Characterizing the Complexity of Curricular Patterns in Engineering Programs

Prof. Gregory L. Heileman, University of New Mexico

Gregory (Greg) L. Heileman serves as the Vice Provost for Teaching and Learning at the University of New Mexico (UNM). From 2011 until 2016, he served as the Associate Provost for Curriculum at UNM. During that time he led campus-wide student academic success initiatives, and worked with key stakeholders on campus, to produce all-time record retention and graduation rates. In 1990 he joined the Department of Electrical and Computer Engineering (ECE) at the University of New Mexico, Albuquerque, NM, where he is currently a Professor. From 2005-2011 he served as Associate Chair (Director of Undergraduate Programs), and led the department through two ABET accreditation visits. In 2011 he became an ABET program evaluator. In 2009 he was also awarded the IEEE Albuquerque Section Outstanding Educator Award. He was the recipient of ECE's Lawton-Ellis Award for combined excellence in teaching, research, and student/community involvement in 2001 and again 2009. He held ECE's Gardner Zemke Professorship from 2005-08. He received the School of Engineering's Teaching Excellence award in 1995, and the ECE Department Distinguished Teacher Award in 2000. During 1998 he held a research fellowship at the Universidad Carlos III de Madrid, and in 2005 he held a similar position at the Universidad Politénica de Madrid. He earned the BA degree from Wake Forest University in 1982, the MS degree in Biomedical Engineering and Mathematics from the University of North Carolina-Chapel Hill in 1986, and the PhD degree in Computer Engineering from the University of Central Florida in 1989.

Mr. Michael Hickman

Michael Hickman received his BS in Computer Engineering from the University of New Mexico in 2014 and is now pursing his MS degree. He is currently working for UNM's Institute of Design & Innovation to provide the University with tools and analytics that can aid in improving student success.

Mr. Ahmad Slim
Prof. Chaouki T. Abdallah, University of New Mexico

Chaouki T. Abdallah started his college education at the Ecole Supérieure d'Ingénieurs de Beyrouth - Université Saint-Joseph in Beirut, Lebanon, but finished his undergraduate studies at Youngstown State University, with a Bachelors of Engineering degree in Electrical Engineering in 1981. He then obtained his MS and Ph.D. in Electrical Engineering from GA Tech in 1982, and 1988 respectively. He joined the Electrical and Computer Engineering department at the University of New Mexico where he is currently professor and since 2011, the provost & EVP for academic affairs, and acting president since January 2017. Professor Abdallah conducts research and teaches courses in the general area of systems theory with focus on control and communications systems. His research has been funded by national funding agencies, national laboratories, and by various companies. He has also been active in designing and implementing various international graduate programs with Latin American and European countries. He was a co-founder in 1990 of the ISTEC consortium, which currently includes more than 150 universities in the US, Spain, and Latin America. He has published 7 books, and more than 300 peer- reviewed papers. His PhD students hold academic positions in the USA and in Europe, and senior technical positions in various US National Laboratories.

Professor Abdallah is a senior member of IEEE and a recipient of the IEEE Millennium medal. He is also active in the IEEE Control Systems Society most recently serving as the general chair of the 2008 Conference of Decision and Control CDC 2008.

Characterizing the Complexity of Curricular Patterns in Engineering Programs

Gregory L. Heileman, Ahmad Slim, Michael Hickman and Chaouki T. Abdallah
{heileman,ahslim,mshickman91,chaouki}@unm.edu
Department of Electrical & Computer Engineering
University of New Mexico

Abstract

Engineering programs tend to follow common patterns for educating undergraduate students through the sophomore year. For instance, a portion of a common curricular pattern for electrical engineering involves the sequence: Calculus I → Calculus II → Differential Equations → Circuits I. In mechanical engineering programs the common curricular pattern includes the sequence: Calculus I → Calculus II → Differential Equations → Mechanics. The curricular patterns themselves are more complicated than these sequences, often involving additional pre- and co-requisite courses that must be passed in order for a student to progress through the curriculum. These patterns may be modeled as directed graphs, and the complexity of the pattern can then be characterized according to the delay and blocking factors present in the graphs. The key point is that failure to pass a course that occurs earlier in a curricular pattern, or the inability to start the pattern on schedule (e.g., due to math placement issues) will often necessitate a delay in graduation. Because these engineering curricular patterns are complex, they tend to produce a longer time-to-degree than other disciplines.

