rates for students who would be traditionally deemed underprepared for Calculus [4]. Similarly, Ellis et al. [5] report on a Calculus sequence that was rehoused within the College of Engineering and reworked to focus on applications, trimming the number of courses in the sequence from five down to four. Although pathways like Calculus seem cemented as commonplace, these reform efforts and the diversity in other seemingly standard offerings like first-year introductory experiences is a testament to how different programs can be structured to achieve college- and department-specific goals [6].

In light of the diverse ways faculty can choose to implement a plan of study for engineering students, this paper focuses on understanding the sequencing and overall arrangement of courses in a program. We adopt the terminology from Heileman et al. [7] to formally call these constructs *curricular design patterns,* which they describe as, "collection[s] of curricular and co-curricular learning activities intentionally structured so as to allow students to attain a set of learning outcomes within a given educational context" [p. 5]. Although the term *co-curricular* is used in this definition, there is much greater emphasis on the *structure* of prerequisite and corequisite relationships. Still, by examining these roadmaps for how students are expected to progress through their discipline's plan of study, we can understand how different curricular design patterns can lead to a range of student outcomes – such as graduation rates. This analytical possibility bridges research interests on persistence and student success with administrative benchmarking, leading to the study of curricular design patterns being a fruitful area of inquiry.

Research Aims

We posed the following research question: "What are the common and uncommon curricular design patterns in degree programs across the United States for mechanical, electrical, civil, industrial, and chemical engineering?" As part of this effort, we leverage a new (soon to be) public dataset of curricular requirements for engineering programs linked to a large popular dataset for studying engineering student trajectories in attaining their degrees [8]. Thus, this effort is one example of how the dataset can be used for future research and highlights the analytical richness of curricular data.

Work has begun to emerge regarding how to optimize curricula to reduce "curricular complexity." For example, Zhang [9] used an integer quadratic programming algorithm to rearrange courses that met an established set of learning outcomes but with comparatively lower complexity. Our paper delves into course sequences unearthed within historical plan of study data for 13 institutions in the United States, providing a valuable resource for researchers and practitioners at various institutions. When combined with student data, these discovered sequences offer a fertile ground for researchers to guide their curriculum analysis and redesign efforts.

Background

We have referred to the idea of "curricular complexity" loosely so far, but we can be more precise by using a framework that is growing in popularity when describing curricular design patterns. The formal analysis of curricular design patterns can be accomplished using a framework called *Curricular Analytics* [10]. The adoption of Curricular Analytics reflects a paradigm shift toward a data-driven approach to analyzing curricula and degree requirements. This method quantitatively assesses the "complexity" inherent in a plan of study; at its core, Curricular Analytics captures and models the intricate web of pre- and corequisite relationships within a curriculum, transforming it into a network. This network representation visually depicts degree requirements and, by virtue of its data type, becomes a rich testbed for in-depth analysis using network analysis techniques. The fundamental premise of Curricular Analytics posits a relationship between curricular complexity and completion rates. Both simulation studies and

empirical evidence support the notion that as the complexity of a curriculum increases, completion rates tend to decrease [10], [11]. This insight underscores the practical relevance of understanding and managing the complexity inherent in program structures. If faculty can locate curricular design patterns with high complexity in a program, then making alterations to them to reduce the complexity can have a tangible impact on completion rates.

Curricular complexity is divided into two components: instructional complexity and structural complexity [10]. Instructional complexity attempts to capture the latent factors of the curriculum, such as course difficulty and instructional quality, but it is currently only proxied by a course's pass rate. We focus on the structural complexity of curricular design patterns, which assigns a score to a plan of study based on the interconnectedness of program requirements (i.e., pre- and corequisites). Among the metrics proposed for structural complexity, two have become central to how we calculate the overall structural complexity. These are visualized in Figure 1. 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.

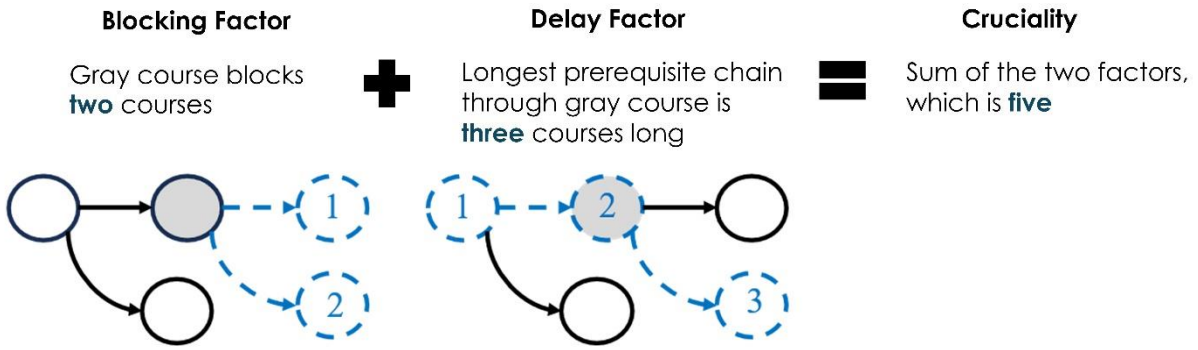

Figure 1. Calculating course cruciality using the blocking and delay factors of the gray course

A notable advantage of representing curricula as networks is the ability to decompose them into smaller subnetworks. These subnetworks serve as representations of distinct curricular design patterns. For instance, one can isolate patterns like the Calculus sequence or the core Mechanics sequence (comprising Statics, Dynamics, and Strength of Materials) – which have seen some attention regarding their impact on completion rates [12], [13], [14]. This decomposition facilitates a nuanced exploration of specific course sequences and their relationships, shedding light on the underlying structure of the curriculum. However, considering the steep data requirements and lack of consistency in course names, little work explores how these curricular design patterns manifest in engineering programs across the United States. However, with a large enough dataset comprised of plans of study across engineering disciplines, it is possible to uncover these patterns within engineering curricula across institutions.

