

https://dspace.org/

Optimising curriculum design using Bayesian structure learning

Mandlazi, Thabo

2025
University of the Witwatersrand, Johannesburg
https://hdl.handle.net/10539/50250
Mandlazi, Thabo. (2025). Optimising curriculum design using Bayesian structure learning [Master's dissertation, University of the Witwatersrand, Johannesburg]. WIReDSpace. https://hdl.handle.net/10539/50250

Downloaded from a DSpace repository. https://dspace.org/

Optimising curriculum design using Bayesian structure learning

Ethics Waiver No: HRECNMW25/08/02

Thabo Mandlazi
Supervised by Prof. Ritesh Ajoodha and Dr. Ashwini Jadhav

July, 2025

Submitted to the Faculty of Science,
in fulfilment of the requirements for the degree of
Master of Science in Computer Science
in the



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

ExplainableAI Lab
— MODELLING. DECISION MAKING. CAUSALITY —

Abstract

Mathematics education is vital in developing students' analytical and problem-solving skills. However, in South African schools, many students opt for Mathematical Literacy over Pure Mathematics, often due to the latter being perceived as overly abstract and challenging. This study investigates the potential of Bayesian Networks and score-based structure learning algorithms to optimise curriculum design, making Pure Mathematics more accessible, structured, and student-centered.

Mathematical concepts were modelled as a Bayesian Network, enabling the extraction of prerequisite relationships using probabilistic graphical models. The study evaluated the performance of several structure learning algorithms, including the Chow-Liu algorithm, the Tabu Search algorithm, the Simulated Annealing algorithm, and the Hill Climbing algorithm. The accuracy of the learned structures was assessed against a known ground truth using Structural Hamming Distance (SHD) and Kullback–Leibler (KL) Divergence.

Results show that learned structures consistently recover the main prerequisite structures, with SHD and KL Divergence improving as sample size increases. Sparse learners (Hill Climbing, Chow–Liu) yield interpretable backbones that align well with expert expectations and are effective for auditing scope and sequence. Denser searches (Tabu, Simulated Annealing) surface additional cross-topic dependencies that highlight remedial and enrichment routes, although they're less simple. Applied to ASSISTments data, the pipeline confirms meaningful prerequisite relations. It repeatedly elevates "connector" concepts (e.g., factorisation skills, numerical representation, symmetry) as influential bridges whose targeted reinforcement can lower downstream difficulty.

This research provides a foundation for data-driven curriculum modelling that ensures students master foundational topics before progressing to more advanced content. By leveraging adaptive learning frameworks informed by structural learning, educators can personalise instruction based on learners' knowledge gaps, reduce cognitive overload, and enhance engagement. This approach holds particular promise in South Africa, where it can support efforts to shift perceptions of Pure Mathematics from abstract and intimidating to relevant and achievable.

Limitations include sensitivity to skill tagging, priors, and static modelling of time. Overall, BN structure learning emerges as a viable instrument for evidence-aligned curriculum design, supporting iterative "learn–validate–refine" improvement cycles in mathematics education.

Keywords: Bayesian Networks; Structure learning; Curriculum Design; Machine Learning; ASSISTments; Structural Hamming Distance; Kullback–Leibler Divergence; Mathematics Education.

Declaration

I, Thabo Joel Mandlazi, declare that this thesis, titled *Optimising Curriculum Design Using Bayesian Structure Learning*, is my original work and has been conducted under the guidance and supervision of Prof. Ajoodha and Dr. Jadhav.

I affirm that all sources of information and data used in this thesis have been adequately acknowledged. Any assistance received during this research has been fully credited.

I confirm that this thesis has not been submitted, in whole or in part, to any other institution for the award of any degree or diploma. I also declare that this work complies with the academic integrity policies of The University of the Witwatersrand, Johannesburg.

Signature: Mandlazi

Date: 2026-02-09

Acknowledgment

I would like to express my heartfelt gratitude to my supervisors, Prof. Ajoodha and Dr. Jadhav, whose guidance, insights, and encouragement have been invaluable throughout this research journey. Their expertise has significantly shaped the direction and depth of this work, and I am sincerely thankful for their mentorship.

A special thank you also goes to Palisa Mandlazi for her continuous support and to Daniel Nthoka for his unwavering motivation.

Contents

Abstract **2**

Declaration **3**

Acknowledgment **4**

List of Figures **9**

1 Introduction **10**

2 Related Work **13**
2.1 Extracting Curriculum Sequences Using Graphical Models 13
2.2 Bayesian Networks in Education 14
2.3 Structure Learning Techniques in Bayesian Networks 15
2.4 The Use of Synthetic Data for Structure Learning 16
2.5 Using the ASSISTments Dataset 17
2.6 Evaluating Distributions with Kullback-Leibler Divergence and Structure Hamming Distance 18
2.6.1 Kullback-Leibler (KL) Divergence 18
2.6.2 Structural Hamming Distance (SHD) 19
2.7 Identified Gaps and Contributions 20
2.8 Conclusion 21

3 Methodology **24**
3.1 Problem Statement 25
3.2 Experiment Setup 25
3.3 Ground Truth Generation 26
3.3.1 Ground Truth Model 26
3.3.2 Synthetic Data Generation 27
3.4 Data Preprocessing 28
3.5 Structure Learning Models 29
3.5.1 Hill Climbing Algorithm 29
3.5.2 Tabu Search Algorithm 30
3.5.3 SA algorithm (BIC & K2) 30
3.5.4 Chow–Liu Algorithm 31
3.6 Evaluation Methods 31
3.6.1 Kullback-Leibler (KL) Divergence 32
3.6.2 Structure Hamming Distance (SHD) 32
3.7 Conclusion 33

4 Results and Discussion: Implementation of the Probabilistic Structure Learning Experiment 34
4.1 Ground Truth . . . 34
4.2 Ground Truth Data . . . 35
4.3 Sparse vs. Dense Bayesian Network Structures . . . 35
4.3.1 Sparse Graph . . . 35
4.3.2 Dense Graph . . . 36
4.4 Synthetic Data . . . 37
4.4.1 Generating Synthetic Data for Different Sample Sizes . . . 37
4.4.2 Initialising Conditional Probability Distributions (CPDs) . . . 37
4.4.3 Gaussian Weight Matrix in Defining Probabilistic Dependencies . . . 37
4.5 Structure Learning Models Learned . . . 39
4.5.1 Chow–Liu . . . 39
4.5.2 Hill Climbing with BDeu (K2 Score) . . . 40
4.5.3 Hill Climbing with BIC . . . 40
4.5.4 Tabu Search . . . 41
4.5.5 SA algorithm . . . 41
4.6 Evaluation . . . 41
4.6.1 SHD Analysis . . . 42
4.6.2 KL Divergence Analysis . . . 45
4.7 Discussion . . . 50
4.8 Conclusion . . . 52

5 Results and Discussion: Curriculum Extraction through Bayesian Network Structure Learning 54
5.1 Overview of Methodology . . . 54
5.2 Curriculum Extraction . . . 54
5.2.1 Ground Truth Curriculum Structure . . . 55
5.2.2 Extraction of Curriculum using Tabu Search Algorithm . . . 56
5.2.3 Extraction of Curriculum using HC algorithm . . . 58
5.2.4 Extraction of Curriculum using Chow-Liu (CL) Algorithm . . . 60
5.2.5 Extraction of Curriculum using Simulated Annealing Algorithm . . . 62
5.2.6 Comparative analysis of Structure Learning Algorithms . . . 64
5.3 Discussion . . . 65
5.4 Conclusion . . . 66

6 Results and Discussion: Curriculum Design through Bayesian Network Structure Learning on real-world Data 68
6.1 Dataset . . . 68
6.2 Algorithms Applied . . . 69
6.2.1 Hill Climbing with BIC . . . 69
6.2.2 Tabu Search with BDeu . . . 69
6.2.3 Chow–Liu (CL) . . . 69
6.3 Extraction of Curriculumn . . . 70
6.4 hill . . . 70
6.5 Tabu . . . 74
6.6 Chow-Liu Algorithm . . . 78
6.7 Discussion . . . 80
6.7.1 Alignment with Known Sequences . . . 80
6.7.2 Algorithmic Differences and Trade-offs . . . 82
6.7.3 Implications for Curriculum Design . . . 82
6.8 Conclusion . . . 83

7 Conclusion **84**

References **100**

List of Figures

1.1 Comparison of Total Students and Percentage Achieving 30% or Above in Mathematical Literacy and Mathematics (2021–2023). 11

4.1 Visualisation of 10 distinct Directed Acyclic Graphs (DAGs), labelled as DAG 1 to DAG 10. Each graph illustrates nodes and directed edges representing dependencies and relationships among the nodes within their respective structures. 36

4.2 Illustrate the distribution of CPD values, helping to visualise how probabilities vary across states. 38

4.3 Enhances interpretability of dependency strengths of Gaussian-distributed weight. 40

4.4 SHD vs. Sample Size for a Sparse Graph Across Structure Learning Algorithms. The figure illustrates a consistent decrease in SHD as sample size increases, indicating improved recovery of prerequisite relationships. This trend demonstrates that larger learner datasets lead to more stable and reliable curriculum structures, supporting the validity of Bayesian Network models for curriculum mapping and optimisation. 42

4.5 SHD vs. Sample Size for a Dense Graph Across Structure Learning Algorithms. The figure shows a consistent decrease in SHD as sample size increases, indicating improved recovery of prerequisite relationships in complex curriculum structures. However, the higher SHD values compared to sparse graphs highlight the increased difficulty of learning dense dependency networks, emphasising the need for larger datasets when modelling complex curriculum. 44

4.6 SHD vs. Number of Nodes for Sparse Graphs Across Structure Learning Algorithms. The figure shows that SHD increases as graph size grows, indicating greater difficulty in recovering prerequisite relationships in larger curricula. This trend highlights the impact of curriculum complexity on structure learning accuracy and emphasises the need for robust algorithms when modelling extensive concept networks. 45

4.7 SHD vs. Number of Nodes for Dense Graphs Across Structure Learning Algorithms. The figure shows that SHD increases as graph size grows, indicating increasing difficulty in recovering prerequisite relationships within highly interconnected curricula. This trend highlights the impact of curriculum complexity on structure learning accuracy and emphasises the need for larger datasets and robust algorithms when modelling dense concept networks. 47

4.8 KL Divergence vs. Sample Size for Sparse Graphs Across Structure Learning Algorithms. The figure shows a consistent decrease in KL divergence as sample size increases, indicating improved approximation of the true underlying probability distribution. This trend demonstrates that larger learner datasets enhance the reliability of probabilistic modelling and support more accurate representation of prerequisite relationships in curriculum mapping. 48

4.9 KL Divergence vs. Sample Size for Dense Graphs Across Structure Learning Algorithms. The figure illustrates a consistent decrease in KL divergence as sample size increases, indicating improved approximation of the underlying probability distribution in highly interconnected curricula. However, the comparatively higher values relative to sparse graphs highlight the increased difficulty of probabilistic modelling in dense concept networks, emphasising the need for larger datasets to achieve reliable curriculum representations. 50
4.10 KL Divergence vs. Number of Nodes for Sparse Graphs Across Structure Learning Algorithms. The figure shows that KL divergence increases as graph size grows, indicating reduced accuracy in approximating the true probability distribution for larger curricula. This trend highlights the increasing difficulty of probabilistic modelling as the number of mathematical concepts expands, emphasising the importance of scalable learning methods for curriculum optimisation. 51
4.11 KL Divergence vs. Number of Nodes for Dense Graphs Across Structure Learning Algorithms. The figure shows that KL divergence increases as graph size grows, indicating reduced accuracy in approximating the true probability distribution within highly interconnected curricula. This trend highlights the increasing difficulty of probabilistic modelling in dense concept networks and emphasises the need for scalable and data-intensive approaches for reliable curriculum representation. 53

5.1 Ground Truth Directed Acyclic Graph (DAG) represents the mathematical curriculum's hierarchical structure. 55
5.2 Curriculum DAG extracted using the Tabu Search algorithm 56
5.3 DAG extracted using the HC algorithm. The graph illustrates the learned curriculum structure, and the accompanying table identifies common and missing edges relative to the ground truth. 59
5.4 DAG extracted using the CL algorithm. Structure captures dependencies between concepts with a simplified representation. 61
5.5 Curriculum DAG extracted using the Simulated Annealing algorithm. 63

6.1 Hill Climbing, Top 20 Nodes by In-Degree 70
6.2 Hill Climbing, Top 20 Nodes by Centrality. 72
6.3 Tabu Search (TS), Top 20 Nodes by In-Degree. 74
6.4 Tabu Search (TS), Top 20 Nodes by Centrality. 76
6.5 Chow–Liu (CL), Top 20 Nodes by In-Degree. 78
6.6 Chow–Liu (CL), Top 20 Nodes by Centrality. 80

1 This figure represents the ground truth graph, a reference structure showcasing the true dependencies among variables in the dataset. The orange nodes indicate individual variables, and the directed edges define the prerequisite relationships or causal dependencies. This graph is the benchmark for evaluating the accuracy and fidelity of learned graphical models. The structure reflects the explicit relationships that guide the curriculum or sequence of dependencies for the analyzed concepts. 89
2 illustrates a complete Chow-Liu model. 90
3 illustrates a complete Hill Climbing with BIC. 91
4 illustrates a complete Chow-Liu. 92
5 illustrates a complete Simulated Annealing 93

Chapter 1

Introduction

In South Africa, Grade 10 learners face a critical academic decision between Mathematical Literacy and Pure Mathematics, a choice that significantly impacts their educational opportunities and career prospects. Mathematical Literacy is functional for disciplines such as the humanities, social sciences, and vocational training, but it typically restricts access to professional programs, such as engineering, medical sciences, and computer science, that require Pure Mathematics as a prerequisite for admission [Machaba and Du Plooy 2019].

Recent trends indicate a steady increase in preference for Mathematical Literacy. While this trend supports students in meeting the minimum requirements for school completion, it limits pathways into mathematics-intensive tertiary programs. Conversely, the number of learners opting for Pure Mathematics has declined, raising concerns about long-term skills shortages in STEM fields [ADvTECH Group 2023; Sebatana 2023].

Several factors influence the preference for Mathematical Literacy over Pure Mathematics. Learners often perceive Pure Mathematics as difficult, sometimes due to gaps in foundational knowledge or previous negative experiences [Li *et al.* 2021]. Furthermore, mathematical concepts are frequently introduced in ways that lack structured sequencing. This can make the subject intimidating, leading to lower performance and reduced confidence.

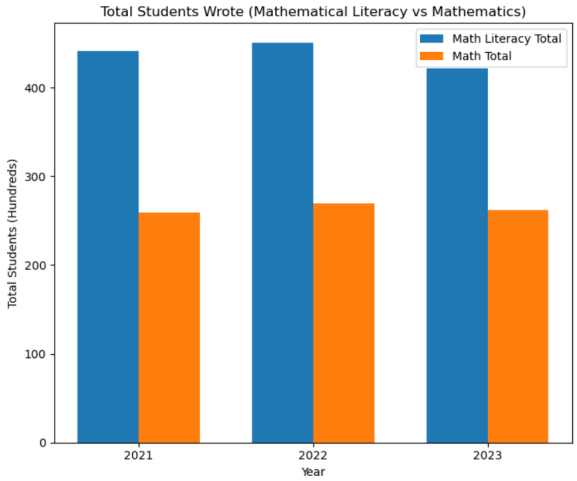
Although recent statistics indicate an increase in the number of learners enrolled in Pure Mathematics, this growth remains modest and uneven across regions and schools. Consequently, concerns persist regarding sustained participation and progression in the subject, particularly when compared to Mathematical Literacy. While the Department of Basic Education (DBE) considers a 30% pass mark as the minimum requirement for both Mathematical Literacy and Pure Mathematics, South African universities typically require substantially higher levels of achievement in Pure Mathematics. As a result, learners who opt for Mathematical Literacy face limited access to many university programmes [of Basic Education and Training 2023].

Statistical data further illustrates the gap. Pass rates for Mathematical Literacy at 30% and above have risen steadily, from 74.50% in 2021 to 85.70% in 2022 and 82.30% in 2023. In contrast, the pass rate for Pure Mathematics has fluctuated, with 57.60% in 2021, 55.00% in 2022, and 63.50% in 2023 [of Basic Education and Training 2023].

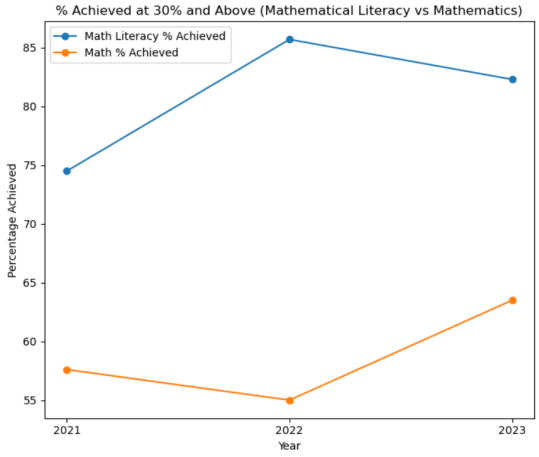
Figure 1.1 compares the total number of students and the percentage achieving 30% or higher in Mathematical Literacy and Mathematics from 2021 to 2023. The bar graph depicts the total student numbers alongside the percentages of those achieving 30% or above in both subjects. In contrast, the line graph illustrates the trends in the percentage of students meeting this benchmark over the three years. This

visualisation is valuable for analysing trends and performance changes over time.