A number of schools have implemented engineering curricular reforms that are aimed at improving on-time graduation rates. These generally involve modifying the patterns described above in some way that is meant to improve student success. We can think of curricular design patterns as being constructed so as to yield a particular set of student learning outcomes. In this paper we apply curricular analytics techniques to these patterns in order to quantify the extent to which particular reforms should improve graduation rates. Our work involves breaking curricular complexity into two components: (1) the structural complexity, which is determined by the manner in which the courses in a curriculum are organized, e.g., prerequisites, number of courses, etc., and (2) the instructional complexity, which is determined by the inherent difficulty of the courses in the curriculum, the quality of the faculty and academic support, etc. We then demonstrate how these measures can be used within a simulation environment to estimate the impact that particular curricular improvements will have on student outcomes. This will reveal that many engineering curricula have highly "sensitive" course patterns (and in some cases individual courses) that will yield large increases in graduation rates for small improvements in course success rates. Finally, we demonstrate how curricular analytics can be used to compare the complexities of similar programs at different institutions, as well as how these tools can be used to guide faculty discussions around curricular reform.

Introduction

Changing the culture of an educational institution or program is one of the most difficult challenges academic leaders face. This is due to the fact that organizational culture is shaped by an interlocking set of goals, processes, traditions, values, roles, assumptions and attitudes that evolve over many years. In the case of higher education, there are cultural elements that have been shaped over a thousand year history. Nevertheless, engineering programs are under mounting pressure to change in response to numerous external forces. Foremost among these is the fact that engineering programs are not producing a sufficient number of graduates to fill the expanding job market for engineering talent that is the cornerstone of the rapidly emerging information economy. The growing diversity of our population leads to additional challenges. Historically, many engineering disciplines have underrepresentation from ethnic minorities, women and the lower socioeconomic strata of society. That is, the most readily available human resources for addressing the need are from the same segments we have struggled the most to serve. Our ability to adapt to these challenges has significant consequences for national competitiveness and economic leadership going forward.

In response to issues just described, we have seen numerous well-intended studies, funding opportunities and initiatives aimed at transforming engineering education.[5,6,11] Furthermore, institutions across the country are engaged in improvement efforts, with varying degrees of success. In many cases these efforts are based upon intuition, or they attempt to replicate interventions that have been reported as successful elsewhere. It is often difficult to determine the reasons for success or failure, or why something that worked at one institution did not work at another.

In our work, we have been proponents of the importance of recognizing that reforms occur within a larger educational context that must be properly understood and characterized if one hopes to optimize the impact of particular interventions. We treat the university as a system, comprised of a set of sub-components that interact in order to create the system as a whole, with each component contributing in some manner, either directly or indirectly, to the success of improvement efforts.[1] An important point to note is that from one university to the next, the properties of the university system will differ. Ideally, before launching improvement initiatives at great effort and expense, one would determine the most important factors that contribute to attrition and persistence, and would use these to construct a model that can be used to accurately predict the expected improvements that can be obtained by implementing specific reforms at particular universities. This requires a formalization of the university system model we have just described.

A difficult aspect of the systems formulation of a university involves determining the proper metrics and measurements that can be used to quantify the various sub-components of the educational system. This is necessary if we hope to analyze the system in order make predictions, quantify the impact of interventions, and focus efforts where they are most likely to succeed. In this paper we present one contribution to this systems way of thinking about educational reform. Specifically, we consider formal ways of characterizing the important patterns that exist within engineering curricula, and of we consider one way of quantifying their complexity. In addition, we argue a direct relationship between the complexity of particular curricular patterns, and the success of students attempting to navigate these patterns.