Insights drawn from interrogating curricular structures can translate to practical revisions in engineering programs, as exemplified by the redesign efforts at the University of Virginia to improve their Calculus sequencing [15]. Their work resulted in creating new courses across two tracks: a Core track and an Honors track that repackages topics from Calculus 1 and 2, deviating from the conventional Calculus sequence at their university that assumed students came in ready to take Calculus II. The success of the redesign initiative described by Pisano et al. [15], substantiated by positive outcomes, underscores the pivotal role that course sequence information and student performance data can play in steering curricular innovation. In fact, the aforementioned Wright State model has been viewed through the lens of Curricular Analytics to explain the program's success [16].

Methods

Data Collection and Preprocessing

The dataset used in this research encompasses curricula data from five engineering disciplines (i.e., Civil, Chemical, Electrical, Industrial, and Mechanical) across thirteen institutions in the Multiple Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) [17], which included required courses, prerequisites, and corequisites. We collected the curricular data starting from the most recent record in the dataset and went back nine more years to gather a longitudinal perspective on how the curriculum changed over time. There were minor gaps in the dataset in cases where the plan of study was not readily available or when the discipline was not offered at an institution, leading to a final count of 494 plans of study. Additional details of the data collection process can be found in Reeping et al. [8].

Prior to analysis, a preprocessing stage was conducted to enhance the quality of the data. The preprocessing involved addressing irregularities – such as special characters introduced from copying and pasting data from a website – and any typographical errors.

Mining Curricular Design Patterns

We used Python along with established libraries such as NetworkX and pandas to process the data. Each step conducted during this analysis is shown in Figure 2.

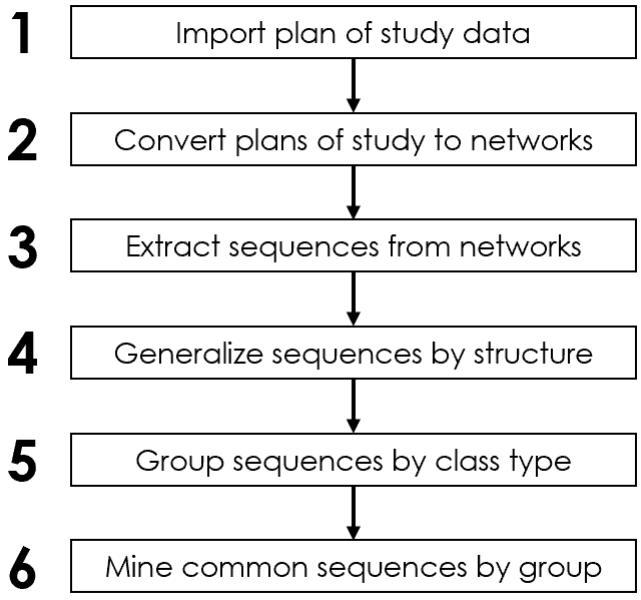

Figure 2. Flow chart for the network analysis process

Step 1: Importing plan of study data: In this step, the pandas library was used to import all the plan of study files (.csv format) into dataframes and organize each one with their id's (names) in a dictionary format. This format was crucial because there should always be a way to retrieve the file path for each imported plan of study in the later stages of analysis.

Step 2: Convert plans of study to networks: Networks are an effective way to represent plans of study programmatically; classes can be treated as nodes and requisite relations between classes can be saved as edges. Nodes can also carry extra information like the term in which they appear in the plan of study, which, while not a focus of this paper, can be helpful in different analyses regarding curricular complexity or sequence mining in specific years of the plan of study, for example, the first-year engineering curriculum or Mechanics sequences in the middle years.

Step 3: Extract sequences from networks: The saved networks were passed to a function that traversed each network and collected all sequences from the plan of study. As shown in the example in Figure 3, we retrieve the relationships between courses step by step to build each distinct curricular design pattern. After discovering that Calculus 1 is a prerequisite for Calculus 2, it looks deeper into the sequencing for Calculus 1 and finds a corequisite relationship with Physics 1. This relationship is added to the larger sequence of courses leading into Calculus 2.

When comparing sequences, using lists was found to be the simplest way for Python to interpret sequences and check for equivalencies between them. These sequences (lists) were saved in dictionaries with keys for each sequence. These keys contained information about the class that was the subject of the sequence (Calculus 2 in Figure 3) and which plan of study it was extracted from.