Table 1.1 highlights this trend, showing a steady increase in students opting for Mathematical Lit-



(a) This bar chart compares the total number of students who wrote Mathematical Literacy and Mathematics exams from 2021 to 2023.



(b) This line chart illustrates the (%) of students who achieved 30% and above in Mathematical Literacy and Mathematics over the years 2021 to 2023.

Figure 1.1: Comparison of Total Students and Percentage Achieving 30% or Above in Mathematical Literacy and Mathematics (2021–2023).

eracy, rising from 441,067 in 2021 to 450,005 in 2022 and 421,835 in 2023. Conversely, the number of students sitting for the Pure Mathematics exam has declined, dropping from 259,143 in 2021 to 269,734 in 2022 and 262,016 in 2023.

Table 1.1: Number and percentage of students achieving $\geqslant$30% in Mathematical Literacy and Mathematics (2021–2023)

Year	**Subject**	**Total Candidates**	**Candidates $\geqslant$30%**	**% $\geqslant$30%**
2021	Mathematical Literacy	441 067	328 382	74.5%
	Mathematics	259 143	149 177	57.6%
2022	Mathematical Literacy	450 005	385 515	85.7%
	Mathematics	269 734	148 346	55.0%
2023	Mathematical Literacy	421 835	347 227	82.3%
	Mathematics	262 016	166 337	63.5%

Existing curriculum designs in South Africa are often based on expert judgment rather than data-driven models. While this experience is valuable, it may not optimally sequence mathematical concepts to support incremental knowledge building. Research in Bayesian structure learning suggests that probabilistic models can identify dependencies between concepts, allowing for a curriculum structure that supports more effective learning [Koller and Friedman 2009a]. Despite these potential benefits, few studies have applied Bayesian network methods to determine prerequisite relationships and compare them systematically.

This study aims to apply Bayesian Network structure learning to extract and evaluate curriculum structures from both synthetic and real-world educational datasets, with the broader goal of optimising mathematics curriculum design. The central focus is on modelling the structural dependencies among 200 mathematical concepts, identifying optimal prerequisite pathways, and validating these pathways against

a ground truth curriculum model.

To achieve this aim, the study sets out several objectives. First, it seeks to model dependencies between mathematical concepts using a range of structure learning algorithms, including the Hill Climbing (HC) algorithm with BIC, the Tabu Search (TS) Algorithm with BDeu, the SA algorithm, and the Chow–Liu (CL) Algorithm. These models are evaluated using quantitative metrics such as Structural Hamming Distance (SHD) and Kullback–Leibler (KL) Divergence, which measure structural and distributional similarity to the ground truth. Second, the study aims to extract prerequisite pathways from the learned networks and compare them with established curricular sequences, such as the progression from arithmetic to algebra to calculus. Third, it compares the results derived from synthetic data with those obtained from real-world data (ASSISTments), thereby highlighting differences in stability, interpretability, and robustness. Finally, it evaluates the implications of the extracted structures for curriculum improvement, with particular attention to identifying overlooked but influential concepts that are underrepresented in formal curricula.

Guided by these objectives, the study addresses three overarching research questions. The first concerns the approximation of ground truth: to what extent can the ground truth distribution be approximated by learning associations between mathematical concepts using probabilistic structure learning, compared to randomly generated associations as the number of training samples increases? The second focuses on curriculum extraction from synthetic data: to what extent can structure learning algorithms extract a curriculum that lists explicit prerequisite dependencies comparable to those in the ground truth Bayesian network? The third investigates curriculum extraction from real-world data: to what extent can we extract a curriculum from authentic educational data using structure learning algorithms, and how does the extracted curriculum compare to the sequence of prerequisite concepts in the ground truth model?

By addressing these research questions, the study not only demonstrates how Bayesian Networks can identify and evaluate prerequisite relationships but also provides a data-driven approach to sequencing mathematical concepts. In doing so, it offers insights into how probabilistic modelling can support the optimisation of curriculum structures in ways that enhance comprehension, learning progression, and ultimately student performance.

This chapter introduced the rationale for the study, emphasising the limitations of traditional curriculum design and the potential of Bayesian Network structure learning to address these gaps. By framing the curriculum as a directed acyclic graph of interdependent concepts, the study aims to recover prerequisite relationships in a systematic and data-driven way. Three guiding research questions were established: the extent to which probabilistic structure learning can approximate ground truth distributions; the ability of structure learning algorithms to extract curricula from synthetic data; and their effectiveness in recovering curricula from real-world educational datasets. Collectively, these questions highlight the dual goals of validation and discovery, confirming established sequences while uncovering overlooked dependencies that can enhance learning progressions.

The chapter also outlined the overall structure of the dissertation. Chapter 2 reviews related literature, focusing on curriculum sequencing, probabilistic modelling, and evaluation metrics such as SHD and KLD. Chapter 3 details the methodology, including data preprocessing, model design, and evaluation procedures. Chapter 4 presents the implementation of probabilistic structure learning experiments using synthetic data, followed by Chapter 5, which explores curriculum extraction from the learned models. Chapter 6 applies the framework to real-world data from the ASSISTments platform. Finally, Chapter 7 concludes the study by synthesising findings, discussing implications for curriculum design, and suggesting directions for future research.

Chapter 2

Related Work

The field of curriculum design has undergone a significant transformation with the integration of data-driven methodologies, particularly Bayesian structure learning. Traditional curriculum models often relied on static frameworks, which lacked the adaptability to cater to diverse learner needs and dynamically changing educational goals. Recent advancements in educational technology have enabled intelligent systems that optimise curriculum design by leveraging large datasets and sophisticated algorithms.

This literature review examines the methodologies employed in curriculum optimisation and the role of Bayesian structure learning in education.

Key evaluation metrics, including Kullback-Leibler Divergence and Structural Hamming Distance, are analysed for their role in assessing model performance and structure alignment. Through this analysis, the literature review provides a comprehensive understanding of how contemporary methodologies can enhance curriculum design, making it more effective, personalised, and scalable.

2.1 Extracting Curriculum Sequences Using Graphical Models

The use of graphical models for extracting curriculum sequences has gained traction due to their ability to model knowledge dependencies in an interpretable and structured manner. Graphical models, such as Bayesian Networks and Markov models, have been applied to sequence learning, curriculum design, and knowledge tracing to represent prerequisite relationships between concepts. These models allow for a formal representation of statistical dependencies, that is, probabilistic relationships that describe how mastery or acquisition of one concept influences the understanding of another [Koller and Friedman 2009b; Jensen 2001].

Several studies have contributed to this area, each addressing different aspects of curriculum sequencing. Probabilistic modelling techniques, in particular, have been instrumental in defining curriculum sequences by leveraging statistical dependencies among concepts. For example, Van der Linden and Glas [2010] proposed a probabilistic approach to sequence learning using Item Response Theory (IRT) to model transitions in student ability. This enables adaptive learning paths tailored to individual performance. [Beck and Chang 2007] introduced Bayesian Knowledge Tracing (BKT) to model students' mastery levels over time, which has influenced adaptive curriculum sequencing.

Gong *et al.* [2010]; Yudelson *et al.* [2013] refined BKT by incorporating individual student parameters, which improved prediction accuracy for knowledge acquisition. Khajah *et al.* [2016] extended cognitive modelling frameworks to adjust curriculum sequences based on predictive analytics dynamically. Baker *et al.* [2008] developed the original Knowledge Tracing model, which laid the foundation for many later

studies. Desmarais and Baker [2012]; Baker *et al.* [2019] provided a comprehensive review of student modelling techniques that inform curriculum sequencing and knowledge state estimation.

While these methods provide a strong foundation for modelling learning progressions, they also have limitations. For instance, traditional BKT approaches often struggle with computational scalability when applied to large concept maps with complex dependencies. This limitation motivates the computational efficiency considerations in this dissertation's modelling approach. Moreover, many studies rely on assumptions of conditional independence between concepts, which may oversimplify real-world learning pathways.

Koedinger *et al.* [2015] proposed an automated curriculum sequencing approach leveraging cognitive mastery models, an instructional strategy that emphasises ensuring students fully understand prerequisite concepts before progressing to advanced topics. While this enhances learning outcomes, it can also result in rigid learning paths that may not suit diverse learner profiles.

Brusilovsky and Millán [2007] reviewed adaptive hypermedia and intelligent tutoring systems that personalise curriculum sequences. Kuldas *et al.* [2014] examined the effects of cognitive load on sequencing concepts optimally for learners, while Goguadze *et al.* [2018] developed concept map–based curriculum sequencing in intelligent tutoring systems. Shute and Rahimi [2017] introduced an adaptive learning framework integrating real-time assessment feedback into curriculum design. Shute and Rahimi [2017] investigated the application of learning analytics to curriculum sequencing by employing data mining and visualisation techniques to track learner interactions, identify performance gaps, and recommend personalised learning paths. Their approach aimed to utilise analytics-driven insights to dynamically adapt sequencing in response to individual learner needs, thereby enhancing engagement and mastery. However, the authors also highlighted significant challenges, including the difficulty of establishing causal links between analytics-based interventions and long-term mastery outcomes, as well as the complexity of integrating these analytics into existing curricula. This raises the broader question of not only whether such approaches can enhance student outcomes, but also through what mechanisms and under what conditions these improvements occur.

2.2 Bayesian Networks in Education

In educational research, several methodologies have been used to model knowledge acquisition and optimise learning pathways. Common approaches include decision trees, hidden Markov models, neural networks, and reinforcement learning algorithms. Each of these methods has strengths in detecting learning patterns and predicting student performance. Among these, Bayesian Networks (BNs) have gained prominence due to their ability to explicitly represent the probabilistic relationships between concepts and to provide interpretable reasoning behind predictions.

A Bayesian Network is a probabilistic graphical model consisting of nodes, which represent random variables (such as mathematical concepts or student skills), and directed edges, which indicate conditional dependencies between these variables. The structure forms a directed acyclic graph (DAG), and each node is associated with a Conditional Probability Distribution (CPD) that quantifies the likelihood of the variable given its parents. This structure enables the BN to model how mastery of one concept influences understanding of related concepts.

The concept of probabilistic dependencies refers to the idea that the probability of a student mastering a concept depends on the mastery of prerequisite concepts. For example, a student's likelihood of correctly solving problems in calculus might increase significantly if they have already mastered alge-

bra. In personalised learning, these dependencies allow the BN to dynamically adjust recommendations, directing the learner towards prerequisite topics when gaps are detected, or advancing them when mastery is demonstrated.

Studies have explored the use of BNs in educational contexts extensively. For example, Conati and VanLehn [2002] implemented BNs in intelligent tutoring systems to model student knowledge and adapt instructional strategies in real-time. Mislevy [2003] applied BNs in large-scale educational assessments to represent uncertainty in student performance. Millán and de-la Cruz [2010] integrated BNs into adaptive learning systems, enabling delivery of content tailored to the learner's needs.

More recent work, such as Yudelson *et al.* [2013], has combined BNs with knowledge tracing models to predict student performance more accurately, particularly for identifying at-risk students. Comparative studies, like Gong and Zhai [2020], have evaluated BNs against deep learning models for educational data mining. While deep learning often achieves strong predictive accuracy, BNs offer greater interpretability, a critical feature in academic settings where understanding the reasoning process is as important as the predictions themselves.

2.3 Structure Learning Techniques in Bayesian Networks

Structure learning methods for Bayesian networks play a crucial role in uncovering dependencies among variables and inferring causal relationships from data. These methods can be broadly classified into score-based, constraint-based, and hybrid approaches, each with distinct advantages and limitations in terms of computational efficiency, accuracy, and scalability.

Score-based structure learning involves optimising a predefined scoring function to determine the best-fitting network structure. Common scoring metrics include the Bayesian Information Criterion (BIC) and the Bayesian Dirichlet equivalent uniform (BDeu). Scutari [2010] and Constantinou *et al.* [2016] demonstrated that score-based methods are highly effective in high-dimensional datasets, efficiently capturing complex dependencies.

Similarly, Bouckaert [1995] provided foundational insights into scoring-based Bayesian model selection, emphasising the trade-off between model complexity and likelihood estimation. Tsamardinos *et al.* [2006a] proposed the Hill-Climbing algorithm, a popular search-based method that iteratively refines network structures based on score improvements. Recent studies, such as those by Nagarajan *et al.* [2013] and Wang and Sun [2021], have introduced heuristics to improve convergence rates and reduce computational costs, making these methods more feasible for large-scale applications. Despite their effectiveness, score-based methods may suffer from local optima issues, requiring extensive search strategies such as the SA algorithm [de Campos 2006].

Constraint-based techniques infer network structure by detecting conditional independence relations among variables in the dataset. One widely known example is the PC algorithm [Spirtes *et al.* 2000]. The PC algorithm (named after its creators, Peter Spirtes and Clark Glymour) begins by constructing a fully connected undirected graph, then systematically removes edges by testing for conditional independencies. Afterwards, it orients the remaining edges to form a directed acyclic graph, ensuring that no cycles are introduced. The PC algorithm is efficient in large datasets and particularly suitable when causal discovery is the goal. Colombo and Maathuis [2012] and Glymour *et al.* [2019] have focused on improving the efficiency of constraint-based learning by integrating parallel computing techniques and adaptive pruning strategies. However, it remains sensitive to statistical errors in independence tests, which can affect accuracy when data is noisy or sample sizes are small.

Hybrid approaches combine score-based and constraint-based learning elements to strike a balance

between performance and scalability. For instance, Ramsey *et al.* [2012] introduced the Fast Greedy Equivalence Search (FGES) algorithm, which blends greedy optimisation with constraint-based filtering to reduce computational complexity. Yu and Wang [2022] expanded on this by incorporating regularisation techniques that improve convergence in high-dimensional datasets.

Recent studies, such as those by Zhu and Zhao [2020] and Chickering [2002], have explored dynamic structure learning in Bayesian networks, particularly in domains where the structure evolves. These methods demonstrate potential for applications in personalised learning systems by adapting to changing curriculum demands.

Despite significant advancements in Bayesian network structure learning, challenges remain. While Bayesian networks have been extensively applied in fields such as genetics and economics, their use in education remains relatively less common. Many educational studies focus on predictive modelling of student performance rather than extracting prerequisite structures from educational data [Doroudi *et al.* 2019; Rafferty *et al.* 2016].

2.4 The Use of Synthetic Data for Structure Learning

Synthetic data has played a crucial role in probabilistic modelling, particularly in evaluating and validating structure learning algorithms for Bayesian networks and other probabilistic graphical models. By providing controlled environments where ground truth structures are known, synthetic datasets enable researchers to systematically assess the performance of learning algorithms in ways that real-world data often cannot.

Koller and Friedman [2009b] emphasised the importance of synthetic data in probabilistic graphical models, noting that it allows for unbiased benchmarking of learning algorithms. Similarly, Scutari and Denis [2021] explored the application of synthetic datasets in Bayesian network learning, demonstrating how such datasets aid in understanding algorithmic biases and performance variations across different conditions.

Hellewell *et al.* [2020] investigated the robustness of structure learning algorithms using synthetic epidemic datasets, highlighting how synthetic data allows for stress testing under diverse outbreak scenarios. In the educational domain, Stamper and Barnes [2010] examined the use of synthetic student data to model learning progressions and prerequisite relationships, revealing how artificially generated student behaviours can inform instructional design.

Furthermore, Chickering [2002] provided a foundational study on the implications of synthetic data for Bayesian structure learning, demonstrating the trade-offs between accuracy and computational complexity. More recently, Gao and Ji [2023] extended this work by incorporating deep generative models to create synthetic data distributions that better mimic real-world dependencies.

Despite these advancements, significant gaps remain in the literature, particularly regarding the extent to which synthetic datasets faithfully represent the complexities of real-world student learning. Many existing synthetic data models rely on overly simplistic assumptions about student behaviours, failing to capture nuances such as latent misconceptions, adaptive learning pathways, and interdependencies among concepts [Van de Sande 2013].

Additionally, while synthetic datasets provide a valuable tool for validating structure learning methods, their limited fidelity to real-world data presents challenges in ensuring the generalizability of findings. For instance, Ritter *et al.* [2022] argued that most synthetic datasets used in educational modelling do not fully replicate student engagement patterns, leading to potential biases in model evaluation. Similarly, Busemeyer and Wang [2021] discussed how cognitive models of learning, when used to generate

synthetic data, often struggle to account for real-world variability in student responses.

To address these challenges, researchers have focused on developing more sophisticated generative models that incorporate real-world data distributions. In comparison, this area has been explored in domains such as medical imaging (e.g.,[Chen *et al.* 2017]); its application to educational data remains comparatively limited. One promising avenue is the integration of probabilistic programming frameworks with empirical student data to refine the assumptions underlying synthetic data generation [Pearl and Mackenzie 2018]. Furthermore, incorporating domain knowledge through expert-driven priors can enhance the validity of synthetic datasets for evaluating structure learning algorithms [Glymour *et al.* 2016].

2.5 Using the ASSISTments Dataset

Several studies in educational data mining and learning analytics have utilised the ASSISTments dataset to examine learner behaviour, knowledge development, and curriculum sequencing. The dataset contains large-scale records of students' interactions with an online mathematics tutoring system, including problem attempts, skill tags, correctness indicators, and temporal information, enabling statistically robust analysis. Previous research has applied techniques such as knowledge tracing, Bayesian modelling, and network-based approaches to identify prerequisite relationships and learning trajectories. Compared with smaller or institution-specific datasets, ASSISTments provides a standardised and well-documented structure that supports reproducibility and cross-study comparison. Its widespread adoption in the literature strengthens its credibility and generalisability, providing a strong empirical foundation for its selection in the present study.

