Enhancing Academic Pathways: A Data-Driven Approach to Reducing Curriculum Complexity and Improving Graduation Rates in Higher Education

Dr. Ahmad Slim, The University of Arizona

Dr. Ahmad Slim is a PostDoc researcher at the University of Arizona, where he specializes in educational data mining and machine learning. With a Ph.D. in Computer Engineering from the University of New Mexico, he leads initiatives to develop analytics solutions that support strategic decision-making in academic and administrative domains. His work includes the creation of predictive models and data visualization tools that aim to improve student recruitment, retention, and success metrics. Dr. Slim's scholarly contributions include numerous articles on the application of data science in enhancing educational practices.

Prof. Gregory L. Heileman, The University of Arizona

Gregory (Greg) L. Heileman currently serves as the Associate Vice Provost for Academic Administration and Professor of Electrical and Computer Engineering at the University of Arizona, where he is responsible for facilitating collaboration across campus t

Husain Al Yusuf, The University of Arizona

Husain Al Yusuf is a third-year PhD candidate in the Electrical and Computer Engineering Department at the University of Arizona. He is currently pursuing his PhD with a research focus on applying machine learning and data analytics to higher education, aiming to enhance student outcomes and optimize educational processes.

Husain Al Yusuf holds an M.Sc in Computer Engineering from the University of New Mexico and brings over fifteen years of professional experience as a technology engineer, including significant roles in cloud computing and infrastructure development at a big technologies company and financial services industry.

Dr. Yiming Zhang, The University of Arizona

Yiming Zhang completed his doctoral degree in Electrical and Computer Engineering from the University of Arizona in 2023. His research focuses on machine learning, data analytics, and optimization in the application of higher education.

Asma Wasfi
Mohammad Hayajneh
Bisni Fahad Mon, United Arab Emirates University
Ameer Slim, University of New Mexico

Enhancing Academic Pathways: A Data-Driven Approach to Reducing Curriculum Complexity and Improving Graduation Rates in Higher Education

Ahmad Slim[†], Gregory L. Heileman[†], Husain Al Yusuf[†],

Ameer Slim[‡], Yiming Zhang[†], Mohammad Hayajneh[•],
Bisni Fahad Mon[•], Asma Wasfi Fayes[•]
{ahslim@arizona.edu, heileman@arizona.edu, halyusuf@arizona.edu,
ahs1993@unm.edu, yimingzhang1@arizona.edu, mhayajneh@uaeu.ac.ae,
bisni.f@uaeu.ac.ae, 201180954@uaeu.ac.ae}
[†]The University of Arizona
[‡]The University of New Mexico
[•]United Arab Emirates University

Abstract

Curriculum structure and prerequisite complexity significantly influence student progression and graduation rates. Thus, efforts to find suitable measures to reduce curriculum complexity have recently been employed to the utmost. Most of these efforts use the services of domain experts, such as faculty and student affairs staff. However, it is tedious for a domain expert to study and analyze a full curriculum in an attempt to reform its structure, given all the complexities associated with its prerequisite dependencies and learning outcomes. Things can become even more complicated when a set of curricula is examined. Therefore, efforts to automate the process of restructuring curricula are beneficial to helping the university community find the best available practices to reduce the complexity of their institutional curricula. This study introduces an innovative framework for automating curriculum restructuring, employing a combination of graphical models and machine learning techniques. In particular, we use latent tree graphical models and collaborative filtering to induce curriculum reforms without needing a domain expert. The approach used in this paper is data-driven, where actual student data and actual university curricula are utilized. Five thousand seventy-three student records from the University of New Mexico (UNM) are used for this purpose. Results demonstrate the restructuring impact on an engineering curriculum, particularly the computer engineering program at UNM. The effect is an improvement in the graduation rates of the students attending the revised engineering programs. These results are validated using a Markov Decision Processes (MDP) model. Furthermore, the findings of this paper showcase the practical benefits of our approach and offer valuable insight for future advancements in curriculum restructuring methodologies.

keywords: curricular complexity, Markov decision processes, collaborative filtering, latent tree graphical models, student success, graduation rates, educational data mining

1 Introduction

In our study, we explore analytics focusing on a crucial aspect of student success: the curriculum pathways that lead students toward achieving their learning outcomes and ultimately earning their degrees. In the realm of higher education, the role of analytics is increasingly recognized as a tool for decision-making that enhances student success outcomes. For example, various initiatives have used student demographics and prior academic performance to guide interventions such as counseling, mentoring, and tutoring to improve retention and graduation rates[1,2,3]. Our perspective emphasizes that the core of student academic success lies in progression within a curriculum. Obstacles in curricular pathways can delay graduation and increase the likelihood of students discontinuing their education. Thus, examining these interventions in the context of their direct effect on degree progression is crucial. Our approach to studying student success takes a reductionist stance, similar to how natural sciences interpret complex biological phenomena through underlying chemical and physical principles. However, this approach faces several challenges. One significant difficulty is quantifying the impact of specific interventions or reforms on a student's progress within their degree program. An example of this complexity is the assessment of the effect of an internship program on student progression across various academic programs. This problem is often compounded by a need for coordination between those implementing interventions and those responsible for curriculum design, leading to a disconnect in understanding the curriculum and its intended learning outcomes. This scenario exemplifies the challenges arising from operational "silos" within educational institutions. Furthermore, the prevalent shared governance model in universities can further strengthen these silos. A common belief in academic circles is that "faculty own the curriculum," which can be interpreted as a directive to avoid interfering with faculty governance. Despite this, faculty are generally more receptive to curricular changes that are supported by data demonstrating benefits to student success, as opposed to changes imposed from the top down. These curricular or pedagogical experiments, often inspired by successful implementations at other institutions, can sometimes resemble uninformed attempts, with results frequently based on anecdotal evidence rather than concrete data. Identifying critical courses within curricula is another challenge, with some courses perceived as more crucial than others for various reasons, such as being strong predictors of overall academic success or acting as foundational prerequisites. The complexity of a curriculum correlates with the number of these significant courses. This paper summarizes previous work used to quantify the concept of essential courses and characterize the overall complexity of a curriculum [4]. We propose an analytical framework to measure the impact of curricular and pedagogical interventions on student progression. This framework has been instrumental in fostering data-driven discussions with faculty and curriculum committees, reducing the influence of subjective opinions in reform debates, and encouraging consensus on proposed changes. By establishing meaningful metrics related to curriculum structure, we enable effective comparisons and informed decisions about curriculum reforms. Moreover, our framework provides a model for predicting the impacts of various curricular and pedagogical reforms within a specific educational environment, supporting a more structured approach to planning and evaluating student success strategies. The potential of curricular analytics lies in directly linking interventions to student success outcomes, acknowledging the importance of understanding the larger educational context to maximize the effectiveness of interventions. We view the university as a complex system comprising interacting subcomponents that collectively influence the success of improvement

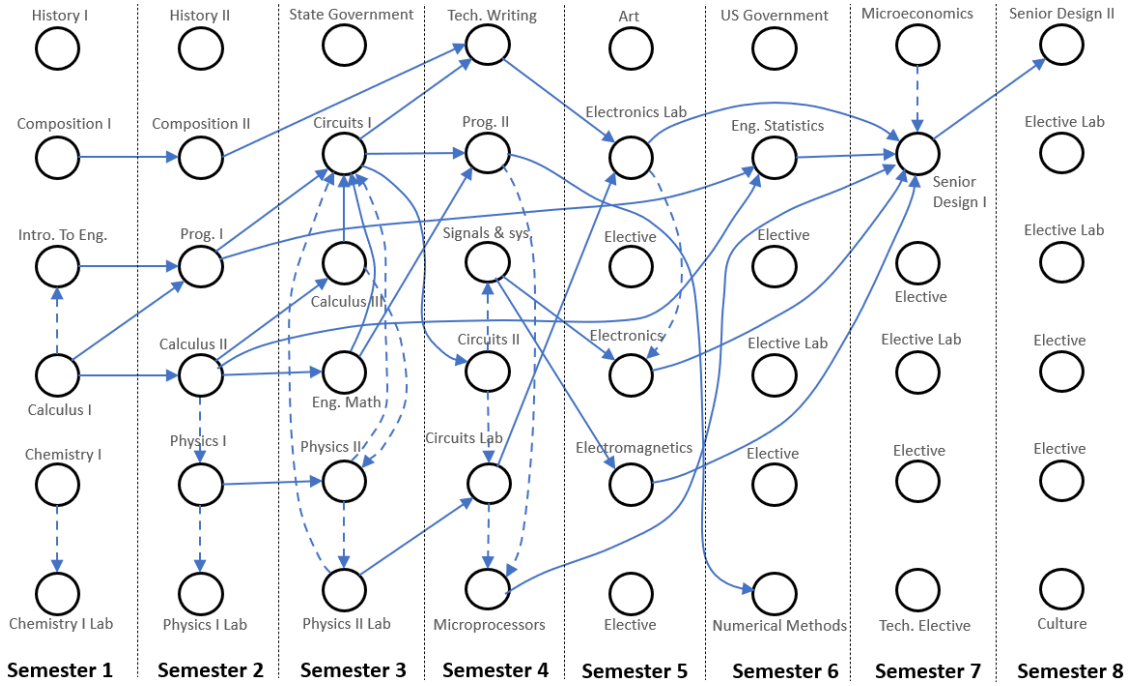
efforts[5,6]. Each university's system properties vary, necessitating tailored models to predict improvements from specific reforms. In this paper, we compile recent developments in curricular analytics, organizing them to support practical applications and further theoretical advances in this field[4,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]. Our significant contribution, however, is a new model aimed at automating the process of curricular interventions. We introduce a framework that streamlines curricula restructuring without needing domain expert intervention, addressing the biases and challenges inherent in traditional expert-driven reform processes. This model, incorporating a Latent Tree Model (LTM) enhanced with machine learning and graph theory techniques, offers a more efficient and unbiased approach. We validate this framework using real student data and various curricular metrics proposed in this study.

2 Theoretical Foundations of Curricular Analytics

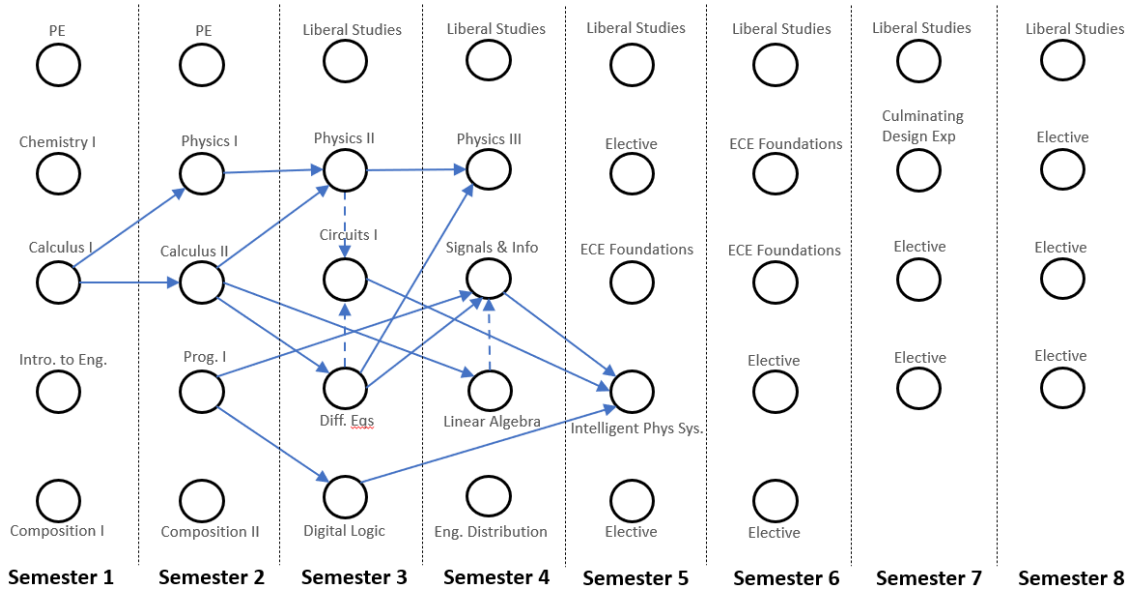
Building on our earlier discussion, the examination of academic curricula is significantly enhanced by the widespread practice of publishing these curricula on public platforms. This transparency allows academic programs to benchmark their curricula against those offered by comparable institutions. For example, as depicted in Figure 1, we examine the undergraduate electrical engineering curricula of two major public U.S. institutions, both accredited by ABET[22]. These curricula are structured into four-year (eight-term) plans, guiding students through their degree completion. We represent these curricula as graphical models, with vertices symbolizing courses and directed edges indicating prerequisite requirements. Specifically, a directed edge from one course (vertex) to another mandates that the former, as a prerequisite, must be completed before the latter. In cases where a directed edge connects two courses within the same term, it signifies a co-requisite relationship, allowing for concurrent or sequential enrollment. Courses obligatory to be taken together within the same term are labeled as strict co-requisites, where the edge's direction is inconsequential. An intriguing observation from Figure 1 is that despite sharing identical ABET accreditation and fulfilling the same set of eleven ABET program learning outcomes, the structural compositions of the two programs are notably different. This disparity raises several pertinent analytical questions. For example, how does this structural difference affect the expected graduation rates of students with similar preparedness in each program? Additionally, one might wonder about the most influential course in each curriculum and the potential impact on student success rates if these key courses were slightly improved. A critical inquiry is whether one program offers superior preparation for students in their chosen field compared to the other. The following sections dive into a detailed framework and toolkit designed for curriculum designers. This toolkit enables a thorough exploration and informed answers to these questions under reasonable assumptions. It provides a means to quantify and analyze the disparities between curricula, such as those illustrated in Figure 1a and Figure 1b, thus offering a systematic approach to improve curriculum design based on data-driven insights.