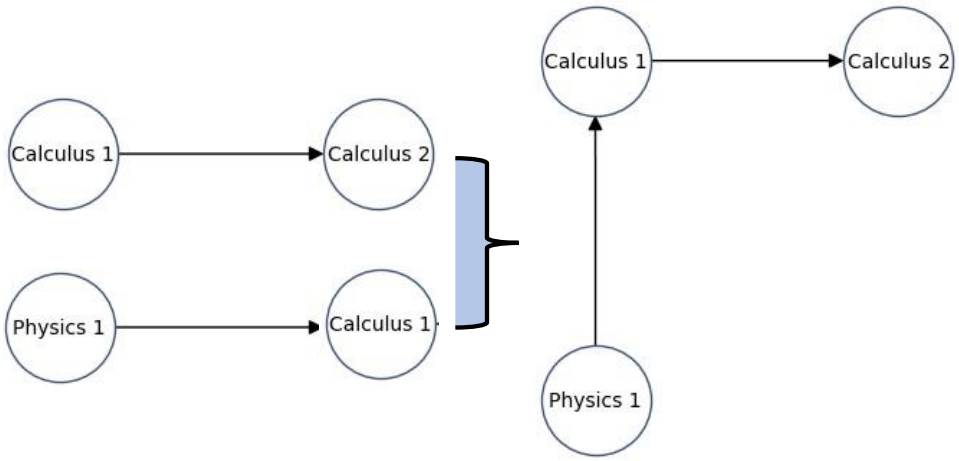

Figure 3. Sequence mining in networks

Step 4: Generalize sequences by structure

When the edges for requisites were added in step 2, the nodes had the course code (e.g., MATH 1001) as their title. Because these codes are different across universities, years, and even within disciplines at the same university, we generalized each sequence by replacing course codes with labels in the form of 'Course n', where n represented the position of the class in the

list/sequence. This way, it was easier to implement equivalency of sequences while the original courses' information was retained with keys, as mentioned in step 3. So, these sequences were stored in a dictionary format with a key-value pair: course information and the sequence.

Step 5: Group sequences by type of class

This study aimed to mine and compare unique sequences for different categories of courses across the dataset, focusing on intrinsic examples like Calculus, Differential Equations, Mechanics, and Introduction to Engineering. The keys containing course information could be used to sort each sequence into the desired categories. However, given the vast scope of the dataset and the diversity in course naming conventions, manual generalization of course chains based on subject matter was impractical. As shown in Figure 4, there were ~12,300 course names in the dataset, with ~1200 of them being unique.

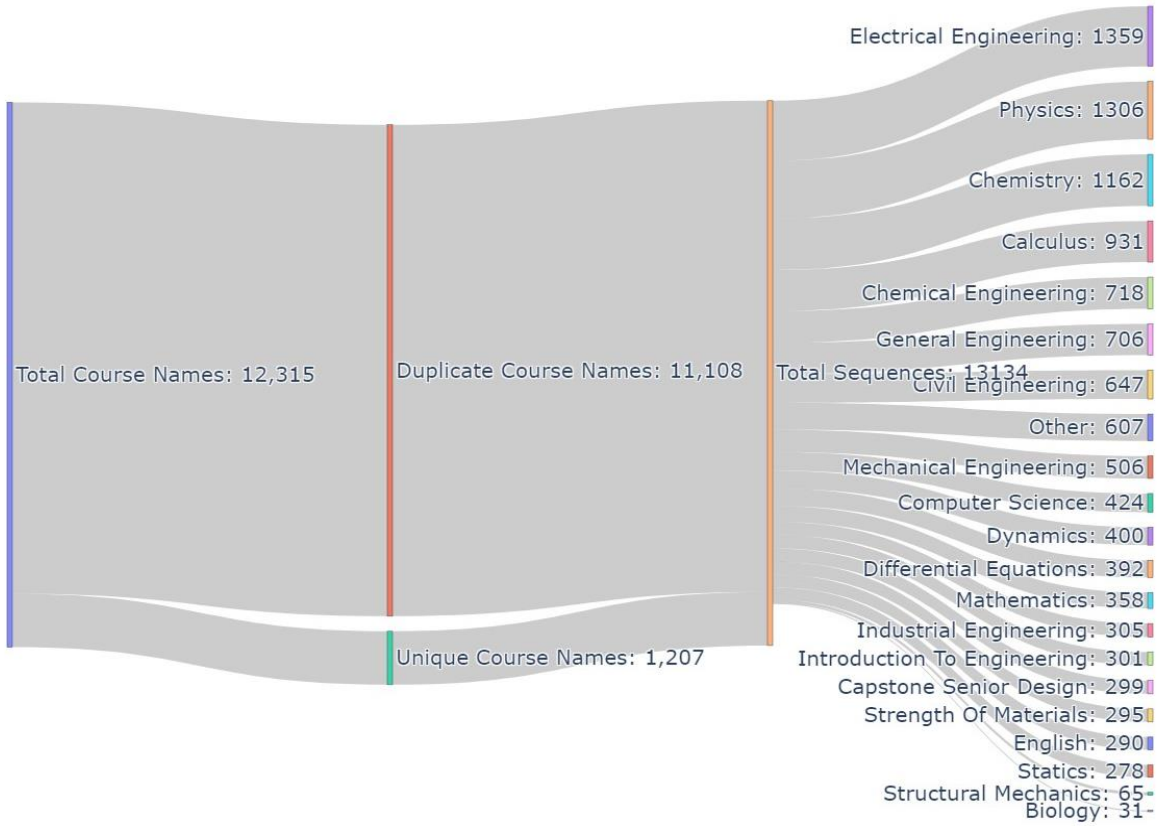

Figure 4. Sankey Diagram of course name and sequence distribution by category

To address this challenge, we employed OpenAI's large language model (LLM), GPT-4, using their API [18]. We initially experimented with GPT-3.5 but faced issues with short context windows (i.e., the amount of text the model could process) and hallucinations. Because of the sheer number of course names, the input needed to be divided into batches of fixed windows to send to the API. When a window width of forty names was used, the model started 'forgetting' the prompt commands given, and output quality degraded. This problem was solved by using a

shorter window width of twenty names, but another issue we faced was formatting. GPT 3.5 often incorporated typos when formatting its output as a dictionary/.json. This was difficult to work with because each file needed to be manually inspected and fixed for formatting errors, which was time-consuming.