The ASSISTments dataset, first introduced by Selent *et al.* [2016], is a widely utilised resource in educational data mining and learning analytics. It comprises extensive data collected from multiple randomised controlled experiments conducted in online learning environments, providing a robust foundation for evaluating educational interventions, knowledge tracing models, and AI-driven predictive modelling approaches. Due to its structured nature and wealth of interaction data, the dataset has become central to research on intelligent tutoring systems, adaptive learning, and curriculum design.

A key strength of the ASSISTments dataset lies in its detailed student interaction records, which capture timestamps, correctness, and hints used. These fine-grained features enable not only static dependency modelling but also the analysis of dynamic learning behaviours. Such detail is particularly valuable for probabilistic models, such as Bayesian Networks, where both the breadth and depth of student data enhance statistical robustness.

Over the years, ASSISTments has supported the development of several influential knowledge tracing models. For instance, the foundational Deep Knowledge Tracing (DKT) model by Piech *et al.* [2015b] demonstrated the dataset's potential for predicting student learning outcomes. More recent advances, such as the graph neural network approach by Zia *et al.* [2023], revealed how complex student trajectories could be captured from the data. Similarly, Zhang *et al.* [2017] introduced a dynamic key-value memory network for tracking long-term dependencies, while Chen *et al.* [2022] refined student skill assessment through a variational Bayesian knowledge tracing model. Generative AI techniques have also been integrated, with Huang *et al.* [2021] employing adversarial networks to simulate realistic student responses, and Chaturvedi *et al.* [2020] applying reinforcement learning to optimise intelligent tutoring interactions. Together, these studies underscore the dataset's role as both a benchmark and a catalyst for methodological innovation in educational research.

Beyond modelling applications, ASSISTments offers unique features that make it particularly suited for curriculum extraction tasks. Each problem is explicitly tagged with a knowledge component (KC),

providing a direct mapping to mathematical concepts. This enables researchers to represent problems as nodes in a Bayesian Network and to extract prerequisite relationships with greater interpretability. By contrast, while other large-scale datasets, such as EdNet, KDD Cup, or MOOC-based logs, provide breadth of data, they often lack explicit concept annotations, which limit their applicability for curriculum design studies.

The dataset also holds a distinctive position as a benchmark resource within the field of educational data mining. Foundational models, including DKT Piech *et al.* [2015b], graph-based approaches Zia *et al.* [2023], and Bayesian methods Chen *et al.* [2022], have all been evaluated using this dataset. Building on such a well-established benchmark not only strengthens comparability with prior work but also ensures that the results of this study are situated within a broader and evolving body of research.

Finally, the choice of ASSISTments over alternatives such as EdNet is further justified by its accessibility and generalizability. Whereas EdNet provides impressive scale, it is highly domain-specific to the Korean educational system, which may limit its applicability in broader contexts. ASSISTments, on the other hand, focuses on widely taught mathematics topics, is openly accessible, and is well-documented. These qualities make it both practical and replicable for academic research. For these reasons, the ASSISTments dataset provides an optimal foundation for extracting and validating curriculum structures using score-based Bayesian structure learning methods.

2.6 Evaluating Distributions with Kullback-Leibler Divergence and Structure Hamming Distance

2.6.1 Kullback-Leibler (KL) Divergence

Kullback–Leibler (KL) divergence is a fundamental concept in information theory that quantifies the difference between two probability distributions [Cover and Thomas 2006; Murphy 2012]. It measures the information lost when using an approximating distribution $Q(x)$ to represent a target distribution $P(x)$, and is defined as:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

(for discrete distributions), or as an integral in the continuous case. The measure is always non-negative and equals zero only when $P(x) = Q(x)$ for all x.

KL divergence has been widely applied in unsupervised learning, model selection, and Bayesian structure learning. For example, Feng and Lee [2017] utilised it to assess linguistic variation in speech processing, while Liu [2024] applied it in anomaly detection by distinguishing between normal and anomalous data distributions. In probabilistic modelling, it is commonly used to evaluate how well an estimated model matches the true distribution, as in variational inference methods [Wang *et al.* 2023] and Bayesian neural network approximations [Choi *et al.* 2022].

Given its ability to provide a principled measure of distributional similarity, KL divergence has become an important tool in evaluating probabilistic models across disciplines, including applications in educational modelling and curriculum sequencing.

Beyond its applications in information theory, the KL divergence is pivotal in optimisation problems. Kobayashi [2022a] reformulated traditional learning laws for value functions and policies as optimisation problems by incorporating the KL divergence into reinforcement learning algorithms. This integration enables the balancing of exploration and exploitation in decision-making processes, thereby

optimising the performance of autonomous systems. Additionally, Kosun [2021] examined vehicular speed characteristics using KL divergence alongside entropy-based methods, revealing its potential in analysing mobility patterns.

KL divergence is frequently used in model selection to compare the suitability of different probabilistic models. Cliff et al. [2018] applied KL divergence to identify optimal models for distributed nonlinear systems, reinforcing its importance in selecting the most informative data representation. Furthermore, Wang *et al.* [2023] extended its usage to variational inference, refining probabilistic graphical models by minimising the divergence between approximate and true posterior distributions. This approach has significantly improved inference accuracy in Bayesian networks and deep learning applications.

The adaptability of KL divergence extends to Bayesian structure learning. Choi *et al.* [2022] demonstrated the effectiveness of this approach in learning probabilistic graphical structures by incorporating KL regularisation terms, which prevent overfitting while ensuring meaningful relationships between variables. Similarly, Ghahramani and Jordan [2023] utilised KL divergence to evaluate Bayesian neural network approximations, ensuring robust and generalizable model predictions.

These applications highlight the role of KL divergence as a foundational tool for comparing and estimating probability distributions, making it indispensable for scientific disciplines that rely on probabilistic inference.

2.6.2 Structural Hamming Distance (SHD)

SHD is a local measure of dissimilarity between two network structures, focusing on the presence or absence of edges within them. It provides a straightforward metric for comparing graphical models, particularly in the context of Bayesian network structure learning. Jurman *et al.* [2015a] demonstrated that SHD offers an effective approach for evaluating learned network structures by quantifying the number of edge additions, deletions, or reversals required to transform one network into another. However, one of SHD's limitations is that it does not capture the global structural properties of networks, prompting the development of more refined metrics.

Researchers have proposed extensions that incorporate structural dependencies and temporal information to address this limitation.Trabelsi *et al.* [2013a] refined SHD for dynamic Bayesian networks, incorporating time-aware adjustments to account for evolving dependencies in longitudinal datasets. Similarly, Mokhtari *et al.* [2022] developed a weighted version of SHD to assess causal inference models, improving sensitivity to edge orientations and indirect influences.

In genomic research, SHD has been applied to privacy-preserving data analysis. Ghasemian *et al.* [2024a] proposed a framework utilising SHD to evaluate genomic similarity, ensuring data security while enabling large-scale analysis. This method aligns with broader trends in privacy-conscious computational biology, such as secure multi-party computation [Zhu *et al.* 2023]. Additionally, SHD has been leveraged for SARS-CoV-2 variant clustering Nazaritiji [2024b], where researchers quantified genetic differences using structural similarity measures. The robustness of SHD in handling large-scale biological datasets underscores its importance in computational genomics.

Barmpas and Tasoulis [2024a] further explored SHD's application in classification tasks, demonstrating its utility in feature selection for graph-based machine learning models. Their findings indicate that SHD-based pruning enhances the interpretability of learned structures by eliminating redundant edges, thus refining model generalisation capabilities. Moreover, Zhao *et al.* [2023] integrated SHD with deep reinforcement learning techniques to improve graph-based decision-making, showing promising results

in autonomous agent planning and robotics.

In educational research, SHD has been employed to evaluate the structure of concept maps in adaptive learning environments. Chen *et al.* [2022] applied SHD to assess differences between student-generated and expert-defined concept networks, providing insights into knowledge acquisition and curriculum sequencing. This application aligns with the growing interest in utilising probabilistic graphical models for educational assessment and curriculum optimisation.

By quantifying structural deviations in learned models, SHD provides a vital evaluation metric for graphical structure learning, supporting advancements in multiple domains, including network analysis, computational biology, and artificial intelligence.

Table 2.1 summarises research papers on graphical models and machine learning in educational setups.

2.7 Identified Gaps and Contributions

The existing body of research has demonstrated the effectiveness of Bayesian Networks and graphical models in modelling student learning and predicting their performance. While structure learning techniques such as score-based, constraint-based, and hybrid approaches have been extensively studied, their application has mainly been limited to fields like genomics, finance, and medical diagnostics, with relatively little exploration in educational research, particularly in curriculum design and concept dependency modelling.

In the context of education, *concept maps* are graphical representations that depict the relationships between concepts within a subject domain, often structured as nodes (concepts) connected by directed edges (prerequisite relationships). For example, in a mathematics curriculum, a concept map may show ”Fractions” as a prerequisite for ”Ratios,” which in turn precedes ”Proportions.” Such maps provide a visual and formal way to capture the sequencing of learning content.

A significant limitation of prior work is the reliance on manually curated Bayesian Network structures, which introduces bias and scalability challenges when dealing with large, high-dimensional educational datasets. In our domain, these datasets may include student performance records, assessment scores, and task completion data, often accompanied by metadata such as timestamps and types of learning activities. Existing methods also struggle with real-world applicability, as they frequently assume complete, noise-free datasets, a condition rarely met in authentic educational settings. While deep learning models have gained popularity in student modelling and adaptive learning, they often lack interpretability, making them less effective for curriculum sequencing, where understanding explicit concept dependencies is essential.

Rather than relying on expert-defined structures, this study employs a data-driven, score-based learning approach (e.g., BIC, BDeu, and K2) to extract dependencies between mathematical concepts, thereby reducing manual bias while enhancing scalability. Unlike previous research, which primarily focuses on predictive modelling of student performance, this work explicitly reconstructs prerequisite relationships using structure learning, aligning data-driven sequences with human-defined curriculum structures.

Most prior studies evaluate structure learning using either Kullback–Leibler (KL) Divergence or SHD in isolation. This research presents a joint assessment framework that integrates both KL divergence and SHD to validate the extracted curriculum structures comprehensively. Given that real-world student data is often incomplete, this study also explores data imputation techniques to improve the robustness of structure learning, addressing a significant limitation in existing research. The findings contribute to educational research and artificial intelligence, offering scalable, interpretable, and adaptive curriculum

structures that more accurately reflect real-world learning pathways.

2.8 Conclusion

This chapter has reviewed the body of literature on curriculum optimisation, with particular attention to the role of Bayesian Network structure learning in educational research. Traditional curriculum design has long relied on fixed, linear frameworks; however, recent advances in data-driven methodologies have made it possible to capture the complexity and variability of student learning more effectively. Graphical models, such as Bayesian Networks, Markov models, and cognitive modelling approaches, have been shown to represent prerequisite relationships between concepts in a structured and interpretable manner, providing a strong foundation for adaptive curriculum design.

Across the literature, evaluation metrics such as Structural Hamming Distance (SHD) and Kullback–Leibler Divergence (KLD) emerge as essential tools for assessing structural fidelity and distributional accuracy. Their application in prior studies has highlighted the strengths and limitations of different algorithms while establishing benchmarks for comparing learned structures to ground truth models. Together, these methods have enhanced the ability to assess how well computational models replicate actual learning dependencies.

Overall, the literature demonstrates that Bayesian structure learning is both a promising and practical approach for curriculum extraction and optimisation. It not only validates canonical learning pathways but also allows for the discovery of overlooked relationships that could enhance sequencing. By synthesising these findings, this chapter has established a foundation for the experimental work that follows, where synthetic and real-world datasets will be used to test the effectiveness of different structure learning algorithms in recovering and refining mathematical curricula.

Table 2.1: Summary of Research Papers on Graphical Models and Educational Data Analysis

Author(s)	**Title**	**Focus/Objective**	**Key Contributions**	**Key Results / Baseline Findings**	**Limitations**
Jurman *et al.* [2015b]	The HIM Glocal Metric and Kernel for Network Comparison and Classification	Develop a metric for comparing and classifying networks.	Introduced the HIM metric.	Achieved improved network similarity classification over traditional distance measures.	Limited validation on diverse real-world networks.
Trabelsi *et al.* [2013b]	Benchmarking DBN Structure Learning Algorithms	Evaluate DBN learning algorithms.	Provided benchmarking framework.	Identified performance trade-offs between DBN algorithms under varying conditions.	Limited testing on noisy real-world data.
Ghasemian *et al.* [2024b]	Safeguarding Privacy in Genome Research	Propose privacy framework.	Developed privacy-preserving model.	Demonstrated secure data sharing with minimal utility loss.	Limited empirical validation.

Author(s)	Title	Focus/Objective	Key Contributions	Key Results / Baseline Findings	Limitations
Zhu *et al.* [2024]	Hybrid Optimisation for Causal Discovery	Improve causal discovery.	Proposed hybrid optimisation model.	Improved causal structure recovery accuracy on benchmark datasets.	Scalability not addressed.
Barmpas and Tasoulis [2024b]	Hyperdimensional Computing in RNA Classification	Apply HDC to genomics.	Demonstrated HDC utility.	Achieved competitive classification accuracy with reduced computational cost.	Limited domain scope.
Nazaritiji [2024a]	SARS-CoV-2 Variant Clustering	Analyse genomic variants.	Provided clustering insights.	Identified distinct variant clusters with high intra-group similarity.	Pathogen-specific.
Selent *et al.* [2016]	ASSISTments Dataset	Create educational dataset.	Released ASSISTments.	Enabled large-scale experimental analysis across multiple studies.	Context-specific dataset.
Zia *et al.* [2023]	GNN Knowledge Tracking	Apply GNNs.	Improved tracking.	Outperformed traditional knowledge tracing models.	Limited cross-dataset testing.
Hui [2024]	AI GEN ASSISTment	Use generative AI.	Enhanced DKT.	Improved prediction accuracy over standard DKT.	Overfitting risk.
Piech *et al.* [2015a]	Deep Knowledge Tracing	Predict learning.	Developed DKT.	Achieved state-of-the-art prediction accuracy.	Limited interpretability.
Yaramala [2023]	Student Behaviour Analysis	Analyse behaviour.	Linked behaviour to scores.	Improved score prediction performance.	Platform constraints.
Razzaza *et al.* [2007]	Transfer Models in ASSISTments	Skill assessment.	Developed transfer models.	Improved skill mastery estimation.	Outdated datasets.
Si [2021]	COVID-19 Learning Recovery	Mitigate learning loss.	Supported recovery.	Demonstrated measurable post-intervention improvement.	Short-term study.
Pedro *et al.* [2013]	College Enrollment Prediction	Predict enrollment.	Long-term modelling.	Achieved significant early prediction accuracy.	Limited generalisation.

Author(s)	Title	Focus/Objective	Key Contributions	Key Results / Baseline Findings	Limitations
Du [2022]	Curriculum Optimisation System	Optimise curriculum.	Decision system.	Improved curriculum alignment with learner outcomes.	No cross-institution testing.
Mekić *et al.* [2024]	AI Curriculum Model	Optimisation model.	Ethical focus.	Demonstrated improved decision support.	Lacks implementation detail.
Backenkoeller *et al.* [2018]	Personalised Curriculum	Personalisation.	ML-based pathways.	Improved learner engagement metrics.	Requires large datasets.
Liang *et al.* [2018]	MentorNet	Curriculum learning.	Efficient training.	Reduced training error and convergence time.	Limited paradigm testing.
Bucci *et al.* [2023]	Curriculum Learning Systems	Model dynamics.	Improved accuracy.	Enhanced predictive stability.	Narrow context.
Boutilier and Chan [2023]	ML in OR Curriculum	Curriculum integration.	Teaching framework.	Improved student performance outcomes.	Single-course case study.
Song *et al.* [2024]	Feedback-Driven Learning	Optimise pathways.	Adaptive framework.	Increased learning efficiency.	Scalability unclear.
Shen [2024]	Talent Cultivation ML	Optimise training.	Personalised learning.	Improved competency development metrics.	Higher education focus.
Yang and Wei [2024]	Digital Literacy Assessment	Assess literacy.	ML assessment.	Improved digital literacy scores.	Domain-specific.

Chapter 3

Methodology

This study adopts a quantitative experimental research methodology, combining computational simulation and empirical data analysis to evaluate Bayesian Network structure learning algorithms for curriculum optimisation. Controlled experiments were conducted using both synthetic and real-world datasets, and performance was assessed using established structural and probabilistic metrics.

In this research, structure learning plays a central role in uncovering relationships among mathematical concepts. The objective is to identify how concepts interrelate and form hierarchical prerequisite sequences. To achieve this, Bayesian Network (BN) structure learning algorithms are employed to infer directed acyclic graphs (DAGs) from the data, thereby representing underlying dependency structures.