The remainder of this paper is organized as follows. First we describe the general utility of design patterns, and how they can be applied to the design of program curricula. We provide a formal definition of curricular design patterns, as well as an example patterns within engineering programs. Next, we describe a methodology that can be used to characterize the complexity of curricular design patterns. In doing so, we provide a means for evaluating different design patterns that aim to yield a similar set of student learning outcomes. Finally, we present a technique for quantifying the complexity of curricular design patterns, and we relate this complexity measure to student success outcomes.

Curricular Design Patterns

The notion of *design patterns* originated as a concept in architecture intended to capture the essence of an architectural design in such a way that it could be easily reused in the design of cities and buildings.[2] The power of design patterns is they allow architects to capture, in general terms, design ideas that have proven successful in solving particular design challenges across many different contexts. The context in this case refers to the range of situations in which the problems and solutions addressed by a pattern apply. Thus, design patterns provide a useful collection of knowledge that other architects may consult in the future when confronting similar design challenges. Furthermore, the design patterns themselves constitute a language that architects may use to more efficiently communicate with one another. The authors of this approach summarize it as follows "[e]ach pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

In 1987, software engineers began experimenting with the notion of applying design patterns to the challenges that are routinely confronted in the design of large-scale software systems.[3] This led to the formalization of a large number of software design patterns that have been shown to greatly aid in the development of complex software systems by providing developers with a reusable set of proven solutions to generalized problems.[7] *Software design patterns* are applicable in object-oriented designs, and consist of a description of interacting software objects and classes that solve a general problem within a context. Again, the notion of context is important with these types of design patterns; software design patterns should be general enough to be applicable across a wide range of systems, but specific enough to provide practical guidance. Software design patterns are similar to their architectural counterparts in that they provide software engineers with a language that can be used to discuss software design issues. For instance, a software engineer may comment during a design review that "the observer design pattern was used to update the state of the dependent objects," and all of the other software engineers at the meeting would immediately understand the design choice that was made.

In this paper we consider the extension of design patterns to curricula. Bland and Kumar previously linked design patterns to computer science curricula at a very high level.[4] Specifically, their conception of design patterns addressed broad curricular reforms that might serve to better engage women in computer science, and included practices such as participation in freshman seminars, the humanization of core computer science classes, and flexibility in designing the major. This work is similar to the creation of engaging early experiences, high-impact practices, and "conditions that matter" for student persistence.[9,10] Our formulation, on the other hand, is more granular, and directly ties to quantifiable design objectives that occur over and over again within particular curricular contexts. This makes the approach more consistent with the general manner in which design patterns have been applied in other domains. We are not aware of other efforts that have involved the formal analysis of curricular design patterns at the course and student learning outcomes levels, or of any other efforts that have involved the creation of metrics for quantifying the complexity of curricular design patterns.

The power of our approach is that it gives educators a rich vocabulary for discussing curriculum, and a means of formally analyzing the improvements that might be attained through particular curricular reforms (e.g., what-if analyses). A formal definition for curricular design patterns is as follows.

Definition 1 (Curricular Design Pattern)**.** *A collection 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.*

The typical manner in which learning outcomes are attained in undergraduate engineering programs is through the offering of courses that have have pre- and co-requisite relationship between them. That is, learning activities are structured as courses that must be passed in sequence in order to attain the learning outcomes. The general notion is that prerequisite courses contain learning activities that must be successfully completed in order to understand the learning activities that occur in follow-on courses. In order to make this more concrete, consider the following curricular design pattern.

Example 1 (Traditional Circuits I – Calculus Ready)**.** This curricular design pattern is constructed so as to attain a set of learning outcomes that involve the ability to design, build and analyze simple electronic circuits, under the assumption that a student is prepared for Calculus I. Some of the specific learning outcomes are as follows.
Students will:

1. Understand the functions of basic electrical circuit elements and sources;
2. Have the ability to apply Ohm's and Kirchhoff's circuit laws in the lumped element model of electrical circuits;
3. Appreciate the consequences of linearity, in particular the principle of superposition and Thevenin and Norton equivalent circuits;
4. Understand the concept of state in a dynamical physical system and have the ability to analyze simple first and second order linear circuits containing memory elements.