These limitations prompted us to use the GPT-4 model, which performed better and kept formatting consistent across outputs. It was also more ideal regarding its context window; we could safely input up to eighty course names at a time and get reasonably accurate categorizations. These minor errors were subsequently corrected by hand. Since there were over one thousand unique names and a smaller window width was necessary to control unintended errors, there needed to be a workflow in the form of a loop, as shown in Figure 5.

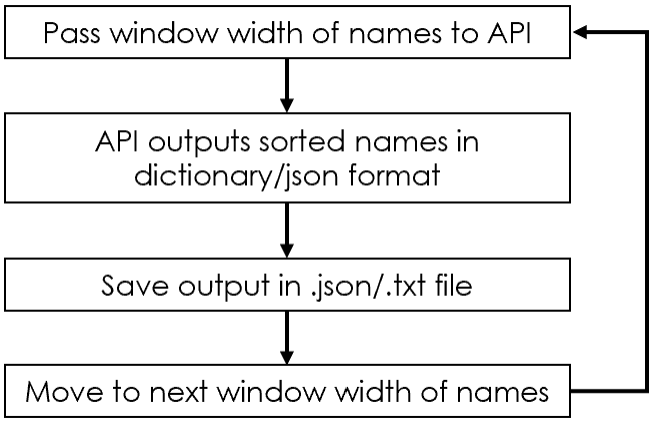

Figure 5. Workflow for ChatGPT API Usage

Because of the large scale of the dataset and the number of course names, we found it better to perform the categorization in two steps. First, we used the API to sort all the course names into general categories like the sciences (math, physics, chemistry, and biology) and engineering (general engineering, mechanical, civil, electrical, and chemical engineering). We then used particular categories to sort them into specific types of courses. For example, we broke down the math category and sorted those courses into Differential Equations and Calculus.

The following code block contains the prompt we used to accomplish the categorization. The {} represents variables; the ‘courses’ variable contains the list of course names, and the ‘categories’ variable contains the list of categories the courses should be sorted into.

```
task f"""{courses}  This is a list with course names.
        Sort all of these using the course name into distinct categories like {categories} only.
        If there are any that you can't sort, place them into the category "Other".
        Don't make any other categories.
        Return a dictionary in JSON format
        {{Physics: course name 1, course name 2, Math: course name 1, course name 2 .....}}.
         The sorted courses absolutely need to be in a dictionary format for Python to use.
         Don't add any lines between the start of the dictionary and the first element
```

or the end of the list and the last element.
Don't write code, just sort the course names and give an answer yourself.
Don't write any text with it."""

Step 6: Mine common sequences by group (e.g., Calculus)

In this step, we imported all the saved API output files and used a dataframe to store the categories and sequences. After this, the sequences were grouped into their respective patterns. The output was a single dataframe with information about the prerequisite chain, such as its category, length, and count of occurrences. We could also use the dictionary keys containing course information from Step 3 to retrieve the occurrences from the plans of study themselves.

Limitations

Despite the measures taken to bolster the quality of the research design, there are limitations regarding the inferences generated. First, it is crucial to note that these data represent a subset of institutions in the United States. They were chosen because of their intrinsic connection to a larger data-sharing agreement that opened more analytical possibilities by linking the curricular data to actual student pathways. Moreover, the data are longitudinal in nature. Therefore, some counts for each curricular design pattern are duplicate observations. However, each curricular design pattern is *unique* – which is what matters for this particular study.

Results

After filtering the data for categories of interest – including Introduction to Engineering, Calculus and Differential Equations, and Statics and Mechanics – we retrieved course sequences that occurred frequently and infrequently. We present a selection of rare sequences to highlight the diversity of curricular design patterns across the 13 institutions within the mined categories.

Introduction to Engineering

Introduction to engineering courses are multidisciplinary offerings that students take before entering their discipline-specific coursework and include myriad topics such as problem-solving, graphical representations of data, elementary statistics, engineering drawing and graphics, disciplinary concepts like electric circuits, and computer programming [6]. A typical sequence for these classes involves a corequisite like ‘Calculus 1’ (78 observations), then ‘Introduction to Engineering 1’ followed by ‘Introduction to Engineering 2’ (63 observations). These introductory courses can also be requisites for other classes in the plan of study, such as discipline-specific offerings like Statics or Circuits. There were 24 distinct curricular design patterns for the Introduction to Engineering sequences.

One example of an introductory sequence we found was from East Carolina University. In this case, Engineering Graphics, which can be a topic in Introduction to Engineering courses, was placed as an independent course as a prerequisite to the second Introduction to Engineering course. As shown in Figure 6, along with Introduction to Engineering 1 (just Introduction to Engineering in this case), students must take the Engineering Graphics course in their first term. Peculiarly, the Engineering Graphics course has a Calculus 1 corequisite as well. The sequence continues with the equivalent of Introduction to Engineering 2 (Introduction to Engineering

Design here), which is followed by another design course with project management embedded within the offering. This is one example of our dataset's more complicated introductory engineering course prerequisite chains, which has an overall structural complexity of 21. Note that removing the potentially unnecessary Calculus 1 corequisite alone decreases the structural complexity by 19% (21 → 17).