The aim of this study is to investigate the extent to which Bayesian Network structure learning algorithms can be used to optimise mathematics curriculum sequencing by identifying prerequisite relationships among mathematical concepts. To fulfil this aim, the study constructs ground-truth curriculum models, generates synthetic datasets, and preprocesses real-world learner data for systematic evaluation. Selected score-based Bayesian Network algorithms are applied to learn dependency structures from the data, and their performance is assessed using established structural and probabilistic metrics, including Structural Hamming Distance and Kullback–Leibler divergence. The study further extracts and interprets curriculum pathways from the learned networks to examine their relevance for curriculum planning and learning progression. Finally, the effectiveness of different structure learning approaches is compared under varying data conditions, including sparse and dense network settings.

This chapter presents the methodological framework used to implement and evaluate this approach. It begins with the problem statement and experimental setup, followed by detailed descriptions of the learning algorithms, evaluation metrics, ground truth generation, and synthetic data creation. The methodology is designed to be transparent, systematic, and reproducible, ensuring alignment with the research questions outlined in the introduction.

Ethical considerations

This study conducted a secondary analysis of de-identified *ASSISTments* data and received an ethics *waiver* from the University of the Witwatersrand, Johannesburg Research Ethics Committee (Ref: **HREC-NMW25/08/02**). All procedures complied with institutional policy and applicable data protection laws (e.g., POPIA). No re-identification was attempted.

3.1 Problem Statement

In educational research and curriculum design, understanding the relationships between mathematical concepts is essential for structuring a practical learning pathway. Suppose we have a data sample $\mathcal{D} = \{D_1, D_2, ..., D_n\}$ that provides information about the associations between a set of mathematical concepts $\mathcal{C} = \{C_1, C_2, ..., C_n\}$. The challenge is determining how to construct and validate relationships that indicate an association between these mathematical concepts based on the given data.

The goal of this study is to investigate whether score-based Bayesian Network structure learning algorithms can approximate the ground-truth distribution of concept dependencies and extract prerequisite sequences that align with the true curriculum structure.

3.2 Experiment Setup

Probabilistic graphical models (PGMs) provide a principled framework for representing complex probability distributions. By combining graph theory with probability theory, they provide a compact representation of a joint distribution, where the graph structure captures the conditional dependencies among variables. This separation of structure and parameters makes it easier to reason about, manipulate, and learn the underlying distribution.

In this study, we adopt a simulation-based pipeline to evaluate structure-learning methods. We first construct a ground-truth Bayesian network comprising 200 concepts, which serves as the reference model of dependencies. From this model, we generate synthetic datasets at multiple sample sizes via sampling from the ground-truth distribution, thereby creating controlled conditions with known structure. We then apply structure-learning algorithms to each dataset to recover network structures from data alone. Finally, we assess the learned structures against the ground truth using quantitative criteria, primarily topology, focused measures (e.g., Structural Hamming Distance) and distributional fit metrics, summarising performance across sample sizes. This design enables a transparent, apples-to-apples comparison of how well different algorithms reconstruct the dependency structure encoded in the ground-truth model. The following research questions guide the experiments:

Research Question: To what extent can we approximate the ground truth distribution by learning the associations between mathematical concepts using probabilistic structure learning, compared to randomly generated associations as the number of training samples increases?

Experiment:
We Let P^* represent the true underlying probability distribution over a set of mathematical concepts $\mathcal{C} = \{C_1, C_2, ..., C_n\}$. Suppose we have a Bayesian network $\mathcal{B}^*$ which is faithful to $\mathcal{B} = \langle \mathcal{G}^*, \Theta \rangle$. Assume that the mathematical associations are accurately represented in the conditional independence relations in $\mathcal{G}^*$. We then sample a dataset $\mathcal{D} = \{\xi_1, \xi_2, ..., \xi_n\}$ from $\mathcal{B}^*$ using forward sampling, where each ξ_i corresponds to an instance in which concept associations manifest.

We apply structure learning algorithms to the dataset $\mathcal{D}$ to recover a structure and parameter setting $\mathcal{B}^* = \langle \mathcal{G}, \Theta \rangle$, where $\mathcal{G}$ represents the set of concept associations $\mathcal{I}(C_i \perp\!\!\!\perp C_j \mid Z)$, where $Z \in \{C_1, C_2, ..., C_n\}, Z \notin \{C_i, C_j\}$. We will utilise the greedy HC algorithm, along with BIC and BDE scores, as well as the CL algorithm, to learn concept associations. We will use a random structure with learned parameters obtained via Maximum Likelihood Estimation (MLE) as a baseline. Finally, to evaluate the effectiveness of our concept association learning procedure, we will plot the KL divergence (relative to $\mathcal{B}^*$) of the output from each structure learning algorithm as the number of training samples increases.

Research Question: To what extent can we extract a curriculum using structure learning algorithms from a learned graphical structure that explicitly lists the dependencies of any chosen concept, which is comparable to the sequence of prerequisite concepts stated in the ground truth Bayesian network, as the number of training samples increases?

Experiment:
We let $\mathcal{B}^* = \langle \mathcal{G}^*, \Theta \rangle$, represent the ground truth Bayesian network over a set of mathematical concepts $\mathcal{C} = \{C_1, C_2, ..., C_n\}$, with each concept potentially depending on others in a hierarchical, prerequisite-based structure. This network serves as our baseline model, where $\mathcal{G}^*$ is a Directed Acyclic Graph (DAG) encoding the true prerequisite sequences. We then sample a dataset $\mathcal{D} = \{\xi_1, \xi_2, ..., \xi_n\}$ from $\mathcal{B}^*$ using forward sampling, where each ξ_i represents an instance of concept mastery or occurrence. This dataset reflects the dependencies among concepts defined by the ground truth structure.

We apply structure learning algorithms on $\mathcal{D}$ to recover a network $\mathcal{B}^+ = \langle \mathcal{G}^+, \Theta^+ \rangle$, where $\mathcal{G}^+$ is the learned structure that may capture concept dependencies. We extract potential prerequisite sequences from $\mathcal{G}^+$ by identifying directed paths between concepts. For each concept $\mathcal{C}_i$, we list its parent concepts as inferred prerequisites within $\mathcal{G}^+$. To evaluate the learned structure $\mathcal{G}^+$ in terms of curriculum extraction, we compare the extracted prerequisites against the ground truth prerequisite sequences in $\mathcal{G}^*$. We measure the alignment between the extracted and actual prerequisites using SHD.

Research Question: To what extent can we extract a curriculum from real-world educational data using structure learning algorithms, and how does the extracted curriculum compare to the sequence of prerequisite concepts?

Experiment:
We use ASSISTment dataset, a real-world educational data $\mathcal{D} = \{\xi_1, \xi_2, ..., \xi_n\}$, where ξ_i represents an instance of student performance or mastery of mathematical concepts $\mathcal{C} = \{C_1, C_2, ..., C_n\}$. Apply structure learning algorithms to $\mathcal{D}$ to recover a $\mathcal{B}^+ = (\mathcal{G}^+, \Theta^+)$. Use the learned graph $\mathcal{G}^+$ to infer prerequisite relationships between concepts by identifying directed paths or parent-child relationships. Extract the inferred curriculum from $\mathcal{G}^+$ where for each concept $\mathcal{C}_i$ its parent nodes in $\mathcal{G}^+$ represent its prerequisite concepts.

3.3 Ground Truth Generation

To evaluate the performance of structure learning algorithms, it is necessary to establish a reliable reference point against which learned models can be compared. In this study, that reference is defined as the ground truth, represented by a Bayesian Network that encodes prerequisite relationships among mathematical concepts. The ground truth serves as a controlled model, enabling systematic experimentation without the confounding factors present in real-world data. By generating both the ground-truth model and corresponding ground-truth data, the study creates a robust framework for testing whether learning algorithms can accurately reconstruct the underlying structure of conceptual dependencies.

3.3.1 Ground Truth Model

The ground truth model refers to the reference Bayesian Network that was artificially constructed for this study and serves as the baseline against which all structure learning algorithms are evaluated. In this model, each node corresponds to a unique mathematical concept, thereby ensuring comprehensive coverage of the domain under investigation. Directed edges capture prerequisite relationships among concepts, such that an edge from node A to node B indicates that concept A is a prerequisite for concept B. To avoid structural bias, edges were created randomly under specified constraints that allowed the construction of both sparse and dense graph structures.

In this study, predefined structural constraints were applied to ensure that the generated networks satisfied DAG properties and reflected realistic curriculum structures. Specifically, constraints included prohibiting self-loops, preventing cyclic dependencies, limiting the maximum number of parent nodes per concept, and enforcing a fixed ordering of prerequisite relationships. These constraints are implemented during graph generation to maintain structural validity and interpretability.

Edge densities for sparse and dense networks were controlled through randomised edge sampling under these constraints. For each ground-truth model, potential edges were first identified based on the predefined ordering, after which edges were selected uniformly at random to achieve target density levels. Sparse networks were generated using lower edge probabilities, while dense networks were generated using higher probabilities. This randomisation procedure ensured structural diversity while preserving consistency across experimental conditions.

To guarantee that the structure conformed to the properties of a Directed Acyclic Graph (DAG), a topological sort was applied to detect and remove cycles. Each node was further assigned a Conditional Probability Distribution (CPD), initialised randomly and uniformly, to establish the probabilistic basis for inference within the network. Thus, the ground truth model is not derived from real-world educational data but is instead a controlled, simulated representation encoding the assumed structural and probabilistic relationships among concepts.

To enhance robustness and reduce reliance on a single structural configuration, ten separate ground truth models were generated. Each model consisted of 200 nodes with randomised edge densities and CPDs, thereby capturing a diverse range of potential prerequisite structures. This design mitigates the risk of bias from any single configuration and ensures that the evaluation of algorithms is based on a broad set of relational patterns. By averaging results across these multiple models, the methodology strengthens the validity and generalizability of the findings.

3.3.2 Synthetic Data Generation

The ground truth data refers to the synthetic datasets generated directly from the ground truth model through forward sampling. While the ground truth model encodes the theoretical dependencies between nodes, the ground truth data represents the observable outcomes implied by this structure. These datasets provide the simulated evidence available to a learner attempting to reconstruct the Bayesian Network and therefore serve as the empirical foundation for evaluating the accuracy of structure learning algorithms.

Synthetic data generation began with a topological ordering of nodes in the network to determine sampling order. Nodes without parents (root nodes) were sampled first from their marginal distributions as defined by their CPDs. Once parent nodes were assigned values, the conditional probabilities specified in their CPDs were used to generate values for child nodes. For instance, if node B depended on node A, then A was sampled first, after which B was drawn according to P(B—A). This process continued recursively until all nodes had assigned values, producing a complete observation.

By repeating this forward sampling procedure, synthetic datasets of varying sizes (e.g., 1000, 2000, 5000 observations) were generated. Each dataset reflected the joint probability distribution of the ground truth model while preserving the encoded dependencies among nodes. Forward sampling was selected for its computational simplicity and reliability, ensuring that the data accurately reflected the probabilistic structure of the underlying model.

A detailed pseudocode outlining the construction of ground truth models, the forward sampling process, and the evaluation metrics is provided in Appendix 7.4. This appendix includes the step-by-step procedures used to implement the Bayesian Network simulations and test structure learning algorithms.

3.4 Data Preprocessing

The study used the ASSISTments dataset, a widely adopted online tutoring dataset that records student–problem interactions. Each entry links a student's response to a problem with one or more Knowledge Components (KCs), which represent specific mathematical skills or concepts. The dataset includes key fields such as:

- **User ID** - *The unique identifier of the student associated with the problem.*
- **Skill** - *The skill tested.*
- **Type** - *The category or type of the question.*
- **Correct** - *Binary indicator of whether the student answered correctly (1) or incorrectly (0).*

Table 3.1: ASSISTments Data Columns Used in This Research

user_id	skill	type	correct
61394	Rounding	LinearSection	Correct
61394	Multiplication and Division Integers	LinearSection	In Correct
78401	Equation Solving Two or Fewer Steps	LinearSection	In Correct
78409	Equation Solving Two or Fewer Steps	LinearSection	Correct
80145	Addition and Subtraction Fractions	LinearSection	Correct

After preprocessing, a total of 200 mathematical concepts spanning foundational (arithmetic, fractions), intermediate (algebra, geometry), and advanced (calculus, probability, discrete mathematics) domains were retained.

The ASSISTments dataset was selected for three main reasons: it reflects real-world student learning behaviours, including variability, misconceptions, and noise. It covers a wide range of mathematical topics, enabling the study of curriculum extraction across multiple domains. As a benchmark dataset in educational data mining, it allows findings to be compared against prior studies Lee and Brunskill [2012]. This makes it particularly well-suited for testing the effectiveness of Bayesian Network structure learning in extracting prerequisite relationships.

A structured preprocessing pipeline was implemented to ensure data integrity and suitability for Bayesian modelling. Removed duplicate rows to avoid redundancy and over-representation of certain skill–student pairs. Standardised mappings of problems to skills to ensure consistency. Missing values were imputed using K-Nearest Neighbours (KNN) imputation (k = N chosen through validation). This method was selected because it preserves local data patterns more effectively than mean/median imputation, and it reduces bias in student mastery profiles. Responses were aggregated at the student–skill level to consolidate repeated attempts. The dataset was reshaped into a binary matrix format (rows = students, columns = skills, values = correct/incorrect), which is required for Bayesian structure learning.

Continuous features (e.g., mastery scores) were transformed into discrete categories using equal-width binning. This ensured compatibility with Bayesian algorithms, which estimate conditional probability distributions over discrete states. Verified that no duplicate student–skill pairs remained. Inspected distributions before and after imputation to confirm that original data patterns were preserved. Ensured no missing values remained after preprocessing.

The final dataset was clean, complete, and properly formatted for Bayesian Network modelling. It aligned with the assumptions of the structure learning algorithms, ensuring that results were both valid and reliable. While limitations remain, for example, the dataset reflects only students who interacted with ASSISTments during the specified period, and KNN imputation assumes comparable performance patterns, the preprocessing ensured that the data was robust enough to support the study's objectives.

3.5 Structure Learning Models

Following preprocessing, the cleaned dataset was used as input for learning the Bayesian Network structure. The aim was to identify the most accurate dependency structures between mathematical concepts, representing prerequisite relationships in the curriculum. To achieve this, four algorithms were implemented, each representing a distinct approach to exploring the search space: the HC Algorithm, the TS algorithm, the SA algorithm, and the CL algorithm.

3.5.1 Hill Climbing Algorithm

Hill Climbing (HC) is a greedy local–search procedure for BNs Structure learning that iteratively improves an initial graph by applying single, edge edits that increase a chosen score [Heckerman 1998]. Starting from an empty, random, or fully connected acyclic graph, HC repeatedly explores a neighbourhood composed of acyclicity, preserving operations, *add*, *delete*, or *reverse* one edge. Because the scoring functions used here are *decomposable*, the change in objective can be computed by re-scoring only the local terms for nodes whose parent sets are affected, making each step computationally efficient. At each iteration, the best-improving move is applied; if no move yields an improvement, the search terminates. To mitigate local optima, we employ multiple random restarts and retain the best solution encountered at each iteration. We use the following Scoring functions. We evaluate candidate DAGs with three standard, decomposable scores for discrete Bayesian networks.

Bayesian Information Criterion (BIC) balances data fit with model complexity by penalising the number of free parameters:

$$S_{\mathrm{BIC}}(\mathcal{G} \mid \mathcal{D}) = \log p(\mathcal{D} \mid \hat{\theta}_{\mathcal{G}}, \mathcal{G}) - \frac{k}{2} \log n, \tag{3.1}$$

where $\hat{\theta}_{\mathcal{G}}$ are the maximum–likelihood parameters for graph $\mathcal{G}$, k is the number of independent parameters, and n is the sample size. Higher values indicate better trade-offs between fit and parsimony.

Bayesian Dirichlet equivalent, uniform (BDeu), for a multinomial–Dirichlet model, the BDeu score equals the log marginal likelihood under a uniform Dirichlet prior with equivalent sample size α:

$$S_{\mathrm{BDeu}}(\mathcal{G} \mid \mathcal{D}; \alpha) = \sum_{i=1}^{p} \sum_{j=1}^{q_i} \left[\log \Gamma(\alpha_{ij}) - \sum_{k=1}^{r_i} \log \Gamma(\alpha_{ijk}) + \sum_{k=1}^{r_i} \log \Gamma(\alpha_{ijk} + N_{ijk}) - \log \Gamma(\alpha_{ij} + N_{ij}) \right], \tag{3.2}$$

with $\alpha_{ijk} = \alpha/(q_i r_i)$ and $\alpha_{ij} = \sum_k \alpha_{ijk}$. Here p is the number of nodes, r_i the number of states of node i, q_i the number of parent configurations of i, N_{ijk} the count of state k of node i when its parents take configuration j, and $N_{ij} = \sum_k N_{ijk}$. Larger α encourages smoother (less sparse) conditional distributions.

K2 score is the special case of the Dirichlet–multinomial score with a uniform. Dirichlet prior ($\alpha = 1$); it is often associated with a search that assumes a user-supplied variable ordering. Its log form is

$$S_{\mathrm{K2}}(\mathcal{G} \mid \mathcal{D}) = \sum_{i=1}^{p} \sum_{j=1}^{q_i} \left[\sum_{k=1}^{r_i} \log \Gamma(N_{ijk} + 1) - \log \Gamma\big(N_{ij} + r_i\big) + \log \Gamma(r_i) \right]. \tag{3.3}$$

Like BDeu, K2 is decomposable and therefore amenable to efficient local updates during the search.