The educational context we will consider is an electrical engineering program, where the assump-

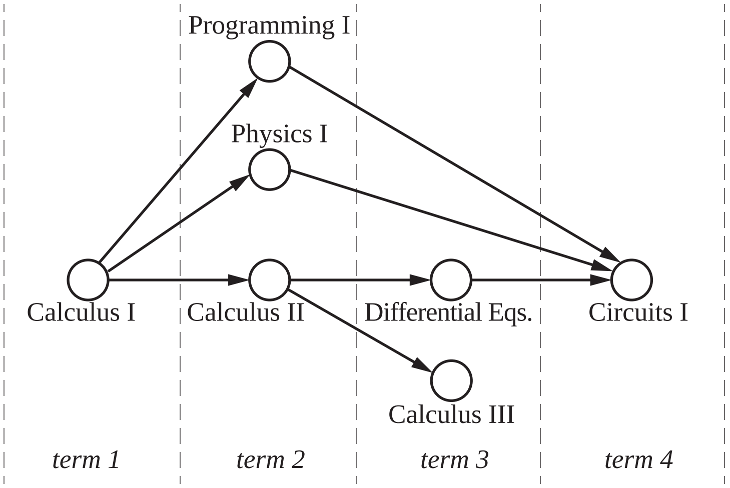


Figure 1: A four-term course sequence associated with the Traditional Circuits I design pattern.

tion is that students begin the program with prior academic preparation that allows them to start in Calculus I.

A curricular pattern of courses designed to allow students to attain the learning outcomes listed above is shown in Figure 1. The pattern is represented as a graph, with nodes representing courses, and directed edges between two courses if one is a prerequisite for the other. The left-to-right ordering of nodes in Figure 1 correspond to terms, as labeled in the figure. In subsequent figures we will omit this labeling, with the understanding that the ordering implies terms. If there is a directed edge between two courses in the same term, then the two courses share a co-requisite relationship.

Remark 1. Due to the prerequisite relationships in Figure 1, a student cannot complete the pattern in less then four terms. This is dictated by a critical path of length four involving Calculus I → Calculus II → Differential Equations → Circuits I. Thus, if students fail to pass any course on this path in the term stipulated, they will automatically be delayed in completing the pattern by at least one term. There are other length-three paths that are not critical in the sense that failure to pass them on the first try will not lead to a delay in completing the pattern. For instance, if students do not pass Programming I in term 2, they will not be delayed as long as they are able to successfully complete this course in term 3.

Remark 2. Only a subset of the learning activities in a given course may be necessary as prerequisite material for the learning activities in follow-on courses. For instance, the fourth learning outcome listed above requires a student to have prior learning that includes the ability to solve first and second order differential equations; however, a traditional Differential Equations course will include the study of a significant number of additional classes of differential equations.

Remark 3. Patterns may overlap. For instance, in the electrical engineering context, this design pattern will have significant overlap with the Traditional Electromagnetics curricular design pattern, as both require mathematics through Differential Equations and Calculus III. Furthermore, an Electromagnetics course will require prior learning that includes the ability to solve partial differential equations. Thus, the Electromagnetics pattern requires prior learning, attained in a Differential

Equations course common to both patterns, that is not required knowledge for this design pattern.