3 Analytical Framework for Curriculum Assessment

Expanding upon our previous discussions on curricular analytics, we examine the nuanced challenge of analyzing the impact of curricula on student progression. This analysis is particularly complex due to the multifaceted nature of curriculum-related components influencing student progress. Our methodology focuses on decomposing the overall complexity of a curriculum into



(a)



(b)

Figure 1: Undergraduate Electrical Engineering program structures at two major public universities with the same ABET accreditation standards.

two primary elements: instructional complexity, which refers to the pedagogical methods and support mechanisms employed in teaching courses, and structural complexity, which relates to the organizational framework of the curriculum itself. We have previously explored structural complexity by examining the prerequisite relationships between courses within a curriculum[4]. Our graph-theoretic approach, primarily based on analyzing total path lengths, aimed to understand how much one course can hinder or delay a student's progression to subsequent courses. This line of analysis led to the development of a ranking system for courses within a course network, categorized by their criticality level.

3.1 A Framework for Analyzing Course Network Structures

The criticality of a course in a network hinges on two pivotal factors: its delay factor and its blocking factor. These elements are further defined by two parameters: the longest path and connectivity. The longest path, denoted as L_i for a node i, is the length of the longest path that passes through that node. The connectivity, V_i, of node i represents the total number of nodes connected to i. The formula to determine the connectivity is:

$$V_i = \sum_j n_{ij}$$

where n_{ij} is 1 if a path exists from i to j and 0 otherwise.

3.1.1 Delay Factor

Many STEM curricula have a sequence of courses that must be completed in a specific order. This sequence often progresses through foundational mathematics courses, each building upon the previous. The ability to complete these pathways without delay is crucial to successful graduation. A delay in any single course within this sequence can result in a domino effect, which delays the entire pathway. We define the delay factor of a course i, L_i, as the number of vertices in the curriculum's longest path that includes course i. Figures 2a and 2c demonstrate this concept using simplified curricular models, where the delay factor for each course is indicated.

3.1.2 Blocking Factor

A course can also act as a structural bottleneck, serving as a prerequisite for several other courses. The failure to pass this 'gateway' course can prevent students from progressing through the curriculum. The blocking factor of such a course is significant, as it holds a pivotal position within the curriculum. Figures 2b and 2d depict the blocking factors for the courses in the illustrated curricula.

Combining these analyses, we introduce a metric to define the cruciality of a course i, denoted C_i, as the aggregate of its blocking and delay factors:

$$C_i = V_i + L_i$$

The curriculum's overall complexity, S, is then calculated as the sum of the cruciality values for all courses:

$$S = \sum_1^m C_i \tag{1}$$

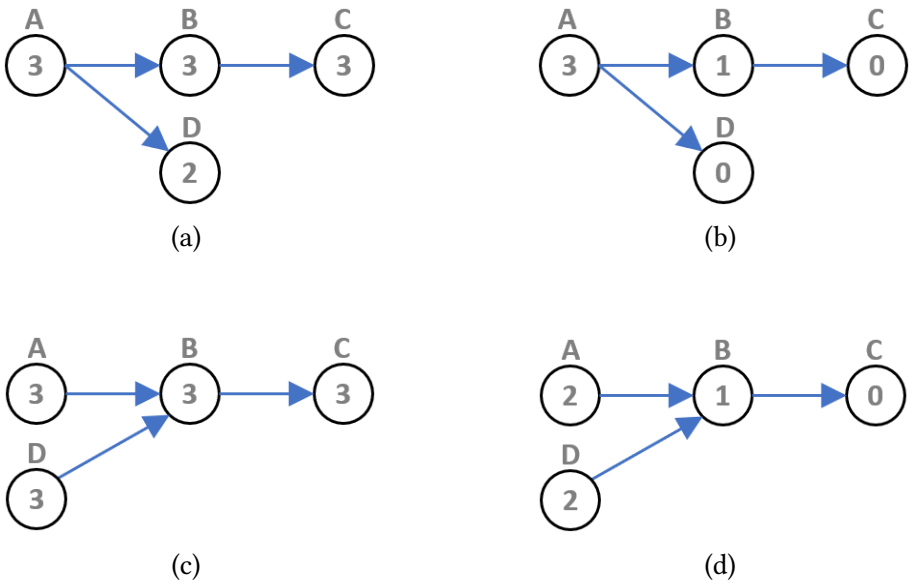


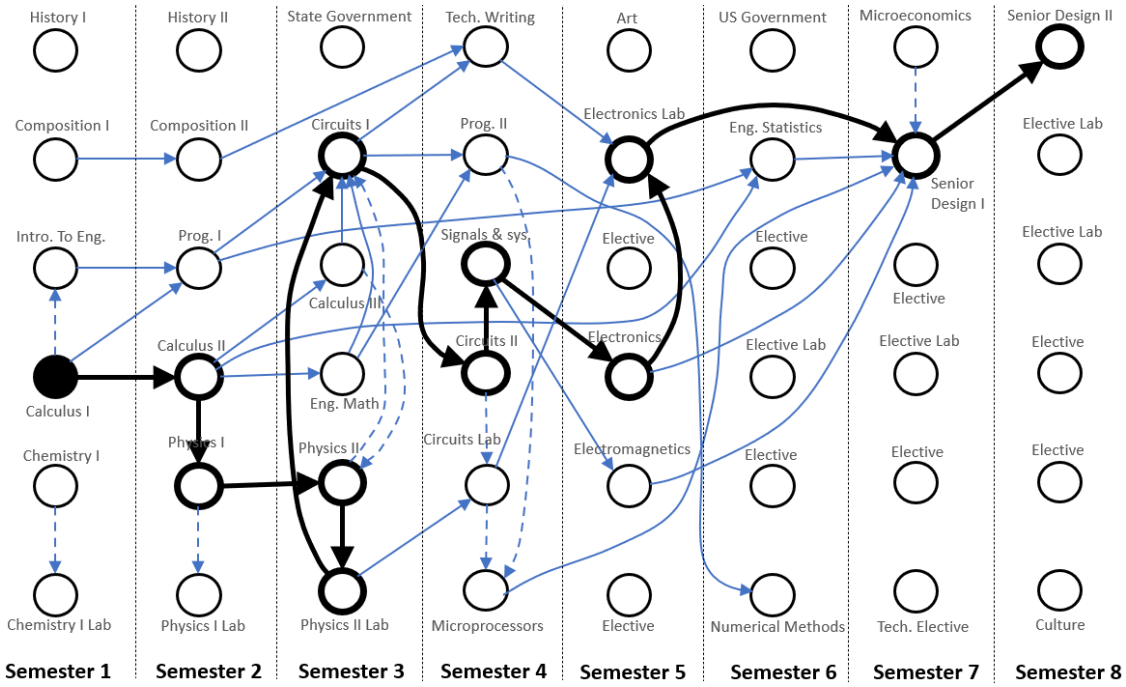
Figure 2: Divergent four-course academic structures: displayed in (a) and (b) is Curriculum 1, while Curriculum 2 is illustrated in (c) and (d). The delay factor for each course is depicted in parts (a) and (c), and the corresponding blocking factor is presented in parts (b) and (d) for these respective curricula.

where m is the total number of courses in the curriculum.

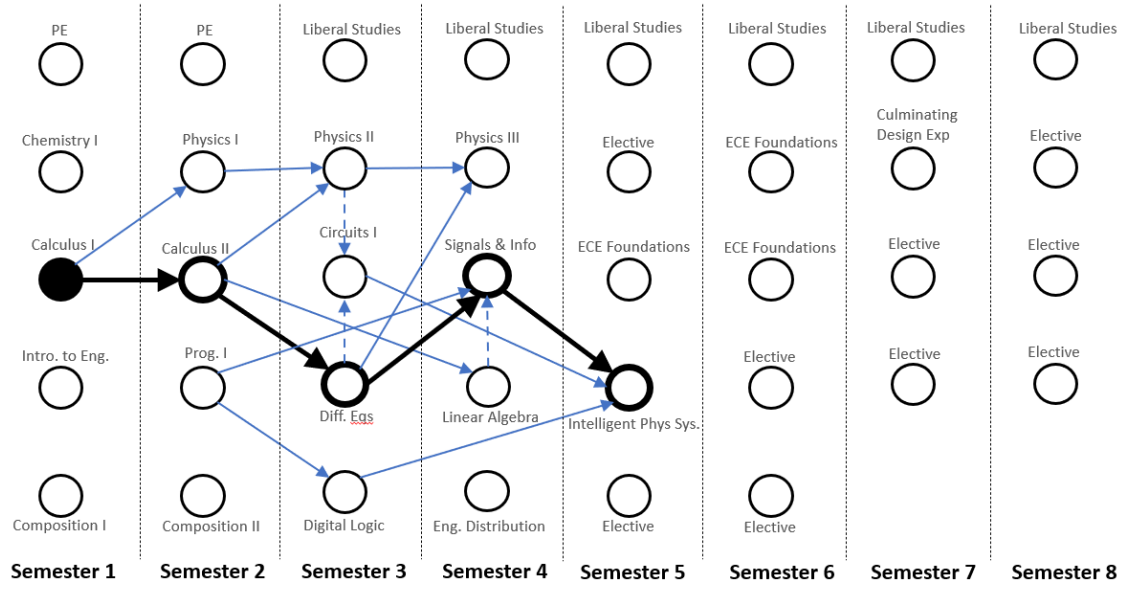
This framework has been instrumental in guiding significant curricular reforms. For instance, the College of Engineering and Computer Science at Wright State University (CECS) reported noteworthy improvements in student performance, retention, and graduation rates following a strategic curricular overhaul, as shown in Figure 4. Such reforms are particularly impactful when they involve introducing or repositioning foundational courses, altering the prerequisite structure and, consequently, the curriculum's complexity. This approach has also been successfully adopted by other institutions, such as the University of New Mexico (UNM), leading to marked improvements in graduation rates and substantial financial benefits for students. In the broader context of curriculum analysis, these findings align with the growing body of literature emphasizing the significance of curriculum structure in student success. Research in this area has repeatedly underscored the importance of strategic course sequencing and the reduction of bottleneck courses to facilitate smoother student progression and higher retention rates. This analytical framework, therefore, not only offers a practical tool for curriculum designers but also contributes to the evolving academic discourse on optimizing curriculum design for enhanced student outcomes.

3.2 Markov Decision Processes in Curricular Analytics Modeling

Transitioning from the analytical framework discussed above, we now focus on applying Markov Decision Processes (MDP) in curricular analytics. MDPs, as graphical models, are adept at representing sequential decision-making in systems defined by states S, actions A, and rewards R[23]. Expanding on the Markov chain model, MDPs equip decision-makers with various non-deterministic actions at each state s, a feature critical in systems exhibiting stochastic behav-

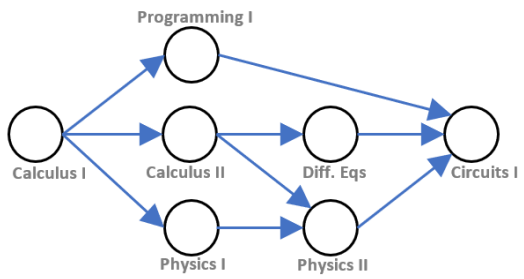


(a)

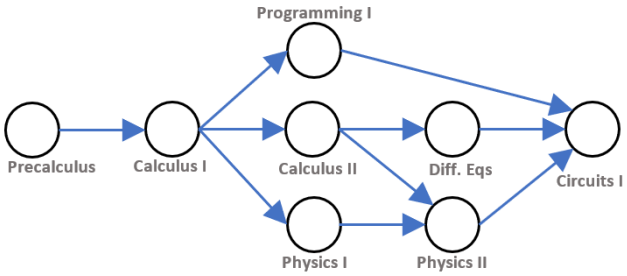


(b)

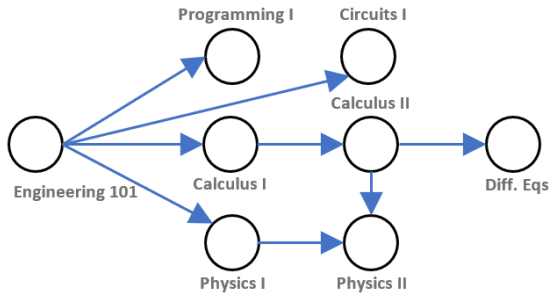
Figure 3: The two curricula featured in Figure 1 illustrate the longest paths (with highlighted edges) and courses blocked (indicated by bold vertices) by Calculus I (depicted as the black vertex). In curriculum (a), the delay factor associated with every course on the longest path is 11, and the blocking factor associated with Calculus I is 23. In curriculum (b), the delay factor for each course on the longest path is 5, and the blocking factor associated with Calculus I is 9. It is crucial to note that multiple longest paths are present in each curriculum



(a)



(b)



(c)

Figure 4: Illustrations of curriculum design frameworks within the scope of electrical engineering, showcasing the achievement of learning outcomes for Circuits I. (a) Displays a four-semester sequence for students prepared to start with Calculus I, characterized by a structural complexity score of 41. (b) Presents a five-semester sequence for students who are not initially prepared for Calculus I, featuring a structural complexity score of 60. (c) Describes an adapted four-semester sequence for students beginning without Calculus I readiness, offering a decreased structural complexity of 51.