Across all three scores, HC moves greedily to the best neighbouring DAG that preserves acyclicity and improves the objective, stopping at a local optimum. Using multiple scoring functions (BIC, BDeu, and K2) allows us to examine how assumptions about model complexity and parameter priors influence the structures recovered from data.

3.5.2 Tabu Search Algorithm

Tabu Search (TS) algorithm augments greedy HC algorithm with a short–term memory that prevents immediate returns to recently explored solutions, thereby reducing cycling and enabling broader exploration of the search space [Glover 1989 1990]. In our setting, a solution is a directed acyclic graph (DAG) $\mathcal{G}$. From the current $\mathcal{G}$ we form a neighborhood $\mathcal{N}(\mathcal{G})$ of acyclicity, preserving single–edge edits (*add*, *delete*, or *reverse*). Each neighbor is scored with a decomposable objective; we use the BDeu score $S_{\text{BDeu}}(\mathcal{G} \mid \mathcal{D}; \alpha)$ with equivalent sample size α, which is robust in sparse or limited–data regimes and allows efficient local updates.

At iteration t, TS selects the best *admissible* move:

$$m_t^\star \;=\; \arg\max_{m \in \mathcal{A}_t} S(\mathcal{G} \oplus m), \qquad \mathcal{A}_t \;=\; \big\{m \in \mathcal{N}(\mathcal{G}) : m \notin \text{Tabu}\big\} \cup \big\{m \in \text{Tabu} : S(\mathcal{G} \oplus m) > S(\mathcal{G}^*)\big\}.$$

Here, $\oplus$ applies the edit m, Tabu is the set of moves forbidden for a fixed *tabu tenure* (a small number of iterations), and $\mathcal{G}^*$ is the best network encountered so far. The second set implements the *aspiration criterion*: a tabu move is permitted if it improves upon the incumbent best score. After applying $m_t^\star$, TS records its inverse (or defining attributes) in the tabu list, updates $\mathcal{G}^*$ if improved, and proceeds to the next iteration. The search terminates when a stopping rule is met (e.g., maximum iterations, time budget, or no improvement over a patience window), returning $\mathcal{G}^*$.

In this study, pairing TS with the BDeu score yielded a robust heuristic for recovering denser prerequisite structures, particularly in noisy or data-constrained settings where the pure HC algorithm can converge prematurely. TS's memory mechanism and aspiration criterion make it especially effective in large, rugged spaces where avoiding local optima is crucial.

3.5.3 SA algorithm (BIC & K2)

The SA algorithm is a probabilistic global search method inspired by thermal annealing [Kirkpatrick *et al.* 1983]. Unlike purely greedy procedures, SA occasionally accepts moves that worsen the objective according to a temperature parameter T that decreases over time. A high initial T promotes exploration and helps the search escape local optima; as T cools, acceptance of deleterious moves becomes rarer and the search becomes increasingly exploitative.

In our application, a candidate solution is a directed acyclic graph (DAG). The algorithm starts from an initial network $\mathcal{G}_0$ and temperature T_0. At each iteration, it proposes a neighbouring DAG $\mathcal{G}'$ by applying a small, acyclicity-preserving edit, *add*, *remove*, or *reverse* a single edge. The change in objective

$$\Delta \;=\; \text{Score}(\mathcal{G}') - \text{Score}(\mathcal{G})$$

It is evaluated using either the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC), also known as the A A A K2 score. The move is then accepted with probability.

$$P(\text{accept}) \;=\; \begin{cases} 1, & \text{if } \Delta > 0, \\ \exp\big(\Delta/T\big), & \text{otherwise.} \end{cases}$$

After each proposal, the temperature is reduced according to a cooling schedule (e.g., geometric $T \leftarrow \alpha T$ with $\alpha \in (0, 1)$). The procedure terminates when a stopping rule is met (e.g., minimum temperature,

maximum iterations, or no improvement for a fixed window), and it returns the best-scoring network encountered during the process.

Using SA with both BIC and K2 complements deterministic heuristics, such as the HC algorithm and the TS algorithm. By explicitly trading off exploration and exploitation, SA can traverse diverse regions of the search space and is particularly useful when the score landscape is rugged or high-dimensional, providing a stochastic benchmark for stability and recoverability of structure in our experiments.

3.5.4 Chow–Liu Algorithm

The Chow–Liu (CL) procedure learns a tree-structured Bayesian network by approximating the joint distribution with the maximum–weight dependence tree [Chow and Liu 1968]. It is computationally efficient and yields interpretable hierarchical structures, making it a useful baseline when speed and transparency a priority, albeit with the limitation that it cannot express multi-parent (dense) dependencies.

Let $\{C_1, \ldots, C_p\}$ denote discrete concepts (variables). The method first quantifies pairwise dependence using mutual information (MI),

$$I(C_i; C_j) = \sum_{c_i} \sum_{c_j} p(c_i, c_j) \, \log \frac{p(c_i, c_j)}{p(c_i)\, p(c_j)} \,, \tag{3.4}$$

where $p(c_i, c_j)$ is the joint empirical probability and $p(c_i), p(c_j)$ are the marginals. A complete undirected graph is then formed with nodes $\{C_i\}$ and edge weights $w_{ij} = I(C_i; C_j)$. The core optimisation A selects the maximum–weight spanning tree (MWST)

$$T^\star = \arg\max_{T \in \mathcal{T}_p} \sum_{(i,j) \in E(T)} I(C_i; C_j), \tag{3.5}$$

where $\mathcal{T}_p$ is the set of spanning trees on p nodes and $E(T)$ the edge set of T. In practice, Kruskal's or Prim's algorithm recovers $T^\star$ in $O(p^2 \log p)$ after MI computation.

To obtain a directed acyclic model, CL chooses an arbitrary root r and orients all edges in $T^\star$ away from r, yielding a tree-structured Bayesian network in which each node has at most one parent. Parameters are estimated by empirical conditionals $p(c_i \mid \mathrm{pa}(c_i))$ computed from the data, So the final model factors as

$$p(c_1, \ldots, c_p) = p(c_r) \prod_{i \neq r} p(c_i \mid \mathrm{pa}(c_i)) \,. \tag{3.6}$$

In this study, CL serves as a fast, interpretable comparator that highlights the trade-off between simplicity and structural completeness. While it cannot capture converging prerequisite patterns with multiple parents, it provides a clear hierarchical backbone of dependencies and a strong baseline for evaluating more expressive search-based learners.

3.6 Evaluation Methods

Once a Bayesian Network (BN) is learned using structure learning algorithms, the next critical step is to evaluate how well the inferred structure aligns with the ground truth. In this study, two key evaluation methods are used: Kullback-Leibler (KL) Divergence, a probabilistic measure that quantifies the difference between the true and learned probability distributions, and SHD, a graph-theoretic measure that compares the structural differences between the ground truth network and the inferred network.

By combining these two evaluation metrics, the study ensures that the learned Bayesian Network is assessed both quantitatively (in terms of probability distribution accuracy) and structurally (in terms of graph similarity to the ground truth).

3.6.1 Kullback-Leibler (KL) Divergence

In BN, each node represents a random variable and directed edges encode conditional dependencies. When we learn structure from data, our aim is to approximate the unknown joint distribution with a model distribution $q(\mathbf{x})$ induced by the learned graph and its parameters. To quantify how close this approximation is to a reference distribution $p^*(\mathbf{x})$ (e.g., the ground truth model), We use the Kullback-Leibler (KL) divergence [Kullback and Leibler 1951]:

$$\mathrm{KL}(p^* \,\|\, q) \;=\; \sum_{\mathbf{x}} p^*(\mathbf{x}) \,\log \frac{p^*(\mathbf{x})}{q(\mathbf{x})} \;=\; \mathbb{E}_{\mathbf{X}\sim p^*}\big[\log p^*(\mathbf{X}) - \log q(\mathbf{X})\big]. \tag{3.7}$$

KL is non–negative and equals zero iff $p^* = q$ almost everywhere; it is asymmetric and thus not a true metric. Lower values indicate a better distributional match.

For discrete Bayesian networks, both p^* and q factorize over the graph:

$$\log p^*(\mathbf{x}) = \sum_{i=1}^{p} \log p^*(x_i \mid \mathrm{pa}_i^*)\,, \qquad \log q(\mathbf{x}) = \sum_{i=1}^{p} \log q(x_i \mid \mathrm{pa}_i)\,, \tag{3.8}$$

where pa_i^* and pa_i are the parent configurations under the ground truth and learned structures, respectively. This decomposition lets us evaluate log probabilities efficiently for any configuration $\mathbf{x}$. The exact sum in (3.7) is exponential in the number of variables. We Therefore, estimate KL via Monte Carlo by drawing N ancestral samples $\{\mathbf{x}^{(s)}\}_{s=1}^{N}$ from the ground truth BN and computing

$$\widehat{\mathrm{KL}}(p^* \,\|\, q) = \frac{1}{N} \sum_{s=1}^{N} \Big[\log p^*\big(\mathbf{x}^{(s)}\big) - \log q\big(\mathbf{x}^{(s)}\big)\Big]. \tag{3.9}$$

To avoid infinite values when $q(\mathbf{x}) = 0$ but $p^*(\mathbf{x}) > 0$, we apply standard smoothing (e.g., Dirichlet pseudocounts / probability floor ε) when estimating the conditional probability tables (CPTs). All logarithms use the natural base, so KL is measured in nats.

We construct a ground truth BN over the mathematics concepts and compare it to networks learned by the HC algorithm, the TS algorithm, the SA algorithm, and CL algorithm. For each sample size, we report the mean and variability of $\widehat{\mathrm{KL}}(p^* \,\|\, q)$ across random seeds. Lower KL values indicate that the learned structure and parameters better capture the probabilistic relationships encoded by the ground truth.

3.6.2 Structure Hamming Distance (SHD)

Whereas KL divergence evaluates distributional fit, Structural Hamming Distance (SHD) [Tsamardinos *et al.* 2006b] measures purely topological error between a learned Bayesian network and a reference (ground truth) network on the same node set. SHD is the minimum number of elementary edit operations, edge *additions*, *deletions*, and *reversals*, required to transform the learned graph $\mathcal{G}$ into the ground truth $\mathcal{G}^*$. Lower values indicate a closer match in the pattern of dependencies.

Let E^* and E denote the sets of directed edges in $\mathcal{G}^*$ and $\mathcal{G}$, respectively. Define

$$\begin{aligned} M &= \{(i \to j) \in E^* :\ (i \to j) \notin E \ \wedge\ (j \to i) \notin E\} && \text{(missing edges)},\\ X &= \{(i \to j) \in E :\ (i \to j) \notin E^* \ \wedge\ (j \to i) \notin E^*\} && \text{(extra edges)},\\ R &= \{(i, j) :\ (i \to j) \in E^* \ \wedge\ (j \to i) \in E\} && \text{(reversed edges)}. \end{aligned}$$

Then the SHD we report (counting a reversal as one operation) is

$$\mathrm{SHD}(\mathcal{G}^*, \mathcal{G}) \;=\; |M| \;+\; |X| \;+\; |R|. \tag{3.10}$$

$|M|$ captures prerequisite relations present in the ground truth but not recovered; $|X|$ counts spurious relations introduced by the learner; and $|R|$ measures orientation mistakes where the correct adjacency is found, but in the wrong direction. By summarising these components, SHD indicates how faithfully, a learned structure reproduces the dependency skeleton and edge Directions of the ground truth.

We compute $\mathrm{SHD}(\mathcal{G}^*, \mathcal{G})$ for networks learned by the HC algorithm (BIC/BDeu), TS algorithm (BDeu), SA algorithm (BIC/K2), and CL algorithm across sample sizes. Results are averaged over random seeds. Lower SHD signifies better structural recovery of the prerequisite relationships among concepts.

3.7 Conclusion

This chapter detailed the methodology adopted to investigate the extraction of prerequisite relationships between mathematical concepts using Bayesian Network structure learning. The process began with rigorous data preprocessing of the ASSISTments dataset, including data cleaning, handling of missing values through KNN imputation, transformation into a suitable format, and discretisation to meet the model's requirements.

Four distinct structure learning algorithms, the HC algorithm, the TS algorithm, the SA algorithm, and Chow–Liu, were implemented to explore the search space from different optimisation perspectives. The learned models were evaluated against the ground truth using SHD for structural accuracy and KL Divergence for probabilistic similarity.

These steps collectively ensured that the input data was robust, the learning approaches were methodologically diverse, and the evaluation metrics provided a comprehensive assessment of performance. The subsequent chapter presents the results of these experiments, comparing algorithm performance across varying sample sizes and network densities.

Chapter 4

Results and Discussion: Implementation of the Probabilistic Structure Learning Experiment

This chapter presents the results of applying Bayesian Network structure learning algorithms to extract prerequisite relationships between mathematical concepts. The learned structures are compared to a ground truth model, which serves as the reference for evaluating structural and probabilistic accuracy.

The ground truth model is a Directed Acyclic Graph (DAG) of 200 mathematical concepts arranged in a hierarchical curriculum sequence. Each node represents a concept, and directed edges represent prerequisite relationships between concepts. This model was generated using the methodology described in Chapter 3, incorporating both sparse and dense configurations to simulate varying dependency structures.

4.1 Ground Truth

Following the methodology described in Chapter 3, the ground truth model was constructed to approximate the associations between mathematical concepts. This model provides the foundation for analysing dependencies within the domain and serves as a benchmark for evaluating the ability of structure learning algorithms to reconstruct the underlying relationships. By simulating real-world educational dependencies, the ground truth enables a reliable assessment of both structural accuracy and probabilistic performance.

The generated structure encodes prerequisite relationships and dependencies among concepts, ensuring alignment with the expected curriculum sequence. This approach is consistent with prior work, such as Backenkoehler *et al.* [2018], which emphasises the role of data-driven methods in supporting curriculum design.

To create a more reliable and unbiased representation, ten distinct ground truth models were generated, each consisting of 200 nodes and randomised edge configurations. These variations simulate multiple possible dependency structures, reducing the influence of bias from any single model. Each ground truth network reflects different structural properties due to the randomness inherent in the generation process.

To stabilise results and mitigate bias further, the study averaged across the ten generated models. The final ground truth thus reflects the most common dependency structures observed across the different configurations. This approach ensures that a single arbitrary initialisation does not influence evalua-

tion, but instead captures a more generalizable and robust relational pattern among the mathematical concepts.

4.2 Ground Truth Data

In addition to the structural representation provided by the ground truth model, the study also relied on ground truth data, which refers to the synthetic observations generated from the model. These datasets were created through forward sampling, ensuring that the probabilistic dependencies encoded in the Bayesian Network were faithfully preserved. While the ground truth model specifies the theoretical relationships between concepts, the ground truth data represent the observable outcomes implied by those relationships.

For this experiment, multiple datasets of varying sizes (1,000, 2,000, and 5,000 observations) were generated for each of the ten ground truth models. Each dataset reflected the joint probability distribution of its corresponding Bayesian Network, thereby simulating the evidence that would be available to a learning algorithm attempting to reconstruct the structure. By varying the dataset sizes, the study was able to assess how sample size influences the accuracy and stability of structure learning results.

The use of synthetic ground truth data provides a controlled setting for evaluation. It allows the study to isolate the performance of structure learning algorithms from potential noise and inconsistencies that are often present in real-world datasets. This ensures that the results reported in subsequent sections can be attributed directly to the algorithms' ability to recover structure, rather than to extraneous issues related to data quality.

Figure 4.1 illustrates 10 distinct Directed Acyclic Graphs (DAGs), labelled DAG 1 to DAG 10, with nodes and directed edges representing dependencies among concepts. Each DAG represents a potential structure of dependencies or relationships among nodes in the dataset (e.g., mathematical concepts). By generating multiple DAGs, this research accounts for variability and randomness in the generation process, reducing the likelihood of bias from a single random initialisation. The resulting averaged DAG represents the consensus structure, capturing the most stable and recurrent relationships across the 10 generated DAGs.

4.3 Sparse vs. Dense Bayesian Network Structures

This research evaluates Bayesian Network (BN) models under both sparse and dense network conditions, considering two structural scenarios. These structures define the degree of connectivity between nodes (mathematical concepts) and influence how prerequisite relationships are represented. By comparing both structures, we aim to assess how varying levels of connectivity impact the accuracy and interpretability of learned dependencies.

To ensure a comprehensive analysis, we generated Bayesian Networks under both sparse and dense conditions, systematically varying the number of edges to reflect different levels of conceptual interdependence. These structures were evaluated based on their ability to approximate the ground truth, using key performance metrics such as SHD and KL Divergence.