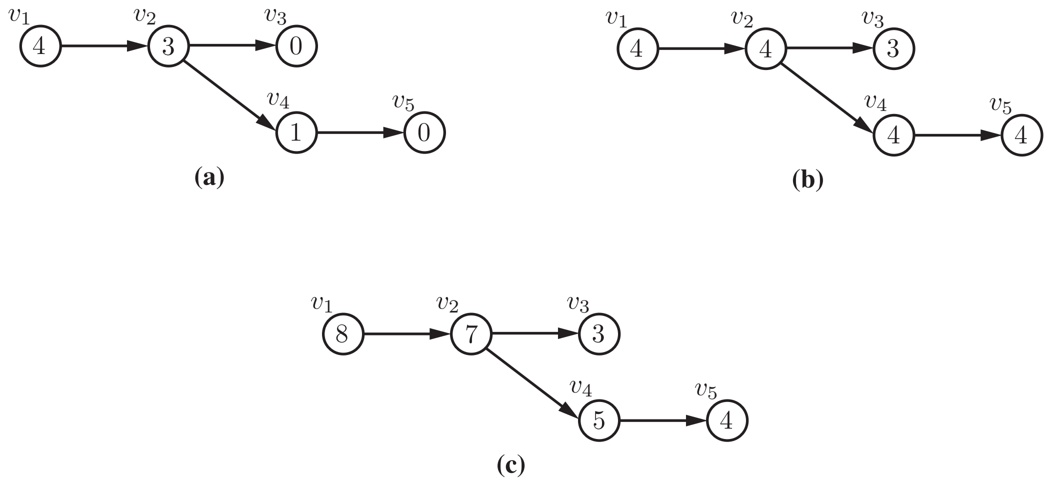
Remark 4. This design pattern is not applicable to students who are not prepared to enroll in Calculus I in term 1. A typical remedy for this situation is to add one or more math classes as prerequisites to Calculus I, thereby creating new curricular design patterns. This automatically extends the number of terms necessary to complete the pattern according to the amount by which the added courses extend the critical path. It is not uncommon to see three or four additional prerequisite courses added to this path, leading to a curricular design pattern that requires seven or eight terms to complete. For these students, if nothing is done to accelerate learning activities, the earliest the sophomore-level Circuits I course can be reached is during the fourth year of study.
□

Now that we have provided a formal means of expressing curricular design patterns, let us investigate how we might apply analytical techniques in order to quantify their complexity. Below we describe one approach that makes use of graph-theoretic techniques to quantify the complexity of a curricular design pattern, as well the complexity of the curriculum it resides in (i.e., its context). It is important to note that these are not the only measures that can be derived for quantifying the complexity of a design pattern. Indeed, we anticipate that the analytical tools for studying curricular design patterns will evolve extensively over time. Nevertheless, the simple formulation we provide has proven useful as a tool for university faculty, administrators and curriculum committees to easily assess the efficiency of their curricula, and to analyze possible changes to programs.

The Complexity of Curricular Design Patterns

We have previously used graph-theoretic concepts to study the complexity of curricula, and it is straightforward to apply these concepts to curricular design patterns.[12,13] A given curricular pattern is represented as a directed acyclic graph by creating a node for each course in the pattern, and by placing a directed edge between two nodes if there is a pre- or co-requisite relationships between the courses the nodes represent. The curricular design pattern shown in Figure 1 is represented as a directed acyclic graph. More formally, we model a curricular pattern C consisting of n courses as a directed graph $G_C = (V, E)$, where each vertex $v_1, \ldots, v_n \in V$ represents a requirement in C, and there is a directed edge $(v_i, v_j) \in E$ from requirement v_i to v_j if v_i must be satisfied prior to the satisfaction of v_j. Typically, a degree requirement is satisfied by passing a particular course, and the precedence relationships expressed in G_C correspond to course pre/co-requisites. Thus, we will refer to G_C as a *curriculum graph.*

Curricular Efficiency Metrics. The graph structure of a curriculum yields two useful metrics. First, the number of courses students are blocked from taking until they pass a given course is referred to as the *blocking factor* of that course. Second, the number of courses on any prerequisite pathway that includes a given course is referred to as the *delay factor* of that course. Figure 2 shows a curricular pattern consisting of five courses, v_1–v_5. Inside of each node in Figure 2 (a), we show the blocking factor of that node, and inside of each node in Figure 2 (b) we show the delay factor associated with that node. We then define the *cruciality* of a course within a curriculum as



(a) **(b)** **(c)**

Figure 2: A portion of a curriculum consisting of five courses, labeled v_1–v_5. Inside of each course node we show, **(a)** the blocking factor associated with the course, **(b)** the delay factor associated with the course, and **(c)** the cruciality of the course.

the sum of its delay and blocking factors. Figure 2 (c) shows the course crucialities of each course in the curricular pattern. We define the *structural complexity* of a curricular pattern as the sum of the course crucialities in the pattern. For instance, the curricular pattern shown in Figure 2 has a structural complexity of 27. The reader can easily verify that the structural complexity of the Traditional Circuits I curricular pattern shown in Figure 1 is 37.