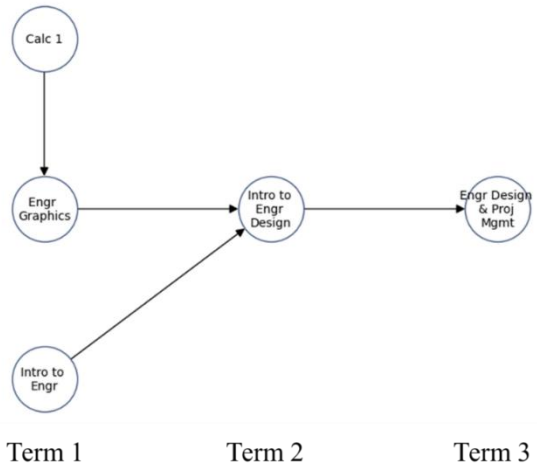

Figure 6. Introduction to Engineering sequence- East Carolina University- Electrical, Mechanical and Industrial Engineering

Whether these introductory courses serve as bottlenecks for students hinges on how the Introduction to Engineering sequences are connected to later offerings. We observed cases of isolated first-year courses and well-connected first-year courses. For example, we found North Carolina State University's introductory engineering course, "Introduction to Engineering & Problem Solving," is only a prerequisite for another introductory course, "Engineering in the 21st Century" – after which the sequence ends. This arrangement would have a structural complexity of only 5. Alternatively, Purdue University's introductory course, "Transforming Ideas To Innovation 1" and "Transforming Ideas To Innovation 2" connect to "Linear Circuit Analysis 1" and later courses in the sequence, "Thermodynamics 1," "Basic Mechanics 1," and "Basic Mechanics 2" – embedding the sequence is a more extensive web of prerequisites.

Calculus and Differential Equations

Differential Equations is often a required course in engineering curricula and is typically the culminating course in the Calculus sequence. It is common to see these sequences as two or three calculus courses followed by Differential Equations (181 and 154 observations, respectively). However, some institutions have a different sequence. For example, East Carolina University had a structure with three calculus courses and a computer applications course followed by Differential Equations. In particular, Elizabethtown College's programs exhibited a unique arrangement in their Calculus sequencing. As shown in Figure 7, Calculus 1 is paired with a lab class, and it is a direct prerequisite for Calculus 3 instead of Calculus 2. Calculus 2 is also a direct prerequisite for Differential Equations. This is an unexpected sequencing and may indicate that the topics taught as part of the third Calculus course may include physics topics solved using 3-dimensional calculus and hence is used as a prerequisite. Decoupling Calculus 2 from Calculus 3 reduces the complexity of the curricular design pattern by 20% (44 to 35), the benefits of which are magnified when considering curricular design patterns this sequencing is connected to.

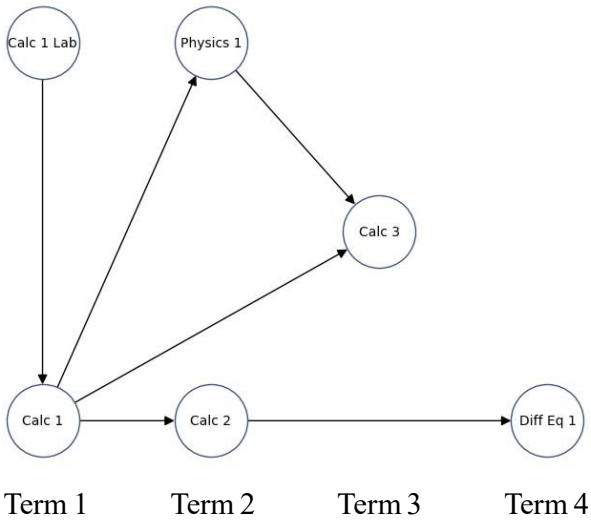

Figure 7. Calculus and Differential Equations sequence- Elizabethtown College- Civil, Electrical, Mechanical, and Industrial Engineering PoS

Still, the Calculus curricular design patterns are mostly homogeneous across the sampled institutions, as shown in Table 1. Note that sequences can be nested in more complicated sequences; for example, 'course 1' -> 'course 2' is nested in 'course 1' -> 'course 2' -> ‘course 3’ – meaning there are 13 'course 1' -> 'course 2' sequences for Calculus.

Table 1. Distribution of curricular design patterns for Calculus, course pairs in the brackets are taken in the same semester together (i.e., there exists a corequisite relation between them)

Curricular Design Pattern	Count
'course 1' -> 'course 2'	444
'course 1' -> 'course 2' -> ‘course 3’	431
('course 1' -> 'course 2') -> ‘course 3’	12
'course 1' -> ('course 2' -> 'course 3') -> 'course 4'	14
('course 1' -> 'course 2')	22
('course 1' -> 'course 2') -> 'course 3'	5
'course 1' -> 'course 2' -> 'course 3' -> 'course 4'	3

Statics and Mechanics

Compared to the study of the first-year engineering curriculum and calculus sequencing, courses that sophomores and juniors take in the middle years, like Statics and Mechanics, are still mostly unexplored [19]. However, considering the dataset has nearly 500 complete plans of study across universities, it was possible to explore these core intermediate-level courses and analyze their sequencing.