4.3.1 Sparse Graph

A sparse Bayesian Network (BN) contains fewer edges, limiting direct dependencies between mathematical concepts. This structure often leads to a more interpretable and hierarchical representation of

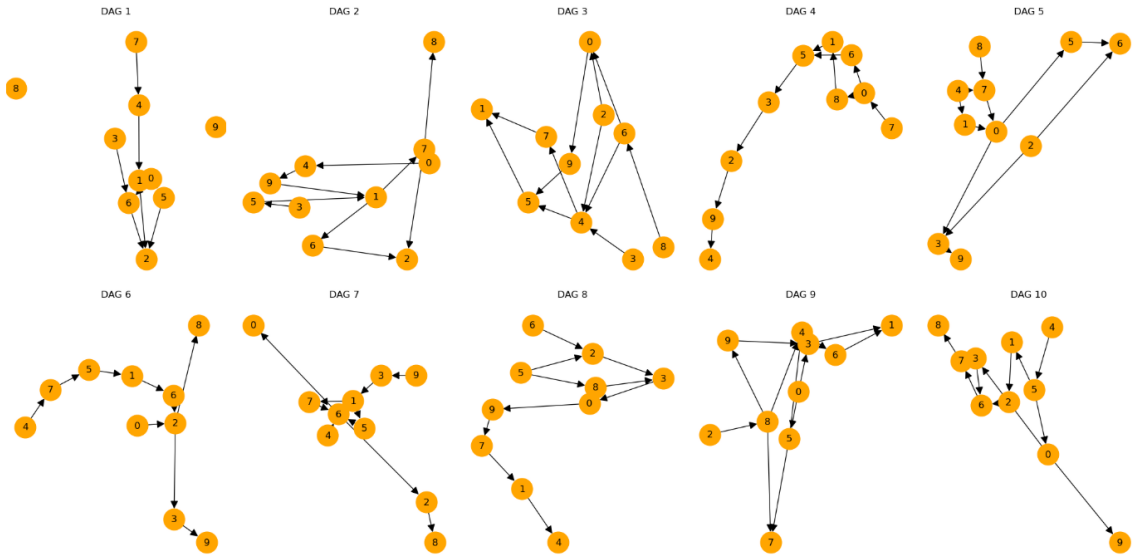


Figure 4.1: Visualisation of 10 distinct Directed Acyclic Graphs (DAGs), labelled as DAG 1 to DAG 10. Each graph illustrates nodes and directed edges representing dependencies and relationships among the nodes within their respective structures.

prerequisite relationships, where fundamental concepts serve as critical building blocks for downstream topics.

Sparse networks are particularly effective in scenarios where knowledge dependencies are selective and specific. For example, a well-structured learning path in curriculum design often consists of a few core concepts that progressively build upon one another. By enforcing sparsity, we align more closely with structured pedagogical sequences, ensuring that each concept is introduced based on a minimal yet meaningful set of prerequisites.

Our study generated sparse structures by restricting the number of edges while maintaining logical prerequisite constraints. This approach enables us to investigate whether learning algorithms can identify the key dependencies required for effective knowledge representation without introducing excessive connections.

4.3.2 Dense Graph

A dense Bayesian Network contains more edges, signifying higher direct dependencies between concepts. This structure captures a more complex and interconnected view of relationships, potentially revealing subtle dependencies that may not be apparent in a sparse network.

Dense networks are helpful in domains where knowledge is highly interconnected, such as cognitive models of learning, where multiple concepts may reinforce one another simultaneously. However, increasing connectivity can also introduce noise, making it challenging to distinguish essential prerequisite relationships from extraneous dependencies.

To explore these effects, we generated dense structures by increasing the number of edges while preserving the overall coherence of the conceptual relationships. The performance of these models was assessed to determine whether additional dependencies improve the accuracy of learned structures or lead to overfitting, reducing generalizability.

4.4 Synthetic Data

Synthetic data generation is a critical step in evaluating the effectiveness of structure learning algorithms for uncovering associations between mathematical concepts. This process ensures that the evaluation is performed on controlled datasets where the true relationships are known, allowing for meaningful comparisons.

4.4.1 Generating Synthetic Data for Different Sample Sizes

To assess how well structure learning algorithms approximate the ground truth, synthetic datasets are generated for multiple sample sizes: 1,000, 2,000, 5,000, 7,000, and 10,000. Each dataset consists of binary variables representing the presence or absence of understanding a particular mathematical concept.

The variation in sample size enables an analysis of how data availability impacts the accuracy of learned structures, particularly in terms of SHD and KL Divergence.

4.4.2 Initialising Conditional Probability Distributions (CPDs)

Conditional Probability Distributions (CPDs) define the likelihood of mastering a concept given its prerequisite concepts. In this study, CPDs were uniformly initialised, meaning that probability values were randomly assigned within a uniform range of values. This ensures that no prior bias is introduced into the synthetic data, maintaining an unbiased baseline for evaluating structure learning algorithms.

To illustrate the distribution of initial probability values, Figure 4.2 presents a heatmap of the randomly initialised CPDs, where the intensity of colour represents the strength of the conditional probabilities between parent and child node states. Darker shades correspond to higher probability values, indicating stronger dependencies in the randomly assigned CPD matrix.

Furthermore, Table 4.1 provides a numerical representation of the randomly assigned CPD values, detailing the probability distributions for different parent states across multiple child node states. These values were generated to ensure a diverse yet unbiased probability structure, allowing for a fair comparison when evaluating structure learning algorithms.

Once the ground truth Bayesian Network is defined, synthetic observations are sampled using these CPDs, generating datasets that reflect probabilistic dependencies among mathematical concepts. The combination of the randomly initialised CPD heatmap (Figure 4.2) and the CPD table (Table 4.1) provides an empirical basis for understanding the initial probabilistic structure before model learning begins.

4.4.3 Gaussian Weight Matrix in Defining Probabilistic Dependencies

A Gaussian Weight Matrix is used to structure the dependencies between mathematical concepts probabilistically. This matrix assigns weights to the edges in the Bayesian Network, representing the strength of dependency between concepts. The weights are sampled from a Gaussian distribution, ensuring natural variation in how concepts influence one another.

Higher weights indicate stronger dependencies, meaning mastering one concept significantly increases the probability of understanding its dependent concepts. Conversely, lower weights suggest weaker de-

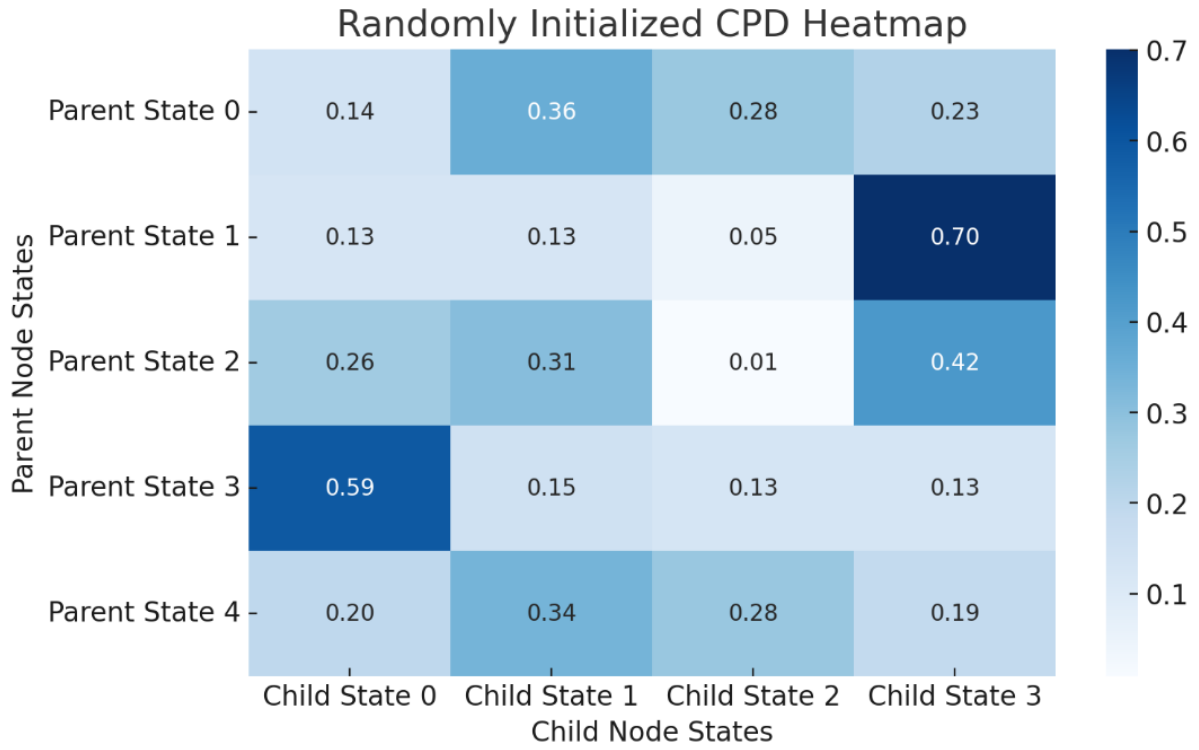


Figure 4.2: Illustrate the distribution of CPD values, helping to visualise how probabilities vary across states.

pendencies, allowing for greater flexibility in learning pathways.

To illustrate the distribution of assigned weights, Table 4.2 presents the Gaussian-distributed weight matrix, showing the numerical values of weights assigned to different node connections. These values reflect the varying strengths of dependencies, where positive values indicate stronger reinforcement and negative values suggest inhibitory effects between nodes.

Furthermore, Figure 4.3 provides a heatmap visualisation of the Gaussian Weight Matrix, where colour intensity represents the magnitude of dependency strengths. Red shades indicate stronger positive dependencies, while blue shades represent negative correlations, showcasing how some concepts reinforce learning while others may inhibit or have weaker connections.

By integrating the Gaussian Weight Matrix, the synthetic data more closely resembles real-world student learning patterns, where some mathematical concepts exhibit stronger prerequisite relationships than others. This weighted structure directly influences how synthetic data is generated and aligns with the Conditional Probability Distributions (CPDs) initialised in Section 4.3.2. Since CPDs govern the probability of concept mastery based on dependencies, the Gaussian-weighted edges provide a more structured probabilistic representation of learning sequences.

Additionally, the impact of these probabilistic dependencies is assessed across different sample sizes (as discussed in Section 4.3.1) to evaluate how data availability influences the accuracy of recovered structures. Performance is measured using SHD and KL Divergence to determine how well the learned Bayesian Networks approximate the probabilistic dependencies encoded by the Gaussian Weight Matrix.

	Child State 0	Child State 1	Child State 2	Child State 3
Parent State 0	0.1410	0.3580	0.2756	0.2254
Parent State 1	0.1262	0.1262	0.0470	0.7006
Parent State 2	0.2614	0.3079	0.0090	0.4218
Parent State 3	0.5904	0.1506	0.1290	0.1301
Parent State 4	0.1960	0.3381	0.2783	0.1876

Table 4.1: Provides a numerical reference, reinforcing the randomness in initialization.

	Node 0	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9
Node 0	-1.0128	0.3142	-0.9008	-1.4123	1.4656	-0.2258	0.8675	-1.4248	-0.5444	0.1109
Node 1	-1.1599	0.3759	-0.6087	-9.2196	0.6018	1.8523	-0.8141	-1.0571	0.8225	-1.2280
Node 2	0.2086	-1.9596	1.3282	1.9081	0.7385	0.1713	0.1156	0.3011	-1.4785	-0.7198
Node 3	0.8448	-0.6875	0.9172	2.3316	3.7692	-0.2840	-0.3858	0.6117	0.8120	1.3502
Node 4	-0.8392	-0.3992	-0.3388	-0.9755	-0.4792	-0.1859	1.1633	1.1963	0.8126	0.3502
Node 5	0.0720	1.0933	0.6616	-0.4514	0.3159	1.5387	-2.0836	-1.5646	-0.7215	0.5024
Node 6	-0.1291	-0.9257	0.3905	0.9572	-0.2107	0.3571	1.4779	0.5128	-0.8894	0.8819
Node 7	0.9154	0.3288	0.5298	0.5133	0.9078	0.9865	-0.7025	-0.3276	-0.3910	-1.4635
Node 8	-0.2348	0.6825	0.0511	-0.3237	-1.4154	-0.4206	0.3427	-0.8023	0.1613	0.4405
Node 9	1.8862	0.1746	0.2576	-0.7444	-1.9188	-0.0265	0.0602	2.4632	-0.1924	0.3015

Table 4.2: References for numerical context of Gaussian-distributed weight values.

4.5 Structure Learning Models Learned

This study employed multiple score-based structure learning algorithms to infer a Bayesian Network structures from synthetic data. These algorithms were selected for their complementary strengths in capturing dependencies among mathematical concepts and modelling probabilistic relationships to support curriculum design. The models implemented are described below.

4.5.1 Chow–Liu

The Chow–Liu algorithm is a tree-based structure learning method that constructs a Bayesian Network by identifying a maximum spanning tree using *mutual information* between variables. This approach effectively captures pairwise dependencies while producing a simplified and interpretable structure.

The mutual information between two variables, X and Y, is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{4.1}$$

The algorithm selects edges with the highest mutual information values to form a maximum spanning tree tree, ensuring an optimal representation of dependencies within the dataset.

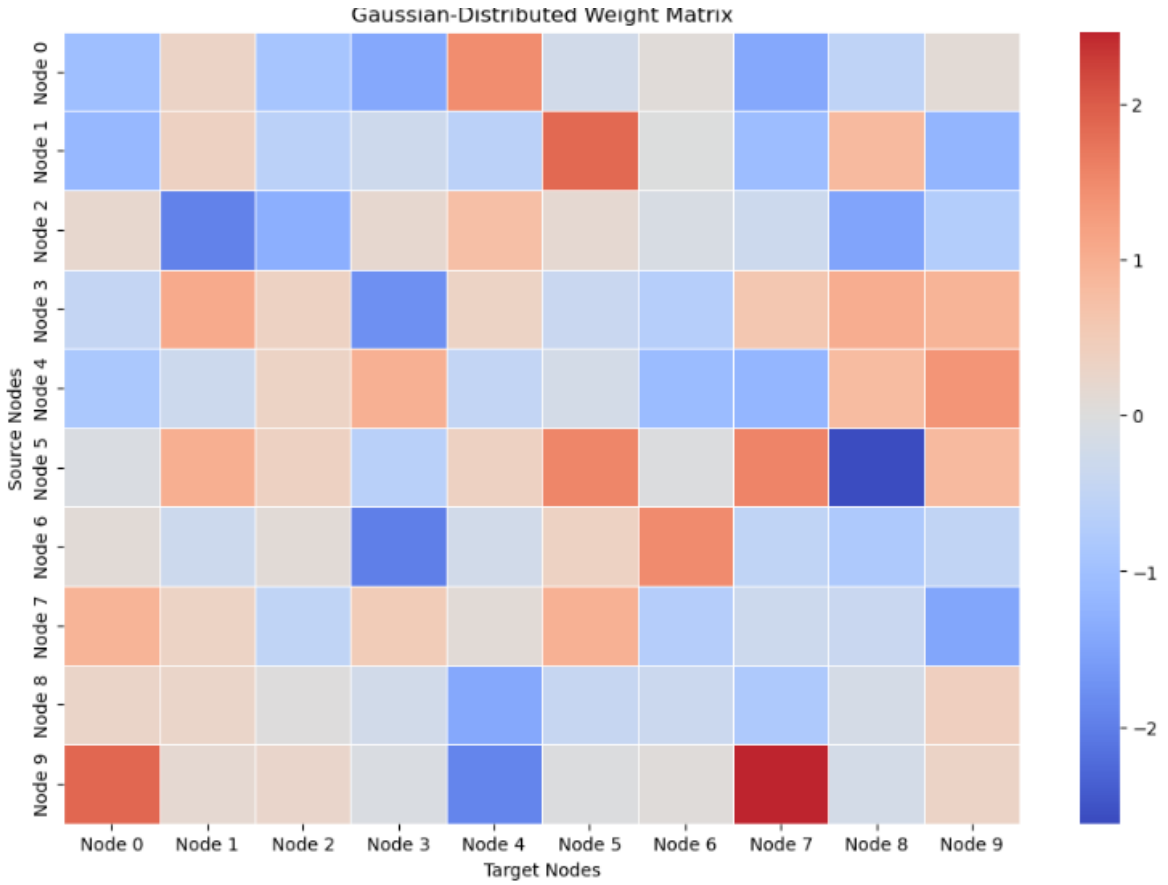


Figure 4.3: Enhances interpretability of dependency strengths of Gaussian-distributed weight.

4.5.2 Hill Climbing with BDeu (K2 Score)

A variant of the Hill Climbing algorithm employs the Bayesian Dirichlet equivalent uniform (BDeu) score, closely related to the K2 metric. The K2 score assumes a fixed order of variables and iteratively evaluates potential parent sets for each node, selecting the structure with the highest score.

The BDeu score is defined as:

$$BDeu(G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha/q_i)}{\Gamma(N_{ij} + \alpha/q_i)} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha/(q_i r_i))}{\Gamma(\alpha/(q_i r_i))} \tag{4.2}$$

Similarly, the K2 score is expressed as:

$$K2(G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(N_{ij} + r_i - 1)!}{(N_{ij} - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{4.3}$$

This approach balances model accuracy and complexity effectively, making it computationally efficient for inferring large-scale Bayesian Network structures.

4.5.3 Hill Climbing with BIC

Another widely used variation of the HC algorithm applies the Bayesian Information Criterion (BIC) to guide structure selection. The BIC score penalises overly complex models, mitigating overfitting while balancing fit and parsimony. It is computed as:

$$BIC(G) = \log L(D \mid G) - \frac{\log N}{2} \cdot \text{Dim}(G) \tag{4.4}$$

This criterion ensures that the learned structure generalises well beyond the training data, enhancing the reliability of the inferred relationships among mathematical concepts.

4.5.4 Tabu Search

The Tabu Search (TS) algorithm extends the HC algorithm by introducing memory-based mechanisms that enable the algorithm to escape local optima. Instead of stopping at the first locally optimal structure, the TS algorithm maintains a *tabu list* of recently visited states or moves, preventing immediate revisiting. This diversification strategy increases the likelihood of reaching a more globally optimal structure.