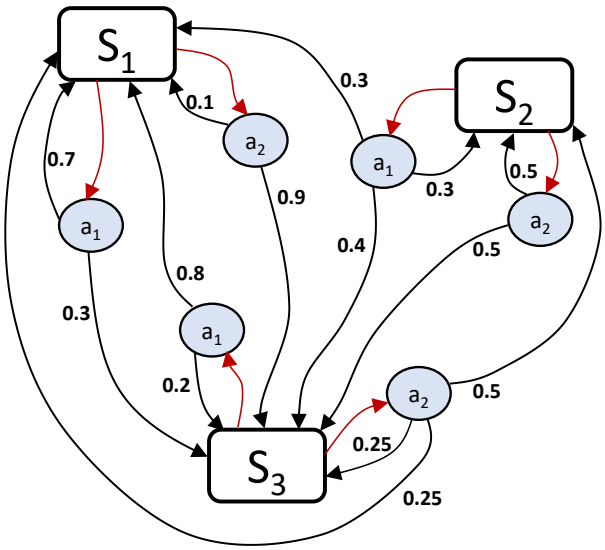


Figure 5: Example of a Markov Decision Process with three states and two actions.

iors[24]. Figure 5 demonstrates an MDP with three states.

In an MDP setting, the system at time t is in a state s, and an agent selects an action a transitioning the system to a new state s' with a corresponding expected reward $R_a(s, s')$. The transition to state s' depends on the action at state s, represented by the transition function $P_a(s, s')$. Hence, an MDP is formalized as a 4-tuple (S, A, P_a, R_a), where:

- S denotes the set of system states.
- A is the set of agent actions.
- $P_a(s, s') = P(s_{t+1} = s'|s_t = s, a_t = a)$ quantifies the transition probability to state s' at time $t + 1$, given the current state s and action a.
- $R_a(s, s')$ is the reward received upon moving from state s to s' using action a.

The agent's objective in an MDP is to identify an optimal policy π that maximizes rewards. The policy π determines the action selection probability in a given state s:

$$\pi(a|s) = P(a_t = a|s_t = s)$$

With policy π established, the MDP transforms into a discrete-time Markov chain (DTMC) with a transition matrix $\mathbf{P}_\pi$:

$$\mathbf{P}_\pi[s, s'] = \sum_{a \in A_s} \pi(a|s) P_a(s, s') \tag{2}$$

The probability of being in state s at time t is thus:

$$\mathbb{P}(s_t = s) = (\mathbf{s_0}.\mathbf{P}_\pi^t)(s) \tag{3}$$

where $\mathbf{s_0}$ is the initial state distribution vector. This equation forms the foundation of our MDP model in curricular analytics, allowing for predicting student graduation rates over time. Given

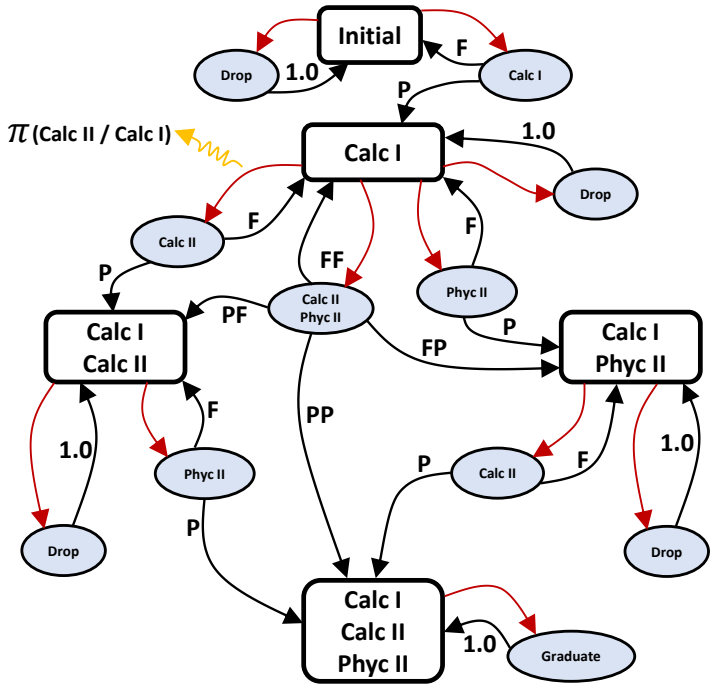


Figure 6: Sample MDP network depicting various courses, prerequisites, transitions, and choices.

the stochastic nature of student course choices each semester, MDP is aptly suited for modeling student progress in curricular analytics. For example, students might opt for different courses or course combinations each semester, introducing variability in actions and progression states. This variability underscores MDP's suitability for capturing students' dynamic progression. To illustrate, Figure 6 represents student progression using an MDP model. The states S indicate the courses completed and passed by a student. The actions A, shown as blue nodes, symbolize the choices available after completing a course, such as enrolling in new courses or dropping out. For instance, upon passing $Calc\ I$ (the $Calc\ I$ state), a student might choose to enroll in $Calc\ II$, $Phyc\ II$, both, or drop out. The transition function $P_a(s, s')$ is determined by the course pass/fail rates. Enrollment in a course leads to two potential outcomes: passing and transitioning to a new state s', or failing/dropping and remaining in the same state s. These transitions are represented by directed edges P and F, where $P + F = 1.0$. Using Eq. 3, the probability of students being in various states within the MDP network can be calculated, enabling the prediction of graduation rates by assessing the proportion of students who have completed all required courses, signified by the absorbing state in the MDP network. Subsequent sections will demonstrate the application of this MDP model in real-world scenarios, showing how minor curricular changes can significantly influence graduation rates. This approach aligns with current research emphasizing strategic course sequencing and bottleneck reduction as key factors in enhancing student success through curriculum structure.

4 A Case Study

The versatility of the Markov Decision Processes (MDP) model allows it to be applied across a diverse range of scenarios, catering to the specific requirements of different users such as faculty,

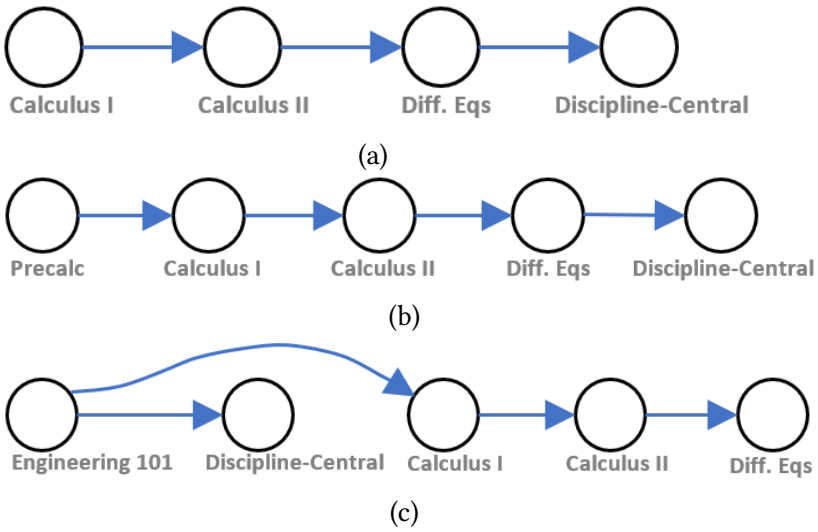


Figure 7: Engineering program curricular patterns, showing course structural complexities within each vertex. (a) Standard design for calculus-ready students, with a complexity of 22. (b) Alternative design for non-calculus-ready students, with a complexity of 35. (c) Revised design for non-calculus-ready students, reducing complexity to 25.

administrators, and students. This case study explores the impact of curricular modifications on graduation rates. We utilize the original and revised curricular patterns depicted in Figure 7, modeling them through MDP as demonstrated in Figure 8. The structural differences between these two patterns are distinct, with the complexity of the original pattern calculated at 35 compared to 25 for the revised pattern, as determined by Equation 1. This decrease in complexity correlates with an improvement in graduation rates. Figures 8a and 8b provide MDP representations of the engineering curricular patterns from Figures 7b and 7c, denoted as M_1 and M_2, respectively. The policies $\pi^{M_1}(a/s)$ and $\pi^{M_2}(a/s)$, integral to these representations, are also included in Figure 8. Generally, these policies are inherently shaped by the curricular structure itself. For instance, in Figure 7b, a student completing the $Precalc$ course is typically limited to enrolling in $Calculus\ I$, barring any dropout scenarios. Therefore, for the $Precalc$ state in Figure 8a, the only viable action is enrolling in $Calc\ I$ (i.e., $\pi(CalcI/Precalc) = 1.0$). The state transition functions $P_a^{M_1}(s, s')$ and $P_a^{M_2}(s, s')$ for M_1 and M_2 are determined by the courses' pass/fail rates, as depicted by the P and F edges in Figure 8. Therefore, when a student registers for a course, there is a probability P of advancing to a new state and a probability F of remaining in the current state. These transition functions, alongside the initial state distributions $\mathbf{s_0^{M_1}}$ and $\mathbf{s_0^{M_2}}$, are detailed in Table 2. Utilizing Equation 2, the transition matrices for both M_1 and M_2 are defined, thereby facilitating the computation of the probability of a student being in a particular state s at any given semester n, as specified in Equations 6 and 7.

$$\mathbf{P}_\pi^{M_1}[s, s'] = \sum_{a \in A_s} \pi^{M_1}(a/s) P_a^{M_1}(s, s') \tag{4}$$

$$\mathbf{P}_\pi^{M_2}[s, s'] = \sum_{a \in A_s} \pi^{M_2}(a/s) P_a^{M_2}(s, s') \tag{5}$$

	State					
	state 1	state 2	state 3	state 4	state 5	**state 6**
semester 1	0.3	0.7	0.0	0.0	0.0	**0.0**
semester 2	0.09	0.42	0.49	0.0	0.0	**0.0**
semester 3	0.027	0.189	0.441	0.343	0.0	**0.0**
semester 4	0.0081	0.0756	0.2646	0.4116	0.2401	**0.0**
semester 5	0.0024	0.0284	0.1323	0.3087	0.3602	**0.1681**
semester 6	0.0007	0.0102	0.0595	0.1852	0.3241	**0.4202**
semester 7	0.0002	0.0036	0.0250	0.0972	0.2269	**0.6471**

(a) The distribution of students in the original pattern in different states at different semesters.

	State								
	state 1	state 2	state 3	state 4	state 5	state 6	state 7	state 8	**state 9**
semester 1	0.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0	**0.0**
semester 2	0.09	0.3465	0.196	0.196	0.1715	0.0	0.0	0.0	**0.0**
semester 3	0.027	0.1306	0.1352	0.1558	0.3284	0.0549	0.1681	0.0	**0.0**
semester 4	0.0081	0.0444	0.0629	0.0833	0.2775	0.0486	0.3288	0.0154	**0.1311**
semester 5	0.0024	0.0143	0.0247	0.0374	0.1700	0.0271	0.3219	0.0182	**0.3839**
semester 6	0.0007	0.0045	0.0088	0.0152	0.0876	0.0122	0.2292	0.0130	**0.6286**
semester 7	0.0002	0.0014	0.003	0.0058	0.0405	0.0048	0.1357	0.0073	**0.8012**

(b) The distribution of students in the revised pattern in different states at different semesters.