There were 28 different design patterns associated with Statics. Most commonly, programs in the dataset have coursework starting with a Statics course in term 3 or 4 and tend to have Calculus and physics courses as prerequisites. As shown in the sequence in Figure 8, the Statics and Mechanics sequences can have some variety in their structure. The difference between universities usually depends on the level of the Calculus prerequisite that their Statics course has. The sequencing can also differ within a university if specific disciplines have their own version of Statics, including different topics that may need more or less Calculus preparation.

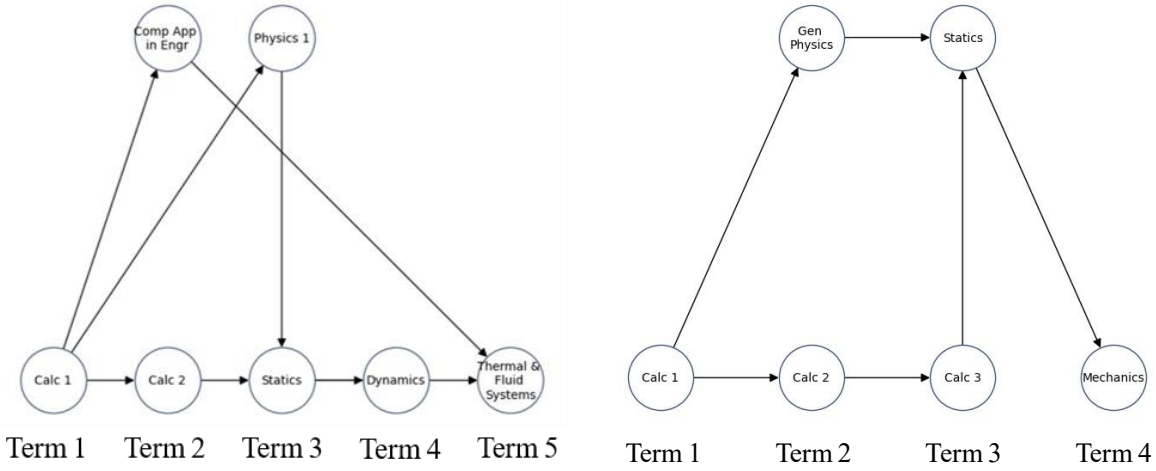

Figure 8. Statics and Mechanics sequence - University of Oklahoma- Civil and Industrial Engineering (left); East Carolina University- Electrical, Mechanical, and Industrial Engineering (right)

In Figure 8, the design pattern at the University of Oklahoma situates Calculus 2 as a prerequisite for Statics with Physics 1 as a corequisite. However, East Carolina University positions General Physics as the prerequisite and Calculus 3 as the corequisite. These arrangements do not differ much on the surface; however, calculating the structural complexity of the sequencing for the four courses (Calculus 1, Calculus 2, Physics 1, and Statics) in the University of Oklahoma design pattern and five courses (Calculus 1, Calculus 2, Calculus 3, General Physics, and Statics) in the East Carolina University design pattern reveals the latter pattern is 30% more structurally complex than the former (i.e., structural complexity of 20 versus 26). Considering the impact of a minor tweak like adding a course to a prerequisite chain, these comparisons can serve as springboards for researchers to analyze their curriculum's sophomore and junior years and thoughtfully revisit their requisite structure.

Discussion

This paper's demonstration of using a curricular dataset to extract course sequences represents an initial exploration of the concept at scale. However, considerable room exists for expanding and delving deeper into these areas. For instance, our discussion of potential avenues for researchers at respective universities to analyze their curricula builds a foundation for extending our work and conducting in-depth examinations in their local contexts. Moreover, leveraging the dataset alongside structural complexity data allows for exploring historical trends. Cross-referencing

structural complexity results with curriculum overhaul plans provides an opportunity to verify the effectiveness of such changes in improving completion rates [10], [11].

It is crucial to note that the structural complexity we've referred to is unweighted structural complexity, wherein the terms in which classes are taken are not considered in the calculation. By extracting curricular design patterns, especially if the extraction is in the middle of the curriculum, the impact of timing on the courses can be lost. However, a more nuanced approach involves term-weighted crucialities (TWC) and term-weighted structural complexity (TWSC), as outlined by DeRocchis et al. [20]. Like the standard cruciality metric, a course with a high TWC is deemed crucial to the curriculum, potentially being a course in the middle years that is a requisite for a lot of other courses or being taken later in the students' degree, indicating less flexibility for retaking the course in case of a non-passing grade. When adding term information into the calculation, courses that previously were the most crucial may be superseded by other courses embedded in similar prerequisite structures but are placed later in the curriculum. TWC can be used to strategically place trivial and non-trivial gatekeeper courses and provide other supports to enhance student success.

The core focus of this paper is to present a robust data-driven methodology for handling extensive datasets and applying network analysis techniques to unearth prevalent and rare curricular sequences. Exploring these sequences holds significant implications for both curriculum planning and discerning the impact of intricate courses on student success. The methodology outlined in this paper, along with the R package introduced in [8], is generalizable and adaptable to other similar datasets. It can be modified to suit other datasets and accommodate various complexity calculations, showcasing its versatility and potential application to broader contexts.