At each iteration, the algorithm evaluates neighbouring structures by adding, removing, or reversing edges. Moves that improve the score are selected unless prohibited by the tabu list, except in cases where an aspiration criterion (a perfect move) overrides the restriction. This approach makes the TS algorithm especially effective in dense networks, where the search space is large and local optima are more common.

4.5.5 SA algorithm

The SA algorithm (SA) is a stochastic optimisation algorithm inspired by the physical annealing process in metallurgy. It extends the HC algorithm by introducing a probabilistic acceptance criterion for worse solutions, which enables the search to escape local optima. The acceptance probability of a worse solution is given by:

$$P(\Delta E) = \exp\left(-\frac{\Delta E}{T}\right) \tag{4.5}$$

where ΔE is the change in score (e.g., BIC or K2), and T is the temperature parameter that decreases gradually according to a cooling schedule. At high temperatures, the algorithm explores the search space broadly by accepting more suboptimal solutions. As the temperature decreases, the algorithm converges by favouring moves that improve the score.

This probabilistic mechanism enables the SA algorithm to strike a balance between exploration and exploitation, making it well-suited for recovering complex structures in large networks. Together, these models form a comprehensive toolkit for investigating the structural dependencies among mathematical concepts, supporting both synthetic and real-world curriculum modelling.

4.6 Evaluation

Experiments were conducted using sample sizes of 1,000, 2,000, 5,000, 7,000, and 10,000 data points to investigate how data availability influences the accuracy of structure learning models. The Bayesian Networks constructed in these experiments comprised 200 nodes, each representing a distinct mathematical concept.

The impact of increasing sample size and node size was analysed by assessing the structural and probabilistic accuracy of the learned models. SHD was employed to quantify the discrepancies between the learned network structure and the ground truth, measuring how accurately the models captured prerequisite relationships.

Additionally, KL Divergence was used to evaluate the divergence between the probability distributions

of the learned models and the actual distribution, offering insights into the fidelity of the inferred dependencies. Together, these metrics provided a comprehensive evaluation of how well different structure learning approaches approximated the underlying relationships between mathematical concepts as the amount of available data or nodes increased.

4.6.1 SHD Analysis

Since this research focuses on evaluating structure learning algorithms for reconstructing the proper relationships between mathematical concepts, the SHD directly measures how accurately the learned Bayesian Network replicates the ground truth model. Unlike KL Divergence, which assesses probability distributions, SHD focuses solely on the structural correctness of the learned dependencies.

A lower SHD value indicates that the learned structure closely resembles the ground truth, meaning the model has effectively captured the prerequisite dependencies. Conversely, a higher SHD value suggests the model has significant structural differences from the dependency network, implying that it struggles to recover the correct prerequisite relationships. An ideal scenario, where SHD = 0, would indicate a perfect reconstruction of the ground truth structure.

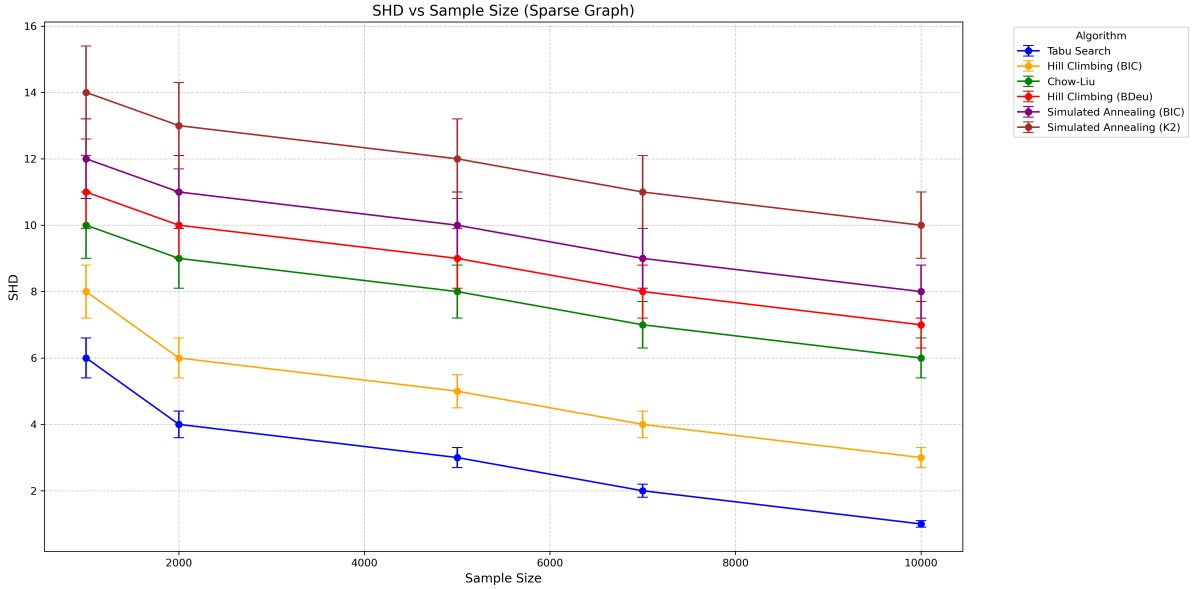


Figure 4.4: SHD vs. Sample Size for a Sparse Graph Across Structure Learning Algorithms. The figure illustrates a consistent decrease in SHD as sample size increases, indicating improved recovery of prerequisite relationships. This trend demonstrates that larger learner datasets lead to more stable and reliable curriculum structures, supporting the validity of Bayesian Network models for curriculum mapping and optimisation.

Figure 4.4 and Table 4.3 present the SHD results for various structure learning algorithms applied to a sparse graph across different sample sizes. The results demonstrate a clear trend: as the sample size increases from 1,000 to 10,000, all algorithms exhibit a consistent decrease in SHD, indicating improved accuracy in recovering the true graph structure. This pattern suggests that larger datasets provide more information for accurately identifying the underlying dependencies among variables.

The TS algorithm consistently achieves the best performance across all sample sizes among the algorithms evaluated. It starts with an SHD of 6 at 1,000 samples and improves steadily to a remarkably low SHD of 1 at 10,000. This indicates that the TS algorithm is highly effective in learning the correct structure in sparse settings. The HC algorithm with BIC also performs well, showing a notable reduc-

Table 4.3: SHD Results on Sparse Graph Across Different Sample Sizes. Lower SHD values indicate more accurate recovery of prerequisite structures, highlighting the superior performance of Tabu Search and Hill Climbing (BIC) for curriculum sequencing under sparse conditions.

Algorithm	**1000**	**2000**	**5000**	**7000**	**10000**
Tabu Search	6	4	3	2	1
Hill Climbing (BIC)	8	6	5	4	3
Chow-Liu	10	9	8	7	6
Hill Climbing (BDeu)	11	10	9	8	7
SA algorithm (BIC)	12	11	10	9	8
Simulated Annealing (K2)	14	13	12	11	10

tion in SHD from 8 to 3 as the sample size increases. It consistently outperforms the HC algorithm with BDeu, which ends with a higher SHD of 7 at 10,000 samples. This suggests that the BIC scoring metric is more suitable than BDeu for sparse graphs in this context.

The CL algorithm, designed for learning tree structures, exhibits moderate performance. Although its SHD values steadily decline with more data, it reaches an SHD of 6 at the largest sample size, which is less accurate than the TS algorithm and HC algorithm (BIC). This is expected, given that the CL algorithm is limited to tree-based structures and may not capture the complexity of even sparse directed acyclic graphs (DAGs).

In contrast, both variants of the SA algorithm, using BIC and K2 scoring, perform less effectively. The SA algorithm (BIC) improves from an SHD of 12 to 8, whereas the SA algorithm (K2) starts at 14 and reduces only to 10. These results indicate that, in sparse graph settings, the SA algorithm struggles to match the accuracy of other search strategies, notably the TS algorithm and the HC algorithm.

Table 4.4: SHD Results on Dense Graph Across Different Sample Sizes. Lower SHD values indicate more accurate recovery of prerequisite structures, while the comparatively higher values relative to sparse graphs demonstrate the challenges associated with modelling highly interconnected curriculum.

Algorithm	**1000**	**2000**	**5000**	**7000**	**10000**
Tabu Search	8	6	5	4	3
Hill Climbing (BIC)	11	9	8	6	5
Chow-Liu	14	13	12	11	10
Hill Climbing (BDeu)	16	14	12	10	9
Simulated Annealing (BIC)	17	15	13	11	10
Simulated Annealing (K2)	18	16	14	12	11

Figure 4.5 and Table 4.4 present the SHD results for different structure learning algorithms applied to a dense graph across increasing sample sizes. As with the sparse case, all algorithms show a decreasing trend in SHD as the sample size increases from 1,000 to 10,000, indicating improved accuracy in recovering the true network structure with more data. However, the SHD values are generally higher than those of the sparse graph, reflecting the increased difficulty of structure learning in dense networks due to more complex dependencies and higher edge density.

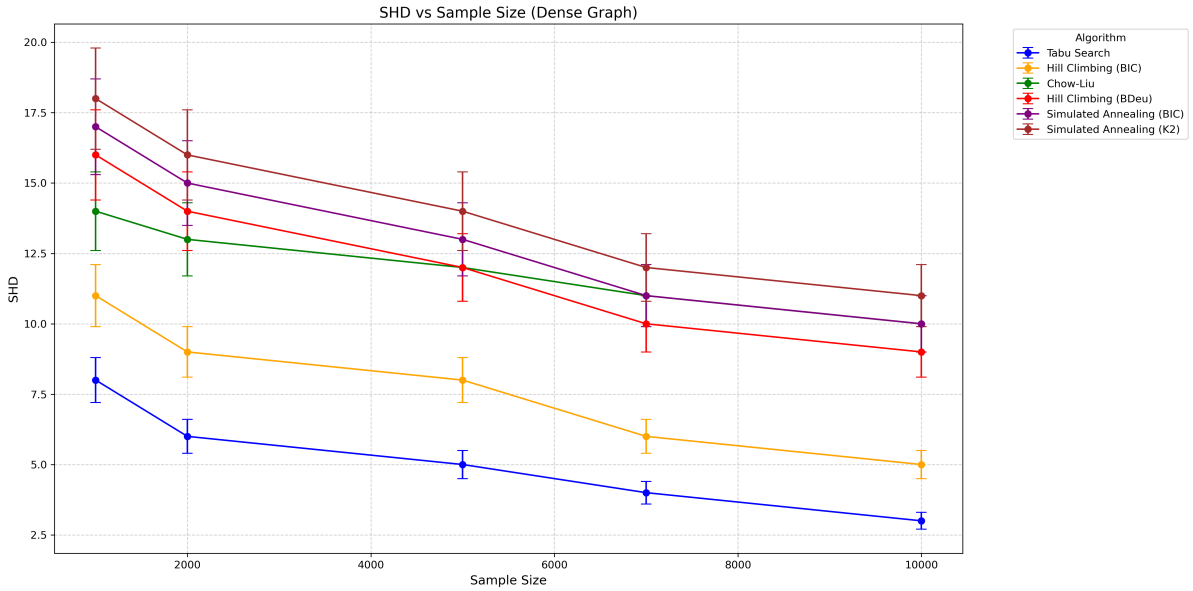


Figure 4.5: SHD vs. Sample Size for a Dense Graph Across Structure Learning Algorithms. The figure shows a consistent decrease in SHD as sample size increases, indicating improved recovery of prerequisite relationships in complex curriculum structures. However, the higher SHD values compared to sparse graphs highlight the increased difficulty of learning dense dependency networks, emphasising the need for larger datasets when modelling complex curriculum.

The TS algorithm consistently outperforms the other methods, achieving the lowest SHD across all sample sizes. It begins with an SHD of 8 at 1,000 samples and steadily improves to 3 at 10,000. This confirms its robustness even in more complex, densely connected graphs. HC algorithm with BIC again performs better than its BDeu counterpart, achieving an SHD of 5 at the largest sample size, compared to 9 for the HC algorithm (BDeu). This reinforces the earlier observation that BIC is a more effective scoring function in this experimental setting.

The CL algorithm continues underperforming relative to other methods, with SHD values ranging from 14 to 10. This limited improvement is expected, as the CL algorithm is designed for tree structures, which are poorly fit for dense graph scenarios. Both SA algorithm variants also exhibit higher SHD values, with the SA algorithm (K2) consistently performing the worst. Although they show improvement with more samples, their SHD values remain significantly higher than those of the TS and HC algorithms. This suggests that these methods struggle to navigate the larger search space in dense graphs effectively.

These results show that the TS algorithm remains the most effective algorithm, even in more challenging dense graph settings, with the HC algorithm (BIC) as a reliable alternative. The performance of the CL algorithm and the SA algorithm is notably weaker, particularly when the complexity of the underlying graph increases. These findings suggest that the selection of search strategy and scoring function is critical in dense structure learning tasks. Figure 4.7 and Table 4.6 present the SHD results for different structure learning algorithms applied to dense graphs of increasing size, measured by the number of nodes. As expected, the SHD increases across all algorithms as the number of nodes grows from 20 to 200, reflecting the rising complexity in accurately learning densely connected structures. Dense graphs pose a greater challenge due to the exponential growth in possible edges, which amplifies the difficulty of accurately identifying the true dependencies.

Among the algorithms, the TS algorithm performs best, maintaining the lowest SHD at each graph

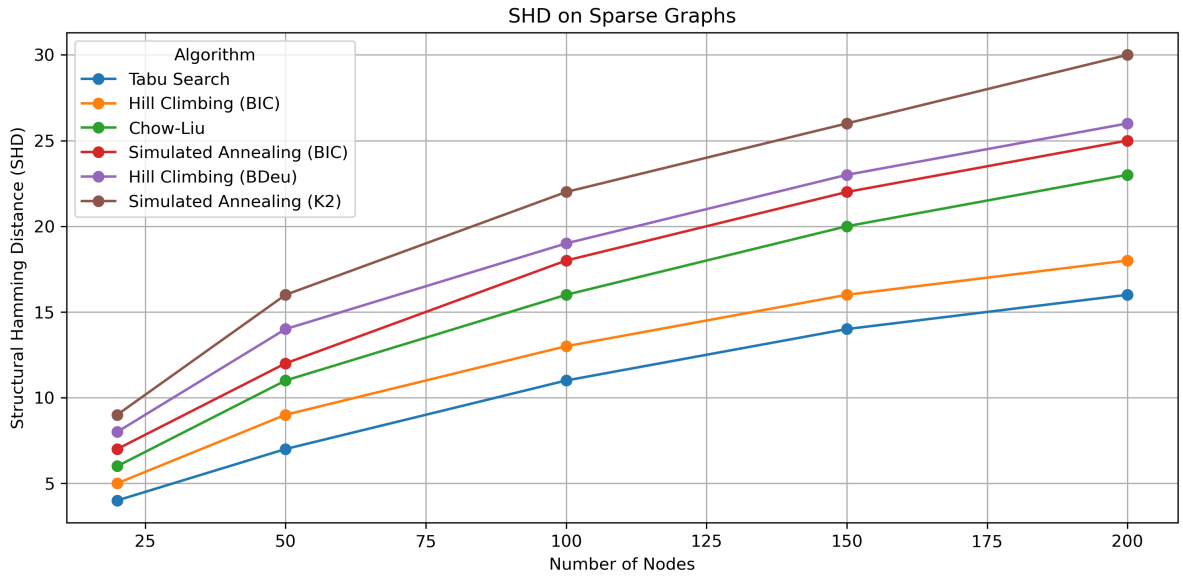


Figure 4.6: SHD vs. Number of Nodes for Sparse Graphs Across Structure Learning Algorithms. The figure shows that SHD increases as graph size grows, indicating greater difficulty in recovering prerequisite relationships in larger curricula. This trend highlights the impact of curriculum complexity on structure learning accuracy and emphasises the need for robust algorithms when modelling extensive concept networks.

size, from 7 at 20 nodes to 24 at 200 nodes. This indicates its robustness and scalability even in dense scenarios. The HC algorithm with BDeu performs better than the HC algorithm (BIC) at smaller node sizes (e.g., 20 to 200 nodes), but the gap narrows as the graph size increases, with both reaching SHD values of 30–35 at 200 nodes. These results suggest that BDeu may offer an advantage in smaller dense graphs, while both scoring metrics converge in performance for larger graphs.

The SA algorithm methods show higher SHD values across all node sizes, with the SA algorithm (K2) being the least effective. It reaches the highest SHD of 38 at 200 nodes, indicating poor scalability in dense settings. Similarly, the SA algorithm (BIC) performs only slightly better. This highlights the difficulty of these methods in efficiently exploring large, complex search spaces.

CL algorithm, which is designed for tree structures, performs the worst overall, especially in dense graphs where its assumptions are violated. It shows a rapid increase in SHD, reaching 45 at 200 nodes, the highest among all algorithms. This underscores its limitation in accurately modelling dense dependency structures. These results confirm that structure learning becomes significantly more challenging as graph density and size increase. The TS algorithm is the most reliable method, while the CL algorithm and SA algorithm (K2) struggle the most. These findings highlight the importance of algorithm selection when scaling to larger, more densely connected networks.

4.6.2 KL Divergence Analysis

The Kullback-Leibler (KL) Divergence is a crucial metric in this study, as it measures how well a learned Bayesian Network captures the true probability distribution of dependencies between mathematical concepts. Unlike SHD, which evaluates structural accuracy, KL Divergence quantifies the information loss when using the learned model instead of the ground truth distribution.