Table 1: A comparison in the distribution of students over the states of the original and the revised patterns computed over seven semesters.

and the probability of being in state s (i.e., passing a course or a set of courses) at semester n can be defined as:

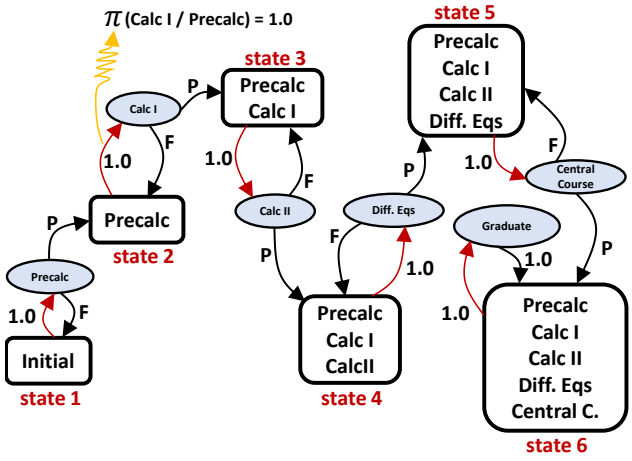
$$\mathbb{P}^{M_1}(s_n = s) = (\mathbf{s_0^{M_1}} .[\mathbf{P}_\pi^{M_1}]^n)(s) \tag{6}$$

$$\mathbb{P}^{M_2}(s_n = s) = (\mathbf{s_0^{M_2}} .[\mathbf{P}_\pi^{M_2}]^n)(s) \tag{7}$$

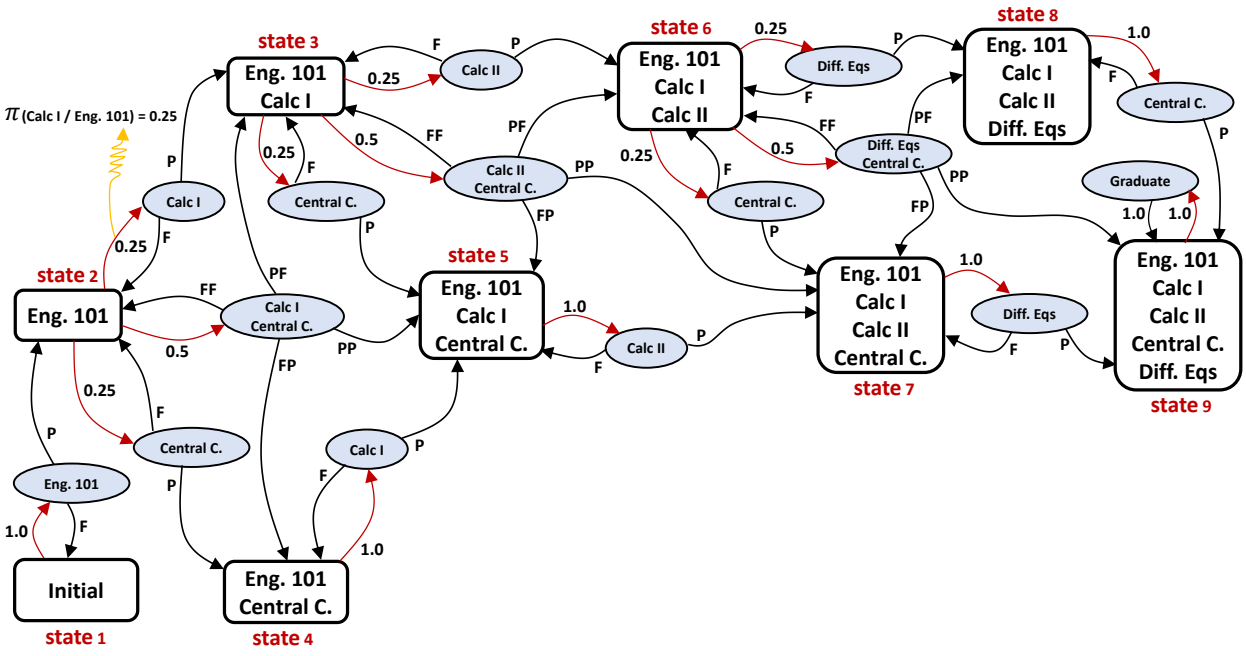
Table 1 presents a comparative analysis of student progression in the original and revised patterns over seven terms. The data indicates that, by the end of the second semester, 49% of students enrolled in the original pattern successfully pass both the $Precalc$ and $Calc\ I$ courses, while 42% pass only the $Precalc$ course, and 9% fail to pass either. It is crucial to note that the final states in Figures 8a and 8b, labeled as state 6 and state 9, represent the absorbing states, signifying successful graduation. As observed in these states, graduation rates demonstrate a consistent upward trend over the semesters. Figure 9 visually underscores this increasing pattern of graduation rates, highlighting that students in the revised pattern are graduating faster than their counterparts in the original pattern. By the seventh semester, there is a notable difference, with 80% of students completing the revised pattern, in contrast to only 65% in the original pattern. This disparity exemplifies the beneficial impact of reducing curricular complexity on enhancing graduation rates.

5 Implementing Curricular Analytics: Practical Approaches

The methodologies developed in Section 3 present a range of potential applications for analyzing curricula and guiding reform efforts to enhance student success outcomes. Traditionally, cur-



(a) The Markov Decision Process depiction of the original curricular pattern.



(b) The Markov Decision Process depiction of the revised curricular pattern.

Figure 8: The Markov Decision Process illustration of both the original and revised curricular designs.

$P_a^{M_1}(s', s)$		s'					
s	a	**state 1**	**state 2**	**state 3**	**state 4**	**state 5**	**state 6**
state 1	*Precalc*	0.3	0.7	0.0	0.0	0.0	0.0
state 2	*Calc I*	0.0	0.3	0.7	0.0	0.0	0.0
state 3	*Calc II*	0.0	0.0	0.3	0.7	0.0	0.0
state 4	*Diff. Eqs*	0.0	0.0	0.0	0.3	0.7	0.0
state 5	*Central Course*	0.0	0.0	0.0	0.0	0.3	0.7
state 6	*Graduate*	0.0	0.0	0.0	0.0	0.0	1.0

(a) The state transition probability function, denoted as $P_a^{M_1}(s, s')$, for the MDP network depicted in Figure 8a.

$P_a^{M_2}(s', s)$		s'								
s	a	**state 1**	**state 2**	**state 3**	**state 4**	**state 5**	**state 6**	**state 7**	**state 8**	**state 9**
state 1	*Eng. 101*	0.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
state 2	*Calc I*	0.0	0.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0
	Central C.	0.0	0.0	0.3	0.0	0.7	0.0	0.0	0.0	0.0
	Calc I, Central C.	0.0	0.09	0.21	0.21	0.49	0.0	0.0	0.0	0.0
state 3	*Calc II*	0.0	0.0	0.3	0.0	0.0	0.7	0.0	0.0	0.0
	Central C.	0.0	0.0	0.3	0.0	0.7	0.0	0.0	0.0	0.0
	Calc II, Central C.	0.0	0.0	0.09	0.0	0.21	0.21	0.49	0.0	0.0
state 4	*Calc I*	0.0	0.0	0.0	0.3	0.7	0.0	0.0	0.0	0.0
state 5	*Calc II*	0.0	0.0	0.0	0.0	0.3	0.0	0.7	0.0	0.0
state 6	*Diff. Eqs*	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.7	0.0
	Central C.	0.0	0.0	0.0	0.0	0.0	0.3	0.7	0.0	0.0
	Diff. Eqs, Central C.	0.0	0.0	0.0	0.0	0.0	0.09	0.21	0.21	0.49
state 7	*Diff. Eqs*	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.7
state 8	*Central C.*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7
state 9	*Graduate*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

(b) The state transition probability function, denoted as $P_a^{M_2}(s, s')$, for the MDP network depicted in Figure 8b.

	state 1	**state 2**	**state 3**	**state 4**	**state 5**	**state 6**
$s_0^{M_1}$	1.0	0.0	0.0	0.0	0.0	0.0

(c) The initial distribution of students within the MDP network as depicted in Figure 8a.

	state 1	**state 2**	**state 3**	**state 4**	**state 5**	**state 6**	**state 7**	**state 8**	**state 9**
$s_0^{M_2}$	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(d) The initial distribution of students within the MDP network as depicted in Figure 8b.

Table 2: The transition functions and initial state distributions for the MDP networks presented in Figure 8.

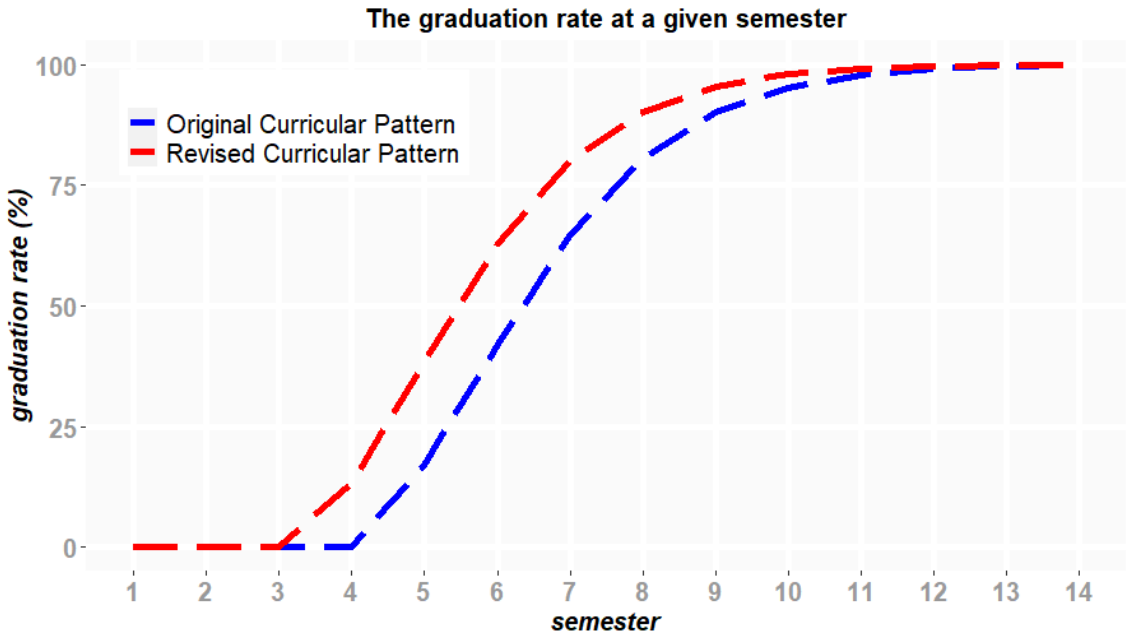


Figure 9: A comparative analysis of graduation rates between students following the original curriculum versus those in the revised program.

ricular reforms often relied on faculty insights and anecdotal evidence, which, while valuable, highlighted the absence of formal frameworks for systematic curriculum study. Historically, curriculum reform has been approached as a modification of a 'black box' system, where student characteristics, including prior preparation, are inputs, and student success rates are outputs. In this paradigm, the effectiveness of any reform was gauged by its long-term impact on student outcomes. The significant contribution of this work lies in its ability to quantify the curricular system within this 'black box', thus reducing system opacity and enabling direct analysis of the curriculum itself. This approach fosters predictive analytics as an essential component of curricular redesign efforts. We now provide specific examples to illustrate the utility of this approach.

5.1 Evaluating and Comparing Different Curricular Designs

One direct application of curricular analytics is comparing the complexities of academic programs. Given a set of curricula $C = \{c_1, \ldots, c_n\}$, there might be interest in comparing the structural complexities within C or relating these complexities to actual student success outcomes. For instance, consider a situation where C comprises all curricula in a specific field of study, categorized by CIP code[25]. A key motivator for the research presented in this paper was understanding the variations in perceived complexities of similar programs across different institutions and how these differences affect student success. The comparison between the two curricula in Figure 1 exemplifies the structural differences observed among similar programs at various schools. Notably, more significant structural variances in curricula within C are evident in fields where sequential knowledge development is crucial, particularly in STEM disciplines. The curricular complexity metrics developed in this paper allow us to rank the elements of C by structural complexity, raising the question of how to utilize this information effectively. Curriculum reformers apply this

information in various ways. Firstly, they compare their program to other institutions' programs to explore potential curriculum reform options. Alternatively, C could represent the historical curricula offered by a single program, providing valuable benchmarks for faculty considering further modifications. These benchmarks can assess how specific changes might affect structural complexity and expected student completion rates. If historical student success data is available, it can further refine the expectations of reform outcomes. For instance, faculty might use a benchmark curriculum to estimate the effect of instructional improvements on the curriculum. The MDP model described earlier could evaluate how enhancing the pass rate of a specific course influences overall completion rates, leading to a curriculum-wide sensitivity analysis for optimal resource allocation toward course improvements. In cases where C encompasses the curricula of all programs an institution offers, we have employed structural complexity rankings to correlate with actual six-year graduation rates using linear regression. At UNM, for example, we observed that every 17-point decrease in structural complexity corresponded to a 1% increase in the six-year graduation rate, motivating many programs to reduce their structural complexities. This result naturally leads to discussions about the relationship between structural complexity and program quality. Contrary to the assumption that higher complexity equates to higher quality, our initial investigations suggest an inverse correlation, particularly in engineering programs. This observed trend, where lower structural complexity aligns with higher perceived program quality (as per U.S. News & World Reports rankings), warrants further examination. We hypothesize that a principle akin to Occam's razor applies to curricula: the simplest curriculum, in terms of structural complexity, that enables students to achieve the program's learning outcomes is likely to yield the most favorable student success outcomes and, consequently, the highest quality program.