We may easily extend the notion of structural complexity to an entire program curriculum. We will refer to the structural complexity of a curricular pattern and to the structural complexity of an entire program as, the *pattern complexity* and *program complexity*, respectively. It is important to recognize that the same curricular pattern, placed within two different program curricula, will generally produce different program complexities. We have built an open access web application that allows any institution to upload and analyze curricular patterns within the context of their particular degree programs using the tools described above. In order to use these tools, please visit: `http://curricula.academicdashboards.org`. Next, we will demonstrate in more details how the complexity measure we have just defined can be used to analyze curricular patterns. The following design pattern is one that many aspiring engineering students are asked to follow.

Example 2 (Traditional Circuits I – Not Calculus Ready)**.** This design is commonly found in engineering programs that include electrical circuits. The assumption in this case, however, is that students are not prepared for Calculus. The common remedy is to simply prepend a Precalculus course to the pattern provided in Example 1. The learning outcomes are the same as those provided in Example 1. The design pattern itself is shown in Figure 3, with the course crucialities shown inside of each node.

Remark 1. The math starting point in this pattern, Precalculus, is actually an extremely common staring point for many freshman, although it rarely appears on the degrees plans provided to these students. At our institution, the majority of freshman engineerings students initially place into Precalculus.

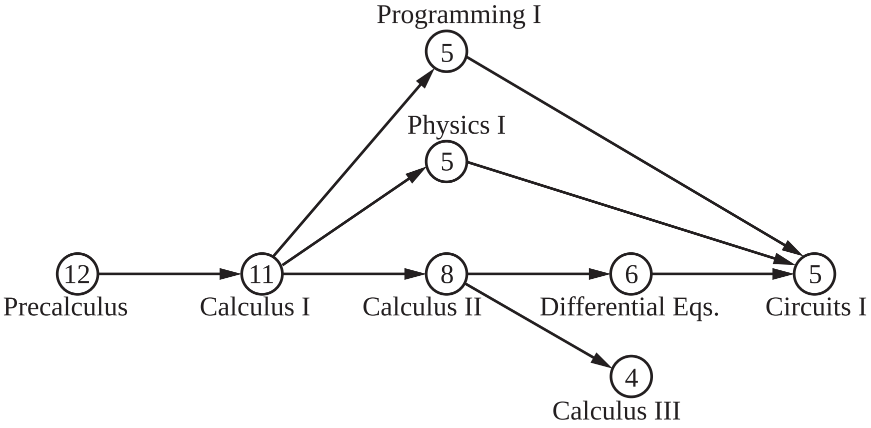


Figure 3: The five-term curricular pattern associated with Circuits I, starting with Pre-Calculus math, with course crucialities shown inside each node. The structural complexity of this subgraph is 56.

Remark 2. Because of where Precalculus occurs in this pattern, it becomes the most crucial course in the pattern. Its addition causes the complexity of the pattern as whole to increase by 19 complexity units (as compared to the pattern in Example 1), to a value of 56. We will later comment on the relationship between complexity and student success.

Remark 3. Circuits I is no longer taken in the sophomore year in this pattern, it moves to the junior year. Given the other courses in some curricula, e.g., electrical engineering, that build upon Circuits I, this shift makes it more difficult to create a 4-year degree plan that includes this pattern. □

We previously alluded to the fact that each course in a curricular pattern contributes learning outcomes that are necessary in order to progress through other courses in the pattern, and that some courses provide additional learning outcomes that are not needed to complete the pattern (though they may be required in other parts of the curriculum). Thus, as long as all of the required learning outcomes needed to attempt the Circuits I class occur in prerequisite courses, we can rearrange the order in which these outcomes are attained. This notion is exploited in the next pattern.

Example 3 (Alternative Circuits I)**.** This design pattern, shown in Figure 4, involves the introduction of a new course entitled Engineering 101 that takes the place of Precalculus in the previous design pattern. This course is modeled after the one created at Wright State University, and the design pattern itself models the curricular reforms implemented at that institution.[8] Notice that Engineering 101 now becomes the sole prerequisite for the Circuits I class. The learning outcomes are the same as those provided in Examples 1 and 2.