Another contribution of our dataset is the ability to analyze different subsections of the curriculum. For example, researchers can delve into the nuances of intermediate coursework by scrutinizing specific sequences within domains like Statics, Circuits, and Thermodynamics. This capability opens avenues for optimizing the requisites of these classes and identifying potential bottlenecks. Analyses can also extend to courses in the first year. Introduction to Engineering (e.g., [21], [22], [23], [24]) and Calculus (e.g., [25], [26], [27], [28]) curricula have been extensively investigated in previous research. The similarity observed in sequences from these categories, with few exceptions, suggests a degree of curriculum optimization across universities or institutional isomorphism [29]. However, the nature of these courses can evolve over time, and their prerequisites may no longer be sensible. For example, Faulkner et al. [30] reviewed homework problems for circuits and statics courses at a single institution that had Calculus as a prerequisite, finding that only 8% and 20% of problems applied Calculus concepts in some way – often only applying basic principles like integration of polynomials. Thus, the network analysis framework presented in this paper offers a valuable tool for scrutinizing how we construct curricula using a data-driven approach – especially when overlaid with these courses' learning outcomes and content.

Conclusion

This paper presented the first analysis of curricular design patterns in a new dataset to understand how curricula in engineering change over time. By understanding how curricular design patterns impact student outcomes in different contexts, we can leverage these findings to inform future curriculum development efforts. We plan to integrate this work with the Multiple Institution Database for Investigating Engineering Longitudinal Development (MIDFIELD) to enrich our study by linking common curricular design patterns to actual student course-taking trajectories. An avenue for further exploration lies in scrutinizing MIDFIELD trajectories, specifically those of students who transferred out of their major or left college altogether. Analyzing these trajectories offers valuable insights into critical courses that may have influenced students' decisions to transfer or stopout. As we continue analyzing and expanding this dataset, we expect the thoughtful analysis of curricula to expand – enabling faculty to have focused conversations about design patterns impeding student success.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BPE- 2152441. 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.

References

[1] J. W. Prados, G. D. Peterson, and L. R. Lattuca, “Quality Assurance of Engineering Education through Accreditation: The Impact of Engineering Criteria 2000 and Its Global Influence,” *J. Eng. Educ.*, vol. 94, no. 1, pp. 165–184, 2005, doi: 10.1002/j.2168-9830.2005.tb00836.x.

[2] L. Lattuca and J. Stark, “Curriculum: An academic plan,” in *Shaping the College Curriculum*, Jossey-Bass, pp. 2–22.

[3] N. Klingbeil, K. Rattan, M. Raymer, D. Reynolds, and R. Mercer, “The Wright State Model For Engineering Mathematics Education: Nationwide Adoption, Assessment, And Evaluation,” presented at the 2009 Annual Conference & Exposition, Jun. 2009, p. 14.1265.1-14.1265.17. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/the-wright-state-model-for-engineering-mathematics-education-nationwide-adoption-assessment-and-evaluation

[4] N. W. Klingbeil and A. Bourne, “The Wright State Model for Engineering Mathematics Education: Longitudinal Impact on Initially Underprepared Students,” presented at the 2015 ASEE Annual Conference & Exposition, Jun. 2015, p. 26.1580.1-26.1580.11. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/the-wright-state-model-for-engineering-mathematics-education-longitudinal-impact-on-initially-underprepared-students

[5] B. Ellis, S. Larsen, M. Voigt, and K. Vroom, "Where Calculus and Engineering Converge: an Analysis of Curricular Change in Calculus for Engineers," *Int. J. Res. Undergrad. Math. Educ.*, vol. 7, no. 2, pp. 379–399, Jul. 2021, doi: 10.1007/s40753-020-00130-9.
[6] K. Reid, D. Reeping, and E. Spingola, "A taxonomy for introduction to engineering courses," *Int. J. Eng. Educ.*, vol. 34, no. 1, pp. 2–19, 2018.
[7] G. Heileman, M. Hickman, A. Slim, and C. Abdallah, "Characterizing the Complexity of Curricular Patterns in Engineering Programs," in *2017 ASEE Annual Conference & Exposition Proceedings*, Columbus, Ohio: ASEE Conferences, Jun. 2017, p. 28029. doi: 10.18260/1-2--28029.
[8] D. Reeping, S. M. Padhye, and N. Rashedi, "A Process for Systematically Collecting Plan of Study Data for Curricular Analytics," in *2023 ASEE Annual Conference & Exposition*, 2023.
[9] Y. Zhang, "Optimal Design of Curricula and Curricular Pathways in Higher Education," Ph.D., The University of Arizona, United States -- Arizona, 2023. Accessed: Feb. 04, 2024. [Online]. Available: https://www.proquest.com/docview/2901645451/abstract/C22C9176095A4ECCPQ/1
[10] G. L. Heileman, C. T. Abdallah, A. Slim, and M. Hickman, "Curricular Analytics: A Framework for Quantifying the Impact of Curricular Reforms and Pedagogical Innovations," *ArXiv181109676 Phys.*, Nov. 2018, Accessed: Aug. 04, 2021. [Online]. Available: http://arxiv.org/abs/1811.09676
[11] A. Slim, "Curricular Analytics in Higher Education," Ph.D., The University of New Mexico, United States -- New Mexico, 2016. Accessed: Feb. 04, 2024. [Online]. Available: https://www.proquest.com/docview/1873863748/abstract/525913CADB01422APQ/1
[12] J. R. Grohs, M. M. Soledad, D. B. Knight, and S. W. Case, "Understanding the Effects of Transferring In Statics Credit on Performance in Future Mechanics Courses," presented at the 2016 ASEE Annual Conference & Exposition, Jun. 2016. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/understanding-the-effects-of-transferring-in-statics-credit-on-performance-in-future-mechanics-courses
[13] K. A. Wingate, A. A. Ferri, and K. M. Feigh, "The Impact of the Physics, Statics, and Mechanics Sequence on Student Retention and Performance in Mechanical Engineering," presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/the-impact-of-the-physics-statics-and-mechanics-sequence-on-student-retention-and-performance-in-mechanical-engineering
[14] D. Reeping, D. B. Knight, J. R. Grohs, and S. W. Case, "Visualization and analysis of student enrollment patterns in foundational engineering courses," *Int. J. Eng. Educ.*, vol. 35, no. 1A, 2019.
[15] S. Pisano, H. Ma, B. Fulgham, G. Guadagni, D. D. Morris, and M. Abramenko, "Redesigning the Calculus Curriculum for Engineering Students," presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/redesigning-the-calculus-curriculum-for-engineering-students
[16] R. Finfrock and N. W. Klingbeil, "Examining the Impacts of the Wright State Model for Engineering Mathematics Education through Curricular Analytics," in *2023 ASEE Annual Conference & Exposition*, 2023.
[17] S. M. Lord *et al.*, "MIDFIELD: A Resource for Longitudinal Student Record Research," *IEEE Trans. Educ.*, vol. 65, no. 3, pp. 245–256, Aug. 2022, doi: 10.1109/TE.2021.3137086.