5.2 Patterns in Curriculum Design and Structure

The concept of curricular design patterns represents an intriguing facet of curricular analytics. The genesis of design patterns lies in the domain of architecture, where Alexander et al. introduced them as general solutions to recurring design challenges[26]. This idea was later adapted by Beck and Cunningham for software development, recognizing that software design patterns offer reusable solutions to common software design issues[27]. Design patterns form a lexicon for designers to articulate and address specific design challenges within a problem domain. In curriculum development, Heileman et al. applied the concept of design patterns to define a structured collection of curricular and co-curricular activities that collectively enable students to achieve specific learning outcomes within an educational context[15]. These activities, typically structured as courses with prerequisite and co-requisite relationships, ensure sequential learning where each course builds upon the knowledge acquired in its prerequisites. Klingbeil and Bourne's work in curriculum redesign illustrates the practical application of curricular design patterns, particularly in addressing the challenges faced by non-calculus-ready students in engineering programs[28]. A sophomore-level central discipline-specific course in many engineering curricula often requires $Differential\ Equations$ as a prerequisite. This prerequisite structure is depicted in Figure 7a, which assumes students are calculus-ready. For non-calculus-ready students, the traditional approach involves appending a $Precalculus$ course, as shown in Figure 7b. Klingbeil and Bourne observed that only a subset of $Differential\ Equations$, specifically linear differential equations, is utilized in most central engineering courses. They proposed a curricular redesign that

integrates the teaching of linear differential equations into a high-impact first-year engineering course, including precalculus topics. This innovative curriculum is presented in Figure 7c. Notably, students in this revised curriculum, starting with $Engineering\ 101$, can progress to the sophomore-level course concurrently or earlier than their calculus-ready peers. Our methodologies allow for quantifying the benefits of Klingbeil and Bourne's approach by demonstrating the reduction in structural complexity it offers[28]. Specifically, their approach results in a ten-point decrease in structural complexity compared to the traditional pattern for non-calculus-ready students. It is only marginally more complex than the pattern for calculus-ready students. To gauge the impact of these patterns on student success, consider a scenario where all courses in the patterns shown in Figure 7 have a 75% pass rate. Utilizing the MDP model from Eqs 3, we find that 82% of students can complete the calculus-ready pattern in Figure 7a within six terms, but only 53% can complete the traditional non-calculus-ready pattern in Figure 7b. However, students attempting the redesigned pattern in Figure 7c achieve a success rate of 83%. This pattern's flexibility, allowing the discipline-specific course to be taken in any of the final three terms, effectively equates its success rate to the calculus-ready students' pattern. Enhancing student success through high-impact courses like $Engineering\ 101$ often involves additional support services. If these services effectively raise the pass rate of $Engineering\ 101$ to 95%, the success rate for students following the pattern in Figure 7c could increase to 88%. To further illustrate the application of this curricular design pattern, we examine the electrical engineering context as shown in Figure 1. A critical course in electrical engineering curricula is $Circuits\ I$, with learning outcomes that include understanding basic electrical circuit elements, applying Ohm's and Kirchhoff's laws, appreciating linearity principles such as superposition, and analyzing first and second-order linear circuits. The innovative approach in curriculum design, as evidenced in Klingbeil and Bourne's work, paves the way for more effective and inclusive engineering education, particularly for students entering with varying levels of preparedness. The application of curricular analytics extends to analyzing and optimizing design patterns within curricula. Consider the seven-course curricular pattern depicted in Figure 7a, crafted to facilitate students in achieving the $Circuits\ I$ learning outcomes, presuming they are prepared for calculus. This pattern's structural complexity, calculated using Equation 1, totals 41. Notably, the longest pathway within this pattern spans four courses, setting a minimum completion time of four terms. In contrast, Figure 4b illustrates an eight-course curricular design pattern, as per Figure 7b, tailored for students not initially prepared for $Calculus\ I$. This design applies the conventional solution of inserting a $Precalculus$ course initially. Consequently, the structural complexity of this pattern increases to 60, marking a 31% rise compared to the pattern in Figure 4a, and extending the minimum completion time to five terms. Klingbeil and Bourne's key insight was that only specific parts of the learning content in a course like $Differential\ Equations$ are necessary for subsequent courses, such as $Circuits\ I$[28]. In particular, understanding first and second-order linear differential equations is crucial for the fourth learning outcome in $Circuits\ I$. This observation enables the application of a more streamlined curricular pattern, as seen in Figure 7c, resulting in the curricular design shown in Figure 4c. This revised pattern for non-calculus-ready students exhibits a structural complexity of 51, only 20% higher than the calculus-ready pattern in Figure 4a. Yet, it can be completed in four terms, unlike the five-term pattern in Figure 4b. The curricular design pattern in Figure 4c not only accommodates students who are not initially calculus-ready but does so with significantly reduced complexity. This reduction in complexity is expected to enhance student success rates. Using the MDP model with a fixed course pass rate of 75%, we find that 72%

of students can complete the pattern in Figure 4a within six terms. In contrast, only 36% manage to complete the more complex pattern in Figure 4b. Remarkably, students following the pattern in Figure 4c achieve a completion rate of 72% within the same timeframe, equating their success to that of their calculus-ready counterparts. This section underscores the potential of curricular analytics in developing alternative curricula, such as the one demonstrated in Figure 4c. The ability to tailor curricular patterns to different student preparedness levels while maintaining or enhancing the success rate exemplifies the transformative power of curricular analytics in academic planning and student success.

6 Building Degree Plans and Deconstructing Curricula

In examining the curricula depicted in Figure 1, we observe structured degree plans that enable students to fulfill all requirements within eight terms. This observation raises the question: are certain degree plans inherently more conducive to student success within a program? While all valid degree plans for a curriculum maintain identical structural complexity, due to the immutable pre- and co-requisite course relationships, the distribution of this complexity can vary across different terms within these plans. Developing criteria for distributing complexity and optimizing degree plans accordingly is a promising approach. Slim et al. emphasized the need for students to complete crucial courses early in their academic journey and proposed an algorithm to generate degree plans optimized with this strategy in mind[11,7]. Alternatively, spreading complexity evenly, especially in the initial years, could benefit students at risk of dropping out due to academic challenges. These methodologies hint at the potential of personalized degree plans tailored to individual student capabilities and aimed at maximizing the probability of successful completion. One innovative angle considers the conditional dependence of instructional complexity on individual student characteristics and the combination of courses in a term. By devising a metric for instructional complexity that reflects the expected performance of various student categories in specific class combinations, we can forge degree plans optimized for the success of these student groups. Further exploration in this area is merited. While assuming a static curriculum graph, these strategies do not account for the potential benefits of altering the curriculum structure itself. As discussed in Section 5.2, curricular redesign can be effectively approached through curriculum decomposition. This method involves breaking down a curriculum into its constituent learning outcomes and mapping the dependencies among these outcomes. The challenge lies in reassembling these outcomes into courses and creating new prerequisites. This systematic approach to curriculum redesign can lead to variations in structural complexities, providing a framework for automated curricular improvement algorithms. However, decomposing courses into learning outcomes and documenting dependencies is a complex and labor-intensive process, especially given the breadth of courses available. Reassembling these outcomes into coherent courses is further complicated by the diversity of topics covered in learning outcomes, which may not align neatly within a single course. To address these challenges, we introduce a novel approach in this work, leveraging machine learning and latent graphical models. This method utilizes actual student data, applying machine learning techniques to identify learning outcomes with compatible topics. Subsequently, latent graphical model algorithms are employed to restructure curricula to reduce their complexities. We demonstrate the effectiveness of our model using real student datasets, modeling the dependency structures of actual curricular patterns in universities. The subsequent results and discussions highlight the practicality and impact

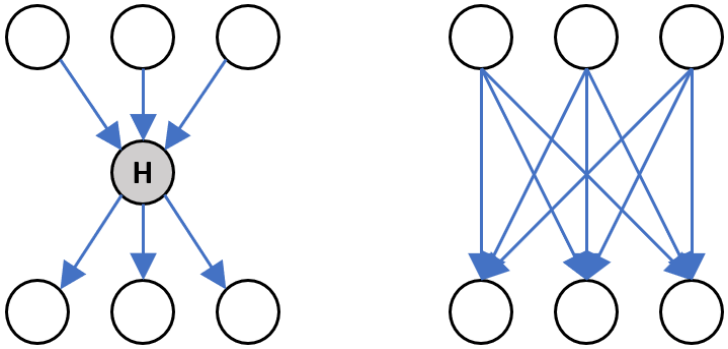


Figure 10: Two Directed Acyclic Graphs (DAGs) representing relationships between symptoms and causes, one with a hidden node and the other without. Symptoms like chest pain are leaf nodes, while root nodes represent causes like smoking and diet. Hidden nodes, signifying mediating factors like heart disease, show how including hidden elements can enhance understanding and reduce network complexity.

of our proposed approach in streamlining curricular complexities and enhancing student success.

7 Latent Graphical Models and Curriculum Complexity

Using latent variables to model intricate systems is widely recognized as a pivotal method in domains like bioinformatics, computer vision, and machine learning research[29]. In the context of curriculum complexity, we focus on latent variable models structured as trees, termed "latent trees". These models effectively illustrate the interplay between observable variables and their latent counterparts. The latent nodes encapsulate common characteristics of their observable descendants. This methodology strikes an optimal balance between representational power (e.g., the capability to model cliques) and the computational feasibility of learning and inference processes (e.g., the exactitude of message passing in tree structures). Why opt for tree graphic models with hidden variables? A primary reason is the potential for constructing networks of reduced complexity. Moreover, these networks can often unveil underlying structures in data[30]. Imagine a scenario with multiple observable evidence variables, E_1 to E_n, ranging from a patient's symptoms to individual movie preferences. A highly connected graph is typically needed to represent the full joint distribution among these variables. However, introducing a "cause" node can substantially simplify the model. This node might symbolize an underlying disease causing various symptoms or a fundamental preference influencing movie choices, as depicted in Figure 10. Translating this to an educational setting, a curricular pattern can be envisioned as a tree graphical model where nodes represent courses and states corresponding to possible grades. An edge connecting courses A and B may indicate a correlation in student performance in these courses. Adding a hidden node in this context is analogous to integrating a new course into the curricular pattern. This hidden node could symbolize an essential prerequisite, much like the underlying disease in the medical example of Figure 10. Building on this concept, restructuring a curriculum would involve steps like adding a new course (a hidden node), introducing a prerequisite (a directed edge), or removing an existing prerequisite (eliminating an edge). While these additions have been manual, the critical question is whether this process can be automated. The answer to this lies in exploring machine learning techniques and latent graphical models, which

we delve into in subsequent sections. This approach promises to revolutionize curriculum design by automating the identification of crucial but previously unobserved interconnections between courses, thereby facilitating more effective and efficient learning pathways.

8 Learning Latent Tree Graphical Models

In pursuing understanding and modeling complex systems, especially in the context of curricular analytics, the transition from traditional models to advanced graphical representations marks a significant leap. This section delves into the realm of Latent Tree Graphical Models (LTGMs), a sophisticated approach to unraveling the intricate web of dependencies and relationships inherent in educational structures. LTGMs stand out for their ability to integrate unseen yet influential variables—latent factors that underpin and shape the observable characteristics of a curriculum. By exploring this advanced modeling technique, we aim to unlock more profound insights into the dynamics of educational pathways, providing a more nuanced understanding of how various courses and learning outcomes interconnect and influence one another. This exploration enhances our ability to map out complex curricular structures and opens new avenues for optimizing educational strategies and outcomes.

8.1 Conceptual Foundations

In high-dimensional data analysis, intricate statistical dependencies often challenge conventional modeling techniques. A solution lies in probabilistic graphical models that bridge observed features with latent variables. By defining a joint probability distribution over both observed and latent variables, these models facilitate the integration of latent variables to derive the observed variables' marginal distribution. This transformation allows the representation of complex distributions over observed variables (like cliques) in more manageable joint models (like tree models) within an expanded variable space. Such models have found diverse applications in areas like document analysis, social networking, speech recognition, and bioinformatics[31].

8.2 Latent Tree Models

The latent tree model is a graphical model Markov on trees, featuring a mix of observed and latent variables. Its computational efficiency lies in its tree-structured nature, simplifying and scaling inference processes. Despite the constrained nature of these models, their relevance in numerous applications validates the exploration undertaken in this paper. Specifically, we align with the acyclic prerequisite dependencies in curricular structures, where latent variables can represent new or potential courses.

8.3 Approach to Learning Latent Variables

To infer latent variables within a curriculum, we adopt the methodology from[29], which begins with a distance matrix of observed variables (existing courses) and iteratively introduces latent nodes. This approach guarantees structural recovery under certain conditions and surpasses heuristic methods previously employed in latent tree structure estimation. Examples of such heuristics include Zhang et al.'s local search heuristic for hierarchical latent class models[32],

Harmeling and Williams' greedy algorithm for binary trees[33], and Bayesian hierarchical clustering for merging clusters based on statistical tests[34].

8.4 The Choi Model

Choi et al.'s development in general Latent Tree Model (LTM) learning represents a significant advancement, especially in analyzing binary data with shared state space[35,29]. Their method leverages the additive tree metric property to recover child-parent and sibling relations, initiating with a minimum spanning tree (MST) among observed variables. Subsequently, local latent trees are integrated, replacing internal nodes and their neighbors with these structures, thereby streamlining the overall complexity of constructing the latent tree. Remarkably, the sample complexity of these algorithms aligns with that of the Chow-Liu algorithm, requiring only logarithmic observations in the number of variables to recover the model with high probability[36].

8.5 Information Distances

Central to Choi et al.'s algorithms are the 'information distances'—functions of pairwise distributions among observed variables and additive for tree-structured graphical models[29]. For Gaussian models, the information distance, d_{ij}, between variables X_i and X_j is defined as:

$$d_{ij} := -\log \mid \rho_{ij} \mid \tag{8}$$

where ρ is the correlation coefficient:

$$\rho_{ij} := \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_j)}} \tag{9}$$

A large d_{ij} implies weak correlation between X_i and X_j, and conversely for a small d_{ij}. This section has laid the groundwork for understanding latent tree models and their learning processes, setting the stage for applying these concepts to curriculum complexity analysis. The ability to uncover and integrate latent courses within a curriculum promises to revolutionize the way educational pathways are structured, potentially leading to more efficient and effective learning outcomes.