Remark 1. The learning outcomes for this design pattern include the ability analyze first and second order linear circuits containing memory elements as a part of the Circuit I class. Thus, in addition to Precalculus topics, the ability to solve first and second order linear differential equations must be taught in the Engineering 101 class.

Remark 2. The math course sequencing starting with Calculus I is unchanged relative to the previ-

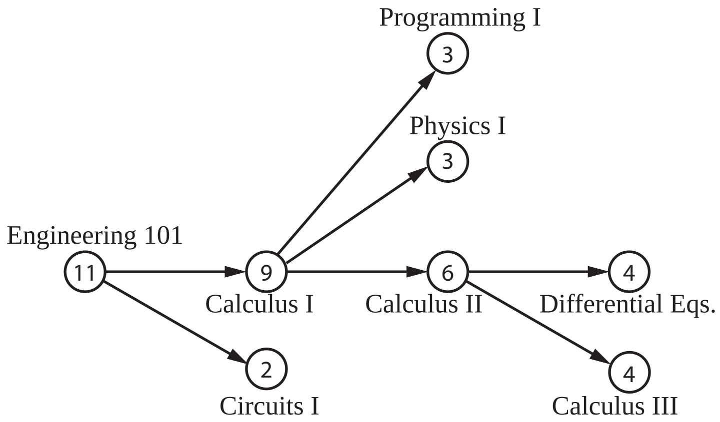


Figure 4: A four-term course sequence for the Alternative Circuits I design pattern. The structural complexity of this subgraph is 41.

ously presented patterns. That is, even though students are exposed to linear differential equations in Engineering 101, they will see these concepts again when they take the Differential Equations class. The notion of providing exposure to a math topic, using that topic in the context of an engineering problem, followed by a more rigorous treatment of the math topic, is a very effective learning strategy for many students.

Remark 3. For the same level of initial math preparation, this design pattern aims to produce the same learning outcomes as Example 2, with a significant reduction in overall pattern complexity. We will demonstrate, under some very simple assumptions, that this significantly increases the chances that a student will complete this pattern. Furthermore, this pattern may be completed in four terms, while the pattern in Example 2 requires five terms. Students who follow this pattern will be exposed to important engineering concepts in the second term, rather than the fifth term when following the traditional approach. This earlier exposure may impact a student's perception of the degree program as a whole.

Remark 4. As compared to students who enter college calculus ready, and who follow the pattern in Example 1, students who follow this pattern have been shown to reach a similar (if not improved) level of outcomes attainment within a larger engineering curriculum.[8] Furthermore, this design pattern can be completed in the same amount of time, putting these students on equal footing with those who enter college calculus ready.
□

Discussion

We have shown that curricular design patterns are subgraphs that occur within the curriculum graphs of degree programs, and that these patterns may be reorganized in ways that affect their curricular complexity. Based upon this formulation, there are a number of interesting questions that immediately arise and merit further investigation.

First, the formal definition of curricular design patterns we have provided supports the ability to search through collections of curriculum graphs in order to find common patterns. We content that the creation of a catalog of common curricular design patterns will facilitate curricular reform by grounding discussion within a well-understand formal framework.

Second, the notion that different learners may be better served by different design patterns should be further investigated. Specifically, to what extent does the organization of the design patterns within a curriculum influence student success for particular student cohorts? Preliminary data indicates that there may be a "pattern effect" for learners who enter higher education with particular background preparation.[8]

It is important to note that the structural complexity of a curriculum is just one component that influences a student's ability to progress through the curriculum. We refer to the other major component as the *instructional complexity*. This refers to the inherent difficulty of the courses in the curriculum, the capabilities of the instructors who teach these courses, the overall quality of the instruction, the academic support services provided, etc. That is, the curricular complexity is determined by the interplay between the structural and instructional complexity of a curriculum.