[18] OpenAI *et al.*, "GPT-4 Technical Report." arXiv, Dec. 18, 2023. doi: 10.48550/arXiv.2303.08774.
[19] S. M. Lord and J. C. Chen, "Curriculum Design in the Middle Years," in *Cambridge Handbook of Engineering Education Research*, A. Johri and B. M. Olds, Eds., Cambridge: Cambridge University Press, 2014, pp. 181–200. doi: 10.1017/CBO9781139013451.014.
[20] A. M. DeRocchis, L. E. Boucheron, M. Garcia, and S. J. Stochaj, "Curricular Complexity of Student Schedules Compared to a Canonical Degree Roadmap," in *2021 IEEE Frontiers in Education Conference (FIE)*, Oct. 2021, pp. 1–5. doi: 10.1109/FIE49875.2021.9637443.
[21] K. Avrithi, "On the 'Introduction to Engineering' Course," presented at the 2019 ASEE Annual Conference & Exposition, Jun. 2019. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/on-the-introduction-to-engineering-course
[22] M. Hoit and M. Ohland, "The Impact of a Discipline-Based Introduction to Engineering Course on Improving Retention," *J. Eng. Educ.*, vol. 87, no. 1, pp. 79–85, 1998, doi: 10.1002/j.2168-9830.1998.tb00325.x.
[23] M. K. Orr, C. E. Brawner, M. W. Ohland, and R. A. Layton, "The Effect of Required Introduction to Engineering Courses on Retention and Major Selection," presented at the 2013 ASEE Annual Conference & Exposition, Jun. 2013, p. 23.1192.1-23.1192.9. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/the-effect-of-required-introduction-to-engineering-courses-on-retention-and-major-selection
[24] G. J. Mazzaro, K. Skenes, and T. A. Wood, "A Review of Multi-Disciplinary Introduction-to-Engineering Courses and Unified-First-Year Engineering Programs," presented at the ASEE Southeast Section Conference, Mar. 2023. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/a-review-of-multi-disciplinary-introduction-to-engineering-courses-and-unified-first-year-engineering-programs
[25] M. W. Ohland, A. G. Yuhasz, and B. L. Sill, "Identifying and Removing a Calculus Prerequisite as a Bottleneck in Clemson's General Engineering Curriculum," *J. Eng. Educ.*, vol. 93, no. 3, pp. 253–257, 2004, doi: 10.1002/j.2168-9830.2004.tb00812.x.
[26] B. Bowen, R. Hall, and J. Ernst, "Calculus eligibility as an at-risk predictor for degree completion in undergraduate engineering," *Technol. Interface Int. J.*, vol. 18, no. 1, pp. 74–80.
[27] K. K. Inkelas, J. L. Maeng, A. L. Williams, and J. S. Jones, "Another form of undermatching? A mixed-methods examination of first-year engineering students' calculus placement.," *J. Eng. Educ.*, vol. 110, no. 3, pp. 594–615, Jul. 2021.
[28] T. D. Ennis, J. F. Sullivan, B. Louie, and D. Knight, "Unlocking the Gate to Calculus Success: Pre-Calculus for Engineers - An Assertive Approach to Readying Underprepared Students," presented at the 2013 ASEE Annual Conference & Exposition, Jun. 2013, p. 23.1285.1-23.1285.23. Accessed: Feb. 06, 2024. [Online]. Available: https://peer.asee.org/unlocking-the-gate-to-calculus-success-pre-calculus-for-engineers-an-assertive-approach-to-readying-underprepared-students
[29] J. Beckert, "Institutional Isomorphism Revisited: Convergence and Divergence in Institutional Change*," *Sociol. Theory*, vol. 28, no. 2, pp. 150–166, 2010, doi: 10.1111/j.1467-9558.2010.01369.x.
[30] B. Faulkner, N. Johnson-Glauch, D. San Choi, and G. L. Herman, "When am I ever going to use this? An investigation of the calculus content of core engineering courses," *J. Eng. Educ.*, vol. 109, no. 3, pp. 402–423, 2020, doi: 10.1002/jee.20344.