9 Restructuring Curricula

Building upon the Latent Tree Model (LTM) framework outlined by Choi et al., this section delves into applying this model to strategically restructure academic curricula. The primary aim is to simplify the curricular structure, aligning it with a tree-like graphical representation to facilitate a more straightforward progression for students through their academic requirements. The restructuring process involves adding new courses (represented as hidden nodes in the LTM), eliminating or modifying existing prerequisite connections, and incorporating new ones (represented as directed edges between nodes in the LTM). The fundamental input for this restructuring process is the *information distance* matrix, denoted as D. This matrix is pivotal in gauging the

inter-course correlations within a curriculum, effectively measuring the "closeness" between various courses. The notion of "closeness" here is multifaceted, contingent upon the specific application context. For our purposes, courses are deemed "close" if they share common foundational requirements, thematic or topical similarities, or comparable levels of academic rigor and competence requirements. In pursuit of a nuanced understanding of this "closeness," our proposed framework to compute the distance matrix D integrates two distinct methodologies: a machine learning approach and a graph theory model. The machine learning component leverages student performance data to deduce the foundational competencies required for each course. These competencies might include critical writing, analytical reasoning, problem-solving aptitude, etc. Concurrently, the graph theory employs the directed links within the curriculum's graphical representation to assess the relative distances or "closeness" between courses. This approach also accounts for the sequential or prerequisite dependencies inherent in course progression (e.g., the normative sequence of taking *Calculus I* prior to *Calculus II*). Subsequent sections will explore these methodologies in-depth, elucidating how they collectively contribute to the calculated restructuring of academic curricula within the LTM paradigm.

9.1 Leveraging Machine Learning in Curricular Analysis

In curricular analysis, machine learning, particularly collaborative filtering (CF), emerges as a potent tool to unearth latent attributes that signify the foundational skills required by various courses in an academic curriculum. Collaborative filtering, a technique ubiquitous in recommender systems, thrives on collaborating among multiple agents or data sources to filter information or discern patterns[37]. Its versatility is evident across numerous domains, including environmental sensing, financial data integration, and user behavior analysis in electronic commerce. In the educational sphere, particularly within a Grading Management System (GMS), collaborative filtering echoes the dynamics of recommender systems, where it typically consists of three elements: the user, the item, and the rating. The core task revolves around predicting the ratings for unrated items and recommending those with the highest anticipated ratings. Analogously, in a GMS, these elements translate to the student, the course, and the grade, respectively, with the primary task being predicting grades for courses yet to be taken. However, applying collaborative filtering in this context transcends mere grade prediction. The principal objective is to extract latent features corresponding to courses within a curriculum. For instance, in a movie recommendation scenario, the discovered factors include dimensions like genre or thematic elements. Similarly, in an educational setting, the factors for courses could encompass dimensions such as required problem-solving skills, critical reading and writing skills, or specific subject knowledge like trigonometry. These extracted latent features are instrumental in measuring the compatibility or "closeness" between courses, which subsequently informs the construction of the *information distance* matrix, D. A prominent class of collaborative filtering models employed for latent feature extraction is matrix factorization[37]. This approach involves approximating a matrix $A \in R^{|S|x|C|}$ as the product of two smaller matrices $U \in R^{|S|x|K|}$ and $V \in R^{|K|x|C|}$. Here, U represents a matrix where each row is a vector of k latent factors for each student s, while V is a matrix where each column is a vector of k latent factors for each course c. In this context, Non-negative Matrix Factorization (NMF) is particularly suitable[38]. Unlike other matrix factorization techniques, NMF mandates that all entries in the matrices U and V be non-negative. This constraint aligns well with the nature of curricular data, where the set of skills required by a

	Course A	Course B	Course C
Student 1	A+	NA	NA
Student 2	NA	C	A
Student 3	C	NA	D
Student 4	NA	C	A
Student 5	A+	B	B-
Student 6	NA	NA	NA

Table 3: A student-course matrix $A \in R^{6x3}$

course or the competence level of a student cannot be negative; the least value is zero, indicating no requirement or competence in a specific skill. Consider a matrix A with S students and C courses, i.e., $A \in R^{|S|x|C|}$, where each entry represents student i performance in a particular course j. Table 3 exemplifies such a matrix. The subsequent steps to define the *information distance* matrix, D_M, using NMF are:

1. Apply the NMF algorithm to decompose A into matrices U and V, ensuring convergence to a stationary point.
2. Employ cross-validation to determine the optimal number of latent factors, K, that yield the minimum root mean square error (RMSE).
3. Each row s_i in U corresponds to student i's competence in K academic skills, and each column c_j in V represents the required skill level for course j in the same K skills.
4. The information distance between courses c_i and c_j is defined as:

$$d_{ij} := -\log \mid \rho_{ij} \mid$$

   where ρ_{ij} is the correlation coefficient:

$$\rho_{ij} := \frac{\mathrm{Cov}(c_i, c_j)}{\sqrt{\mathrm{Var(c_i)Var(c_j)}}}$$

5. Construct D_M, the *information distance* matrix, with elements d_{ij} for all course pairs.

This machine learning approach facilitates a nuanced understanding of course relationships within a curriculum, paving the way for strategic curricular restructuring that aligns with student needs and academic objectives.

9.2 Expanding the Graph Theory Approach in Curriculum Analysis

Our exploration of curriculum restructuring extends into the realm of graph theory, where we leverage the inherent structure of a curriculum to develop an alternative *information distance* matrix, denoted as D_G. This matrix utilizes the concept of prerequisite paths to measure the closeness between courses. The path length between two courses in a curriculum graph reflects the strength of their relationship in terms of learning outcomes. For instance, the connection

	Precalc	Calc I	Calc II	Diff Eqs.	Circuits I
Precalculus	0	1	2	3	4
Calculus I	∞	0	1	2	3
Calculus II	∞	∞	0	1	2
Diff Eqs.	∞	∞	∞	0	1
Circuits I	∞	∞	∞	∞	0

Table 4: A sample *information distance* matrix D_G

between *Calculus I* and *Calculus II* in Figure 4a is intuitively stronger than that between *Calculus I* and *Circuits I*, a fact easily captured by examining the lengths of their connecting paths. The directionality of prerequisite edges is also instrumental in shaping our understanding of these relationships. It dictates the sequential order in which courses must be taken, such as *Calculus II* following *Calculus I* and not vice versa. This asymmetric relationship is reflected in the *information distance* matrix D_G, as showcased in Table 4 for a sample curriculum. In integrating the machine learning and graph theory approaches, we synthesize the final *information distance* matrix, D, through a weighted combination of D_M and D_G:

$$D = \alpha_1 \cdot D_M + \alpha_2 \cdot D_G \tag{10}$$

where $\alpha_1 + \alpha_2 = 1.0$. These coefficients offer flexibility in emphasizing either the prerequisite structure or the background skill requirements of courses. For instance, a higher α_1 focuses more on the background skills required by the courses, while α_2 leans towards the prerequisite paths. To ensure a unified scale for effective comparison, both matrices are normalized using the min-max scaling technique, substituting ∞ with a significantly large number for computational purposes. The choice of latent trees in our model is guided by the need to balance model fitting and complexity to avoid overfitting. This is where the Bayesian Information Criterion (BIC) becomes crucial, providing a quantifiable metric to evaluate the performance of our latent tree models. The BIC formula considers both the log-likelihood of the model and its complexity, with the latter scaling linearly with the number of hidden variables in a tree structure:

$$BIC(T) = \textit{log-Likelihood} - \frac{k(T)}{2}\log(n) \tag{11}$$

Here, T represents the latent tree structure, n is the dataset size, and $k(T)$ is the count of free parameters. The optimal model balances accurately representing the empirical data distribution and maintaining a manageable level of complexity. This approach underlines our commitment to crafting a curriculum model that is both data-driven and practically feasible, ensuring that the resulting structure aligns well with the academic needs and goals of students.

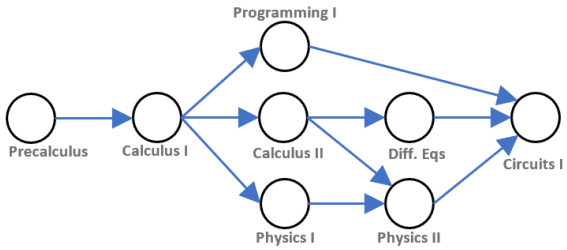
10 Analyzing Experimental Outcomes

This section delves into the practical application of our framework, utilizing actual student data and curricular patterns to assess the efficacy of our model. Our primary focus is the application of the latent tree algorithms from Choi et al. on real-world curricular structures[29]. We commence with a smaller-scale curricular pattern and expand our analysis to encompass a comprehensive

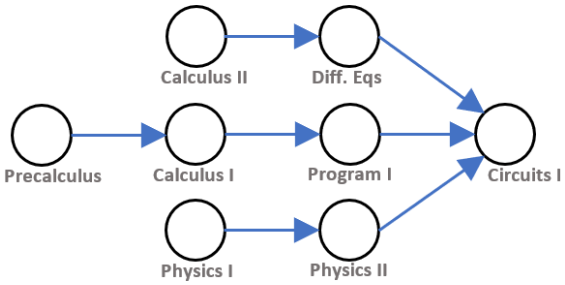
computer engineering program. Both experiments leverage the performance data of computer engineering students from the University of New Mexico (UNM), aiming to unearth latent features that underpin the required skillsets for courses within these curricular patterns. Subsequently, we apply a suite of latent tree learning algorithms, notably the neighbor-joining (NJ), recursive grouping (RG), Chow-Liu Neighbor-Joining (CLNJ), Chow-Liu Recursive Grouping (CLRG), regularized Chow-Liu Neighbor-Joining (regCLNJ), and regularized Chow-Liu Recursive Grouping (regCLRG) algorithms. Our evaluation criteria encompass:

- **Log-Likelihood estimate** to assess the data fit of a tree model.
- **BIC estimate** for evaluating data fit while considering model complexity.
- **Complexity** to gauge the structural complexity of a tree model, calculated via Equation 1.
- **Graduation rates**, estimated using the MDP model (Equation 3), to measure student success.

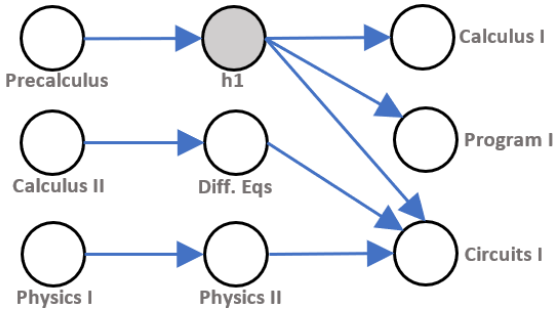
Our initial experiment involves restructuring a curricular pattern (Figure 11a) consisting of 8 courses and 10 prerequisite connections. This pattern's structural complexity is 60. We calculate D_G using the graph structure and utilize 5,073 student records from UNM to construct matrix A for latent feature extraction. These features inform D_M, and we subsequently integrate D_G and D_M as per Equation 10, setting $\alpha_1 = 0.3$ and $\alpha_2 = 0.7$. The flexibility in adjusting α_1 and α_2 allows domain experts to explore various restructuring scenarios, providing a valuable "what-if" analysis tool. While optimal values for these coefficients are subject to future exploration, our current focus is on their immediate impact on restructuring outcomes. Figure 11 vividly illustrates the various tree structures derived using the seven latent tree algorithms. Table 5 provides an in-depth performance comparison of these algorithms, with a particular emphasis on CLNJ, which emerges as the most effective in terms of log-likelihood and BIC scores. However, it's crucial to note that CLNJ, alongside CL and *reg*NJ, did not introduce any hidden nodes, limiting their capacity to uncover significant structural insights. Conversely, NJ and CLRG introduced fewer hidden nodes while not topping the BIC scores. This can be particularly advantageous when the goal is to simplify the latent tree structure or to identify a minimal set of meaningful hidden variables that elucidate the dependencies of observed variables. NJ and CLRG significantly lowered the structural complexity compared to the original pattern in Figure 11a, with complexity values of 40 and 46, respectively, against the original's 60. Figures 11c and 11f showcase these two algorithms' latent tree structures. A key observation is that while NJ yields a structure with low complexity, CLRG presents a more coherent tree structure, maintaining crucial prerequisite dependencies, particularly for the foundational *Precalculus* course. CLRG's suggested hidden node (Figure 11f) symbolizes a new course incorporating a subset of learning activities crucial as prerequisites for courses like *Physics I*, *Calculus II*, and *Circuits I*. This addition encapsulates essential learning outcomes and reduces unnecessary complexities, as evident from the 14-point reduction in complexity. The resemblance between the domain expert-designed pattern (Section 5.2) and the latent tree generated by CLRG is striking (Figure 12). When we consider combining the hidden node from Figure 11f with *Precalculus* to form a new course (*Engineering 101*), the similarity becomes even more pronounced. This unity validates the latent tree models' reliability and underscores the logical rationale underpinning their algorithms. An additional critical finding pertains to the improvement in 5-term graduation rates. As Table 5 indicates, there's a significant increase in the probability of students graduating within five terms when following the CLRG-restructured