A simple model for instructional complexity demonstrates the usefulness of this characterization of curricular complexity. Let us assume that an institution currently supports engineering curricula that contain the Traditional Circuits I – Not Calculus Ready design pattern (Example 2), and that this institution is considering implementing the Alternative Circuits I design pattern (Example 3). The institution would like to estimate the possible improvement in student success outcomes that may accompany such a reform. This can be accomplished via simulation under a few realistic assumptions. First, assume that the institutional complexity of all courses that are common between these two patterns are the same. That is, assume that nothing changes in how these courses are taught and supported when moving to the new curriculum. Next let us assume that if students are unable to complete a given course in either sequence, they will attempt to complete the course in the next semester, but that only three courses from a given pattern may be attempted in a given term. Under these assumptions, if the pass rate for all courses in both patterns is 80%, then the success rate of students after seven terms in the Traditional versus Alternative Circuits I patterns are 83% and 93%, respectively.[1] That is, the alternative pattern leads to an 10% improvement in student completion.

When simulating student flow through these patterns using the historic pass rates for all of the courses in these two curricula at our institution, and assuming the pass rate for Engineering 101 is the same as Precalculus, we found that 79% and 88% of the students are able to succeed after seven terms in the Traditional versus Alternative Circuits I patterns, respectively. It should be noted, however, that the general intention in offering an Engineering 101 courses is to create improved success relative to Precalculus. Under the assumption that the pass rate of Engineering 101 improves to 90%, the success rate of students following the alternative pattern improves to 98%.

Finally, we comment on the usefulness of curricular analytics as a means of informing discussions related to curriculum changes. At many institutions, undergraduate engineering programs require more total credit hours than academic programs in other disciplines. Given the completion imperative discussed in the Introduction, engineering programs are being asked to carefully justify the

---

[1] Success being the passing of all courses in the pattern.

need for requiring credit hours beyond minimal institutional and accreditation requirements, often at the behest of state legislatures and institutional governing boards. Engineering faculty, however, are often reluctant to reduce degree requirements if they believe this action will reduce program quality. The analytical tools we present in this paper have proved useful in guiding discussions around curriculum change on our campus, as they provide a tool for comparing the complexities of similar programs at different institutions, including those that are highly rated, and for considering the possible improvements in graduation rates that could be obtained through particular curricular reforms.

References

[1] C. Abdallah, G. Heileman, and T. Babbitt. The university is a system. *The EvoLLLution*, Jan. 29, 2016.

[2] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, 1977.

[3] K. Beck and W. Cunningham. Using pattern languages for object oriented programs. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, 1987.

[4] D. Bland and D. Kumar. Patterns of curriculum design. In L. Cassel and R. Reis, editors, *IFIP Advances in Information and Communication Technology: Informatics Curricula and Teaching Methods*. Springer, 2003.

[5] J. J. Duderstadt. Engineering for a changing world: A roadmap to the future of engineering practice, research, and education. Technical report, The Millennium Project, The University of Michigan, 2008.

[6] P. D. Galloway. *The 21st-Century Engineer: A Proposal for Engineering Education Reform*. ASCE Press, Reston, VA, 2013.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Boston, MA, 1995.

[8] N. W. Klingbeil and A. Bourne. The Wright State model for engineering mathematics education: Longitudinal impact on initially underprepared students. In *Proceedings of the 122nd ASEE Annual Conference*, Seattle, WA, June 14–17, 2015.

[9] G. D. Kuh. High-impact educational practices: What they are, who has access to them, and why they matter. Technical report, Association of American Colleges and Universities, Washington, DC, 2008.

[10] G. D. Kuh, J. Kinzie, J. H. Schuh, and E. J. Whitt. *Student Success in College: Creating Conditions That Matter*. Jossey-Bass, San Francisco, CA, 2010.

[11] National Science Foundation Program Solicitation, IUSE/Professional Formation of Engineers: Revolutionizing Engineering and Computer Science Departments (RED), NSF 17-501, 2017. `https://www.nsf.gov/pubs/2017/nsf17501/nsf17501.htm#toc`.

[12] A. Slim. *Curricular Analytics in Higher Education*. PhD thesis, University of New Mexico, 2016.

[13] J. Wigdahl, G. L. Heileman, A. Slim, and C. T. Abdallah. Curricular efficiency: What role does it play in student success? In *2014 ASEE Annual Conference & Exposition*, Indianapolis, Indiana, June 2014. ASEE Conferences. `https://peer.asee.org/20235`.