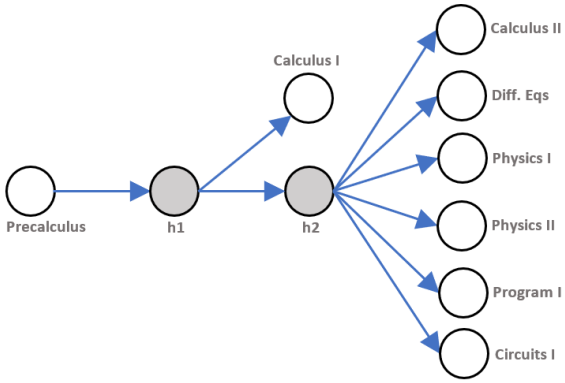
(a) Original pattern



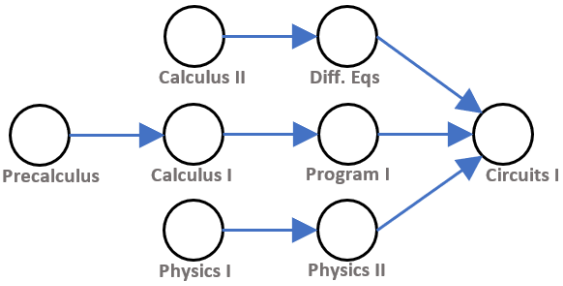
(b) CL



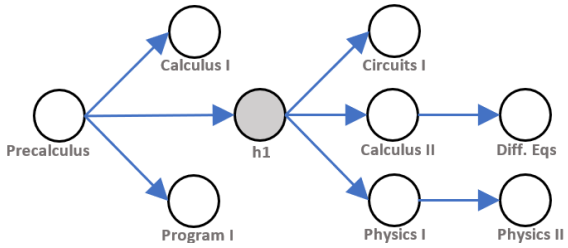
(c) NJ



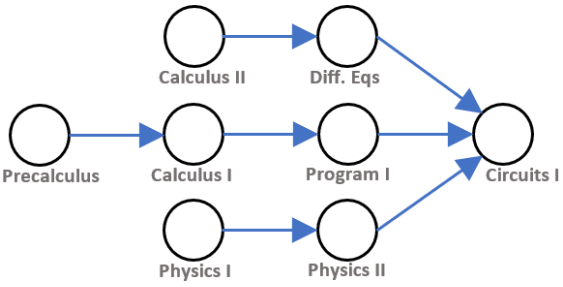
(d) RG



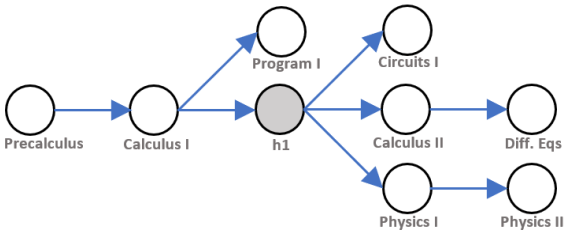
(e) CLNJ



(f) CLRG



(g) *reg*CLNJ

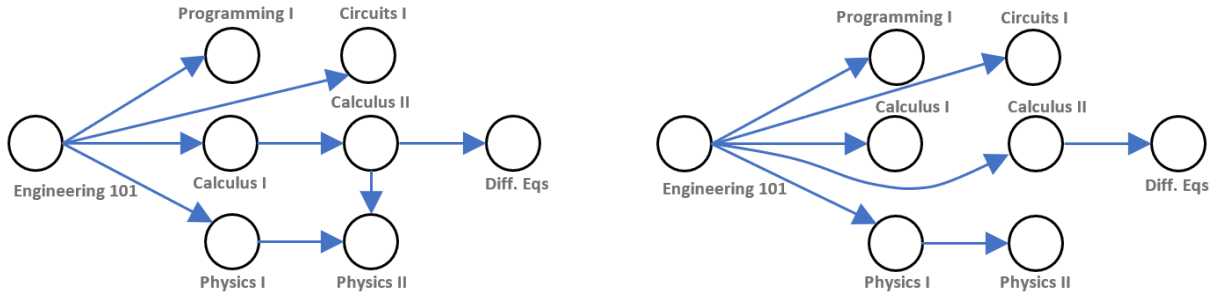


(h) *reg*CLRG

Figure 11: The seven revised curricular patterns generated using the seven latent tree learning algorithms: CL, NJ, RG, CLNJ, CLRG, *reg*CLNJ, and *reg*CLRG.

	Log-Likelihood	BIC	Complexity	Gradation	Hidden
CL	-61038	-61068	40	4.62	0
NJ	-62812	-62846	40	3.2	1
RG	-55614	-55652	62	0.86	2
CLNJ	-54768	-54798	40	4.62	0
CLRG	**-66574**	**-66608**	**46**	**2.23**	**1**
regCLNJ	-59621	-59651	40	4.62	0
regCLRG	-63884	-63918	64	0.20	1
original	-	-	**60**	**0.42**	-

Table 5: The performance of the seven tree learning algorithms used to restructure the curricular pattern of Figure 11a.



(a) Restructured by a domain expert

(b) Restructured using the CLRG method

Figure 12: A prevalent curriculum structure shared by electrical, mechanical, and computer engineering programs.

pattern compared to the original. The 5-term graduation probability increases by approximately 1.8% with a 14-point complexity reduction in the CLRG-generated pattern. This uplift is even more pronounced in larger curricular patterns, underscoring the potential widespread impact of such restructuring. However, it's important to acknowledge that these experiments are predicated on a fixed pass/fail rate assumption for all courses, set at 0.7 for simplicity. Variations in these rates could yield different outcomes, highlighting the importance of considering diverse academic performance scenarios in future analyses.

Building upon the initial exploratory study, our next venture involves applying the latent tree learning algorithms to a larger curricular scope, specifically the computer engineering program at UNM. This extensive curricular pattern omits social sciences and humanities courses, focusing on more complex subjects with prerequisite dependencies. The original pattern for this study (Figure 14a) comprises 26 courses, embodying 34 prerequisite links and a structural complexity of 295, reflecting a dense and intricate curriculum layout. To derive the first *information distance* matrix (D_G), we utilized the graph structure of this pattern. A comprehensive dataset involving 5,409 UNM student records was employed to calculate the second *information distance* matrix (D_M). Mirroring our previous approach, we set $\alpha_1 = 0.3$ and $\alpha_2 = 0.7$. The performance metrics

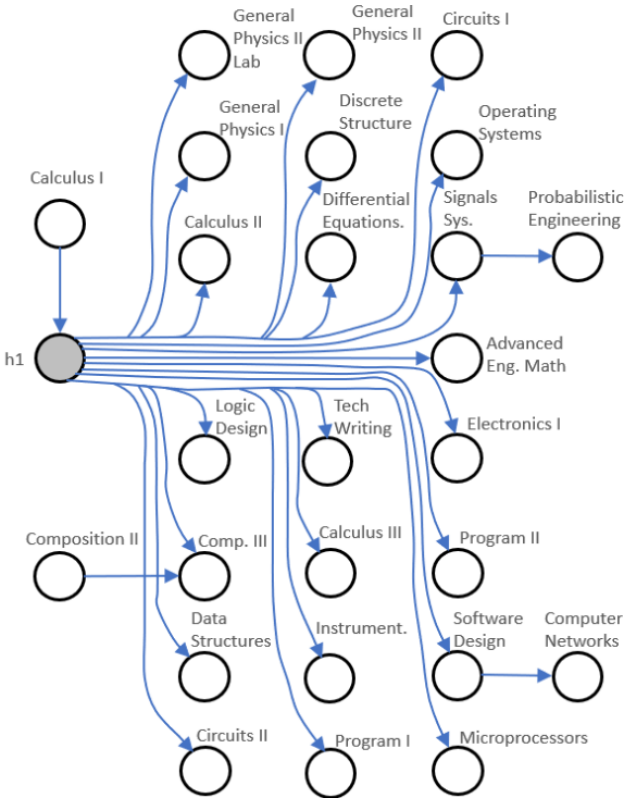


Figure 13: The RG algorithm's restructured version of the UNM computer engineering program.

of the seven tree learning algorithms, as shown in Table 6, reveal diverse outcomes. Although achieving the best BIC score, the CL algorithm did not introduce any new nodes and thus lacked structural insights. While demonstrating better log-likelihood and BIC scores, both NJ and RG models oversimplified the curricular structure, which is evident in their significantly lower complexity values (175 and 138, respectively) compared to the original 295. Figure 13 illustrates an RG-generated pattern, simplifying the structure to an impractical extent by introducing only one new course prerequisite to nearly all courses. Conversely, the CLRG model, despite having the lowest log-likelihood, suggests an oversimplified pattern, reflected in its low complexity value of 117. On the other hand, while showing better fitting scores, the *reg*CLNJ model results in a more complex pattern than the original, with a complexity value of 308. Balancing log-likelihood and complexity, the CLNJ and *reg*CLRG methods emerge as more desirable. They introduce multiple meaningful hidden nodes, providing insights into potential pathways for curricular restructuring. For example, the *reg*CLRG model suggests a new course (*h1*) as a shared prerequisite for courses with common themes, such as English and technical writing courses. In comparing the reconstructed patterns of CLNJ and *reg*CLRG with the original (Figure 14a), the *reg*CLRG model appears more practical and a superior alternative to the original. It maintains essential prerequisite dependencies and introduces new courses like *h2* and *h3*, consolidating prerequisite learning outcomes for key engineering and computer science courses. This restructuring fulfills the program's prerequisite requirements and decreases its structural complexity by 97 points. This complexity reduction directly translates into improved graduation rates. As indicated in Table 6, using Equation 3, the 8-term graduation probability increases by approximately 5.42% with a

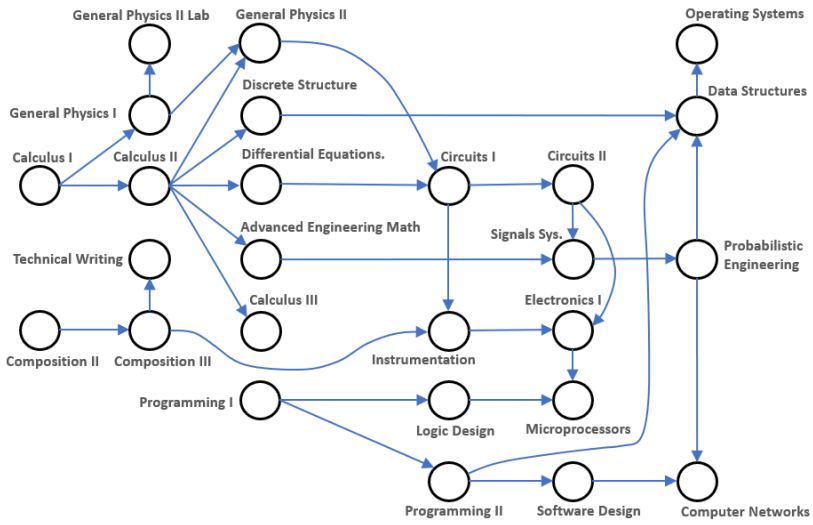
	Log-Likelihood	BIC	Complexity	Graduation	Hidden
CL	-148237	-148345	227		0
NJ	-154251	-154367	175		1
RG	-154738	-154850	138		1
CLNJ	-162063	-162183	212		2
CLRG	-182395	-182511	117		1
regCLNJ	-159950	-160061	308		1
regCLRG	**-177351**	**-177476**	**198**	**12.62**	**3**
original	-	-	**295**	**7.2**	-

Table 6: The performance of the seven tree learning algorithms used to restructure the computer engineering program of Figure 14a.

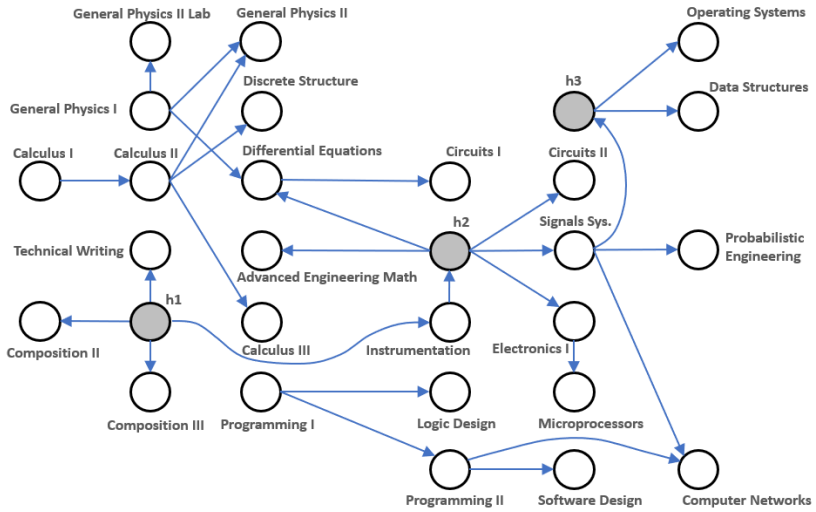
97-point complexity reduction in the *reg*CLRG-generated pattern. Such findings underscore the significant potential of applying these methodologies to large-scale curricular patterns, enhancing educational programs' efficiency and effectiveness.

11 Discussion

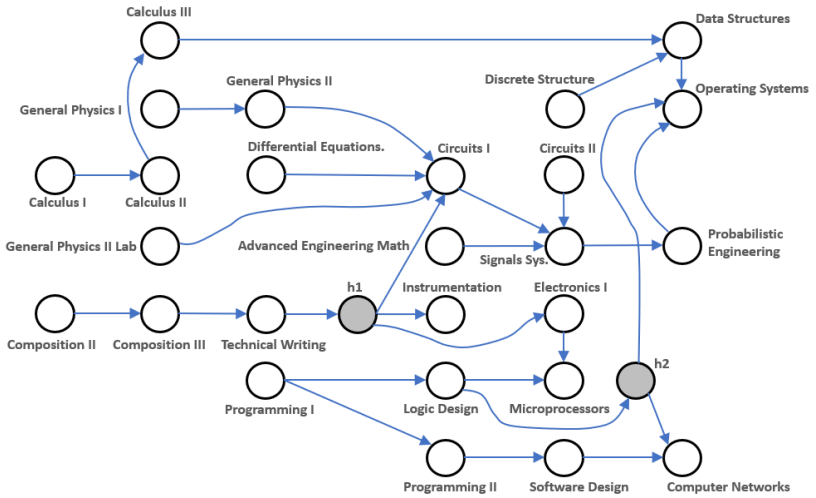
The exploration of curricular reform using latent tree graphical models in this study offers a nuanced understanding of the complexities involved in educational program design and the potential for data-driven decision-making to enhance student outcomes. Our comprehensive analysis encompassed two distinct experiments targeting different curricular scales, from a smaller 8-course pattern to the extensive Computer Engineering program at the University of New Mexico (UNM). Through the application of various latent tree algorithms, we observed the effectiveness of these models in simplifying curricular complexity while maintaining, or even enhancing, the educational integrity of the programs. Notably, the Chow-Liu Recursive Grouping (CLRG) method in the first experiment and the regularized Chow-Liu Neighbor-Joining (*reg*CLNJ) method in the second experiment emerged as the most effective techniques. They offered a balanced approach to curricular restructuring, reducing complexity scores significantly while introducing new courses that encapsulate essential learning outcomes. This balance is particularly noteworthy, as it addresses the critical need to simplify curricular pathways for students without diluting academic rigor or learning objectives. An intriguing outcome of our study was the close resemblance between the restructured patterns derived from the latent tree models and those designed by domain experts. This similarity not only attests to the reliability of these models but also highlights their potential to uncover underlying logical structures within curricula. Such insights can be invaluable for academic institutions seeking to redesign their programs in ways that are both logically sound and beneficial to student progression. Moreover, our findings underscore the potential for significantly improving student graduation rates through curricular reform. We observed marked improvements in projected graduation rates by reducing structural complexities, particularly in more extensive programs. This aspect of the research points to the profound impact that data-driven curricular restructuring can have on student success, providing a compelling case for educational institutions to adopt such approaches. The use of parameters (α_1 and α_2) in combin-



(a) The computer engineering program at UNM



(b) The revised computer engineering program using the regCLRG learning algorithm



(c) The revised computer engineering program using the CLNJ learning algorithm

Figure 14: The computer engineering program at UNM restructured using the *reg*CLRG and the CLNJ learning algorithms.

ing information distance matrices introduces an element of customization, allowing institutions to tailor the restructuring process according to their specific contexts and objectives. Future research focusing on optimizing these parameters could yield even more refined and effective curricular structures. In summary, our study demonstrates the efficacy of latent tree graphical models as a tool for curricular reform, providing a methodological framework that is both innovative and practical. The implications for higher education are significant, offering a pathway to more accessible, navigable, and effective educational programs aligned with the student body's evolving needs and diverse backgrounds. As educational institutions continue to seek ways to improve student outcomes and adapt to changing educational landscapes, the insights from this study could prove instrumental in guiding these efforts.

12 Conclusion

In our research, we introduce an innovative approach to streamline university curriculum restructuring, aiming to significantly lower degree programs' structural complexities. This method is designed to enhance student progression through their academic pathways, thereby potentially boosting graduation rates. Our methodology is grounded in a data-centric approach that leverages actual student records and real-world university curricular structures, eliminating the need for direct intervention by academic experts. The core of our approach involves deploying advanced machine learning techniques and latent tree graphical models, mainly focusing on collaborative filtering methods. We rigorously tested these methodologies on a typical curriculum pattern across electrical, mechanical, and computer engineering departments. The performance of these new curricular structures was critically assessed using various experimental metrics, including Log-Likelihood estimates and the Bayesian Information Criterion (BIC) estimates. Additionally, we employed Markov Decision Processes (MDP) models to gauge the graduation rates of the newly proposed curricular patterns, comparing these with the rates from existing structures. Our findings reveal a marked efficiency in the revised curricular patterns over traditional ones, characterized by reduced structural complexity and a higher potential for improved graduation outcomes. The effectiveness of our model was corroborated using an extensive dataset comprising over 5,000 student records from the University of New Mexico, underscoring our approach's practical applicability and relevance in real-world academic settings. This study offers a significant contribution to educational data analytics and presents a pragmatic solution for universities aiming to optimize their curricular designs in line with student success.

References

[1] B. Dietz-Uhler and J. E. Hurn, "Using learning analytics to predict (and improve) student success: a faculty perspective," *Journal of Interactive Online Learning*, vol. 12, no. 1, pp. 17–26, 2013. [Online]. Available: http://www.ncolr.org/jiol/issues/pdf/12.1.2.pdf

[2] F. Marbouti, H. A. Diefes-Dux, and K. P. C. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," *Comput. Educ.*, vol. 103, pp. 1–15, 2016.

[3] M. D. Pistilli and K. E. Arnold, "Purdue signals: Mining real-time academic data to enhance student success," *About Campus*, vol. 15, no. 3, pp. 22–24, 2010. [Online]. Available: https://doi.org/10.1002/abc.20025

[4] A. Slim, J. Kozlick, G. L. Heileman, J. Wigdahl, and C. T. Abdallah, "Network analysis of university courses,"

in *Proceedings of the 6th Annual Workshop on Simplifying Complex Networks for Practitioners*. Seoul, Korea: ACM, 2014.

[5] C. Abdallah, T. Babbit, and G. Heileman, "The university is a system: The nonlinear impact of various inputs on the institution," *The Evolllution*, 2016.

[6] W. B. Rouse, *Universities as Complex Enterprises: How Academia Works, Why It Works These Ways, and Where the University Enterprise Is Headed*. Hoboken, NJ: Wiley, 2016.

[7] A. Slim, J. Kozlick, G. L. Heileman, and C. T. Abdallah, "The complexity of university curricula according to course cruciality," in *Proceedings of the 8th International Conference on Complex, Intelligent, and Software Intensive Systems*. Birmingham City University, Birmingham, UK: IEEE, 2014.

[8] J. Wigdahl, G. L. Heileman, A. Slim, and C. T. Abdallah, "Curricular efficiency: What role does it play in student success?" in *Proceedings of the the 121st ASEE Annual Conference and Exposition*. Indianapolis, Indiana, USA: IEEE, 2014.

[9] A. H. Slim, G. L. Heileman, J. Kozlick, and C. T. Abdallah, "Employing markov networks on curriculum graphs to predict student performance," in *Proceedings of the 13th International Conference on Machine Learning and Applications*. Detroit, MI: IEEE, 2014.

[10] A. Slim, G. L. Heileman, J. Kozlick, and C. T. Abdallah, "Predicting student success based on prior performance," in *Proceedings of the 5th IEEE Symposium on Computational Intelligence and Data Mining*. Orlando, FL: IEEE, 2014.

[11] A. Slim, G. L. Heileman, E. Lopez, H. A. Yusuf, and C. T. Abdallah, "Crucial based curriculum balancing: A new model for curriculum balancing," in *2015 10th International Conference on Computer Science Education (ICCSE)*, July 2015, pp. 243–248.

[12] A. Slim, G. L. Heileman, W. Al-Doroubi, and C. T. Abdallah, "The impact of course enrollment sequences on student success," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, March 2016, pp. 59–65.

[13] A. Slim, G. L. Heileman, M. Hickman, and C. T. Abdallah, "A geometric distributed probabilistic model to predict graduation rates," in *2017 IEEE Cloud Big Data Computing (CBDCom)*, Aug 2017, pp. 1–8.

[14] A. Slim, D. Hush, T. Ojha, , C. T. Abdallah, G. L. Heileman, and G. El-Howayek, "An automated framework to recommend a suitable academic program, course and instructor," in *Proceedings of the Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*. San Francisco, CA, USA: IEEE, 2019.

[15] 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," *CoRR*, vol. abs/1811.09676, 2018.

[16] G. L. Heileman, M. Hickman, A. Slim, and C. T. Abdallah, "Characterizing the complexity of curricular patterns in engineering programs," in *2017 ASEE Annual Conference and Exposition*. Columbus, Ohio: ASEE Conferences, 2017.

[17] A. Slim, "Curricular analytics in higher education," Ph.D. dissertation, The University of New Mexico, 2016.

[18] M. Hickman, "Development of a Curriculum Analysis and Simulation Library with Applications in Curricular Analytics," Master's thesis, The University of New Mexico, 2017.

[19] B. F. Mon, A. Wasfi, M. Hayajneh, and A. Slim, "A study on role of artificial intelligence in education," in *2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, 2023, pp. 133–138.

[20] A. Wasfi, B. F. Mon, M. Hayajneh, A. Slim, and N. A. Ali, "Optimizing assessment placement and curriculum structure through graph-theoretic analysis," in *2023 15th International Conference on Innovations in Information Technology (IIT)*, 2023, pp. 93–97.

[21] B. Fahad Mon, A. Wasfi, M. Hayajneh, A. Slim, and N. Abu Ali, "Reinforcement learning in education: A literature review," *Informatics*, vol. 10, no. 3, 2023.

[22] *Criteria for Accrediting Engineering Programs, 2017 – 2018*, Accreditation Board for Engineering and Technology (ABET), 2017, available from ABET, http://www.abet.org. [Online]. Available: http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2017-2018/

[23] Q. Hu and W. Yue, *Markov Decision Processes With Their Applications*. Springer, Boston, MA: Springer, 2008, vol. 14.

[24] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov decision processes: A tool for sequential decision making under uncertainty," *Medical Decision Making*, vol. 30, no. 4, pp. 474–483, 2010, pMID: 20044582. [Online]. Available: https://doi.org/10.1177/0272989X09353194

[25] National Center for Education Statistics, *Classification of Instructional Programs*, U.S. Department of Education, National Center for Education Statistics, Washington, DC, 2020, available from National Center for Education Statistics, https://nces.ed.gov/ipeds/cipcode/.

[26] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. New York: Oxford University Press, 1977.

[27] K. Beck and W. Cunningham, "Using pattern languages for object oriented programs," in *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, 1987. [Online]. Available: http://c2.com/doc/oopsla87.html

[28] N. W. Klingbeil and A. Bourne, "The wright state model for engineering mathematics education: Longitudinal impact on initially underprepared students," in *2015 ASEE Annual Conference & Exposition*. Seattle, Washington: ASEE Conferences, June 2015.

[29] M. J. Choi, V. Y. F. Tan, A. Anandkumar, and A. S. Willsky, "Learning latent tree graphical models," 2010. [Online]. Available: https://arxiv.org/abs/1009.2722

[30] "Learning with hidden variables," 2002. [Online]. Available: https://ocw.mit.edu/courses/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/38455cca24675871c5796e67a27cf3b4_Lecture18FinalPart1.pdf

[31] L. Song, H. Liu, A. P. Parikh, and E. P. Xing, "Nonparametric Latent Tree Graphical Models: Inference, Estimation, and Structure Learning," 1 2014. [Online]. Available: https://kilthub.cmu.edu/articles/journal_contribution/Nonparametric_Latent_Tree_Graphical_Models_Inference_Estimation_and_Structure_Learning/6475967

[32] N. L. Zhang, "Hierarchical latent class models for cluster analysis," *J. Mach. Learn. Res.*, vol. 5, p. 697–723, dec 2004.

[33] S. Harmeling and C. K. I. Williams, "Greedy learning of binary latent trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1087–1097, 2011.

[34] K. A. Heller and Z. Ghahramani, "Bayesian hierarchical clustering," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 297–304. [Online]. Available: https://doi.org/10.1145/1102351.1102389

[35] R. Mourad, C. Sinoquet, N. L. Zhang, T. Liu, and P. Leray, "A survey on latent tree models and applications," *Journal of Artificial Intelligence Research*, vol. 47, pp. 157–203, may 2013. [Online]. Available: https://doi.org/10.1613%2Fjair.3879

[36] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky, "Learning high-dimensional markov forest distributions: Analysis of error rates," 2010. [Online]. Available: https://arxiv.org/abs/1005.0766

[37] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, *Collaborative Filtering Recommender Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 291–324. [Online]. Available: https://doi.org/10.1007/978-3-540-72079-9_9

[38] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.