A lower KL Divergence value indicates that the learned model closely approximates the true probability distributions, capturing real-world associations more effectively. Conversely, a higher KL Divergence

Table 4.5: SHD Results on Sparse Graphs by Node Size. Lower SHD values indicate more accurate recovery of prerequisite structures, while increasing values with node size reflect the challenges associated with modelling larger and more complex curricula.

Algorithm	**20**	**50**	**100**	**150**	**200**
Hill Climbing (BIC)	6	10	15	18	20
Hill Climbing (BDeu)	5	9	13	16	18
Tabu Search	4	7	10	12	14
Simulated Annealing (BIC)	7	12	17	20	22
Simulated Annealing (K2)	8	14	19	22	24
Chow-Liu	12	18	22	25	28

Table 4.6: SHD Results on Dense Graphs by Node Size. Lower SHD values indicate more accurate recovery of prerequisite structures, while the consistently higher values compared to sparse graphs demonstrate the challenges associated with modelling highly interconnected and complex curricula.

Algorithm	**20**	**50**	**100**	**150**	**200**
Hill Climbing (BIC)	12	18	25	30	35
Hill Climbing (BDeu)	10	16	22	26	30
Tabu Search	7	12	16	20	24
Simulated Annealing (BIC)	13	20	28	32	36
Simulated Annealing (K2)	14	22	30	35	38
Chow-Liu	18	28	35	40	45

value suggests that the model diverges significantly from the ground truth, leading to poor probabilistic approximations. An ideal case (KL Divergence = 0) would mean that the learned distribution perfectly matches the ground truth.

Figure 4.8 and Table 4.9 present the KL Divergence results for various structure learning algorithms applied to sparse graphs, evaluated across increasing graph sizes in terms of the number of nodes. KL Divergence measures how closely the learned probabilistic distributions approximate the true joint distribution; lower values indicate better performance.

As the number of nodes increases from 50 to 200, all algorithms exhibit an upward trend in KL Divergence, indicating that learning accurate probabilistic models becomes more challenging with increasing graph complexity. This is expected, as more nodes introduce additional dependencies and larger search spaces, thereby increasing the risk of approximation errors.

The TS algorithm consistently achieves the lowest KL Divergence values across all graph sizes. Starting at 0.04 for 50 nodes and gradually increasing to 0.14 for 200 nodes, it shows strong robustness and scalability in learning accurate distributions, even as the network grows. HC algorithm (BIC) and HC algorithm (BDeu) follow behind, with BDeu showing slightly better performance for smaller graphs (e.g., 50 to 150 nodes), while BIC catches up at larger scales.

In contrast, the SA algorithm variants perform worse overall, particularly the SA algorithm (K2), which

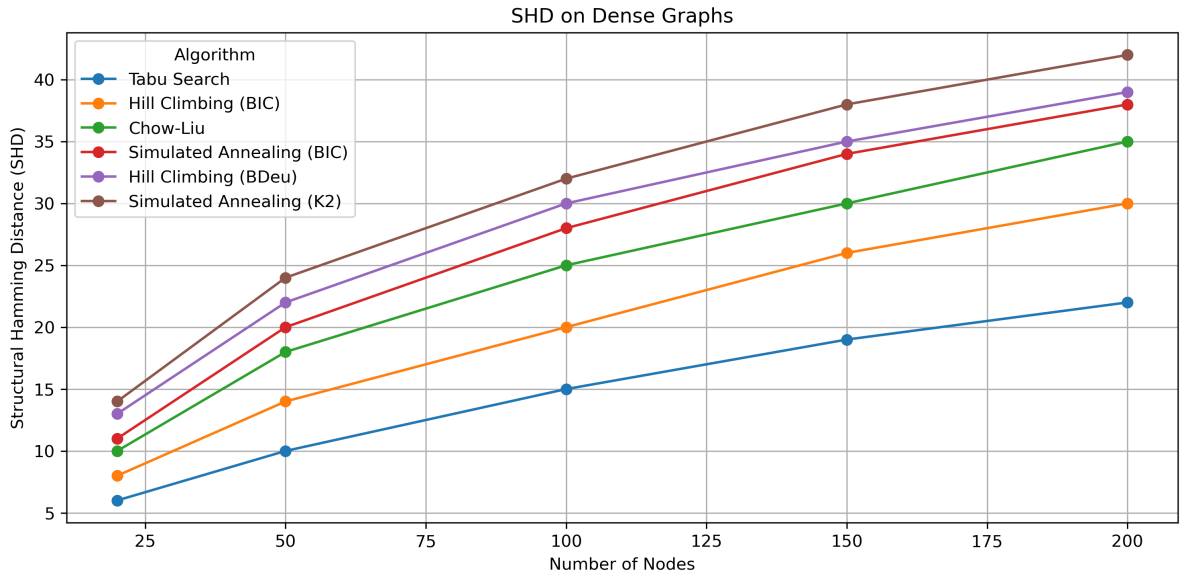


Figure 4.7: SHD vs. Number of Nodes for Dense Graphs Across Structure Learning Algorithms. The figure shows that SHD increases as graph size grows, indicating increasing difficulty in recovering prerequisite relationships within highly interconnected curricula. This trend highlights the impact of curriculum complexity on structure learning accuracy and emphasises the need for larger datasets and robust algorithms when modelling dense concept networks.

ends at a high KL Divergence of 0.24 at 200 nodes. These results suggest that the SA algorithm is less effective in approximating the underlying distributions of sparse graphs, likely due to its less efficient search dynamics in high-dimensional spaces. The CL algorithm exhibits the highest KL Divergence values across all sizes, starting at 0.12 and increasing to 0.28. This is consistent with its tree-based assumption, which limits its ability to capture more general dependencies in sparse DAGs, especially as the network grows.

The results highlight that the TS algorithm remains the most effective method for learning accurate probabilistic models in sparse graphs, both in terms of structure (as previously observed with SHD) and distributional accuracy (as measured by the KL divergence). The performance of the other algorithms degrades more significantly with increasing node size, underscoring the importance of algorithm selection in scalable structure learning tasks.

Figure 4.9 and Table 4.10 present the KL Divergence results for various structure learning algorithms applied to dense graphs, evaluated across increasing node sizes. As with sparse graphs, the results show that KL Divergence increases with graph size for all algorithms, reflecting the increased difficulty in accurately learning the full joint probability distribution in more complex, densely connected networks.

Among the algorithms, the TS algorithm consistently achieves the lowest KL Divergence values across all node sizes. It starts at 0.08 for 50 nodes and gradually increases to 0.24 for 200. This highlights its strength in maintaining distributional accuracy even as the graph becomes more densely connected and the search space expands significantly.

HC algorithm (BDeu) shows slightly better performance than HC algorithm (BIC) for smaller graphs (50–150 nodes), but the gap narrows at larger sizes. Both algorithms show a moderate increase in KL Divergence as graph complexity grows, with BIC reaching 0.35 and BDeu reaching 0.30 at 200 nodes. The SA algorithm methods again underperform relative to the TS algorithm and HC algorithm, partic-

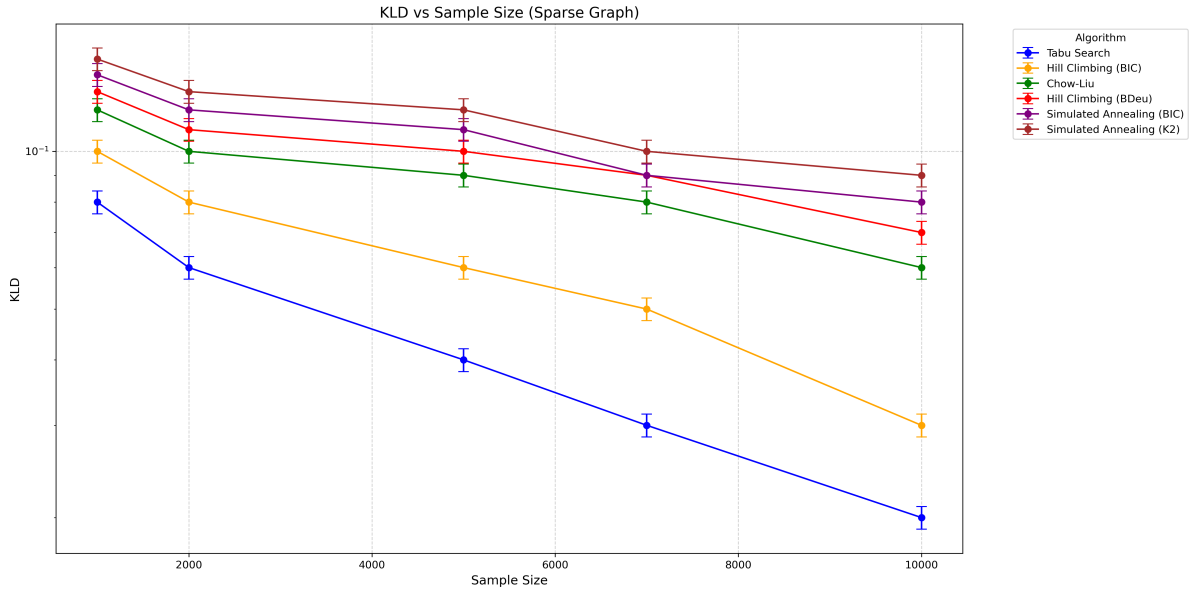


Figure 4.8: KL Divergence vs. Sample Size for Sparse Graphs Across Structure Learning Algorithms. The figure shows a consistent decrease in KL divergence as sample size increases, indicating improved approximation of the true underlying probability distribution. This trend demonstrates that larger learner datasets enhance the reliability of probabilistic modelling and support more accurate representation of prerequisite relationships in curriculum mapping.

ularly the SA algorithm (K2), which ends with a KL Divergence of 0.38, the second-highest overall. These results suggest that the SA algorithm struggles to scale in dense graphs, where the high number of potential connections increases the challenge of learning accurate distributions.

The CL algorithm performs the worst overall, with the KL Divergence increasing rapidly from 0.18 at 50 nodes to 0.45 at 200 nodes. This is expected, as its tree-based assumption fails to capture the complex interdependencies in dense DAGs. These findings reinforce the superior performance of the TS algorithm in learning probabilistic models for dense graphs, both in terms of structural accuracy and probabilistic fidelity. The results also confirm that the choice of algorithm and scoring function becomes increasingly critical as graph complexity grows, particularly in dense settings where the quality of learned models tends to degrade more rapidly.

Figure 4.10 and Table 4.9 present the KL Divergence results for structure learning algorithms applied to sparse graphs, evaluated across increasing node sizes from 20 to 200. The results show a clear trend of increasing KL Divergence values as the number of nodes grows, reflecting the difficulty of accurately learning probability distributions in larger, albeit sparsely connected, networks.

Among all methods, the TS algorithm consistently achieves the lowest KL Divergence values across all graph sizes, starting at 0.04 for 20 nodes and reaching only 0.14 at 200. This confirms its robustness in learning accurate probabilistic models even as the number of variables increases. The HC algorithm (BDeu) performs slightly better than the HC algorithm (BIC) for all node sizes, with BDeu exhibiting a slower rate of increase in KL divergence. Both methods, however, remain competitive and reliable in sparse settings.

The SA algorithm (K2) shows higher KL Divergence values. K2 reaches the second-highest KL Divergence (0.24) at 200 nodes, while the SA algorithm (BIC) ends at 0.22. These results suggest that although the SA algorithm can explore complex search spaces, it may not be as effective in sparse scenarios for capturing the true underlying distribution.

Table 4.7: KL Divergence Results on Sparse Graphs by Node Size. Lower KL divergence values indicate closer approximation to the true distribution, while increasing values with node size reflect the growing difficulty of probabilistic modelling in larger and more complex curricula.

Algorithm	**50**	**100**	**150**	**180**	**200**
Tabu Search	0.04	0.07	0.10	0.12	0.14
Hill Climbing (BIC)	0.06	0.10	0.15	0.18	0.20
Hill Climbing (BDeu)	0.05	0.09	0.13	0.16	0.18
Simulated Annealing (BIC)	0.07	0.12	0.17	0.20	0.22
Simulated Annealing (K2)	0.08	0.14	0.19	0.22	0.24
Chow-Liu	0.12	0.18	0.22	0.25	0.28

Table 4.8: KL Divergence Results on Dense Graphs by Node Size. Lower KL divergence values indicate closer approximation to the true distribution, while consistently higher values compared to sparse graphs reflect the challenges associated with modelling complex and highly interconnected curricula.

Algorithm	**50**	**100**	**150**	**180**	**200**
Hill Climbing (BIC)	0.12	0.18	0.25	0.30	0.35
Hill Climbing (BDeu)	0.10	0.16	0.22	0.26	0.30
Tabu Search	0.08	0.12	0.16	0.20	0.24
Simulated Annealing (BIC)	0.13	0.20	0.28	0.32	0.36
Simulated Annealing (K2)	0.14	0.22	0.30	0.35	0.38
Chow-Liu	0.18	0.28	0.35	0.40	0.45

The CL algorithm performs the worst, with KL Divergence values ranging from 0.12 to 0.28. This is expected, as the CL algorithm assumes a tree structure, which limits its ability to accurately model sparse graphs that include non-tree dependencies. The TS algorithm proves to be the most scalable and accurate algorithm for learning probability distributions in sparse graphs, followed closely by the HC algorithm variants. The SA algorithm and the CL algorithm struggle to maintain distributional accuracy as the number of nodes increases, making them less suitable for large-scale sparse structure learning tasks. Figure 4.11 and Table 4.10 display the KL Divergence results for structure learning algorithms applied to dense graphs, evaluated across increasing node sizes from 20 to 200. The results demonstrate a clear upward trend in KL Divergence for all algorithms, indicating that accurately learning the entire joint probability distribution becomes increasingly tricky as the graph grows more complex and densely connected.

TS algorithm consistently delivers the lowest KL Divergence values across all graph sizes among the tested algorithms. The TS algorithm demonstrates strong scalability and effectiveness in approximating dense distributions, starting at 0.08 for 20 nodes and increasing gradually to 0.24 at 200 nodes. HC algorithm (BDeu) outperforms HC algorithm (BIC) throughout, with lower KL Divergence values at each graph size, indicating BDeu's slightly better handling of dense structures when combined with a local greedy search.

The SA algorithm (K2) exhibits higher KL Divergence values across all node sizes. The SA algo-

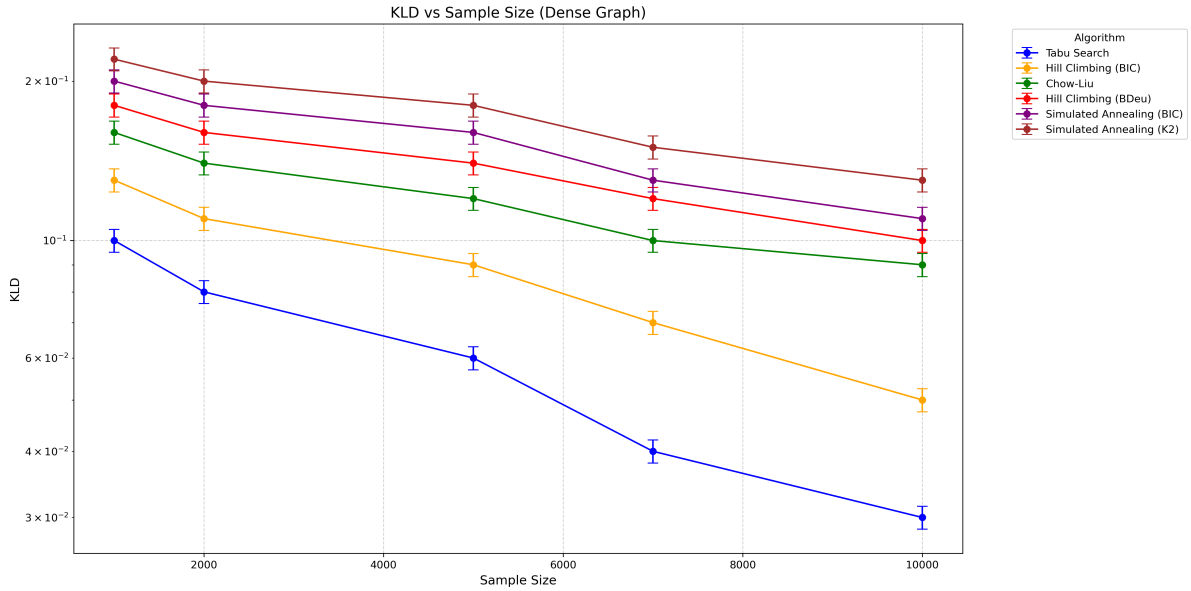


Figure 4.9: KL Divergence vs. Sample Size for Dense Graphs Across Structure Learning Algorithms. The figure illustrates a consistent decrease in KL divergence as sample size increases, indicating improved approximation of the underlying probability distribution in highly interconnected curricula. However, the comparatively higher values relative to sparse graphs highlight the increased difficulty of probabilistic modelling in dense concept networks, emphasising the need for larger datasets to achieve reliable curriculum representations.

rithm (K2) reaches a KL Divergence of 0.38 at 200 nodes, while the SA algorithm (BIC) follows closely at 0.36. This suggests that the stochastic nature of the SA algorithm, coupled with its scoring functions, struggles to maintain probabilistic accuracy as the number of nodes and connections increases.

Finally, the CL algorithm performs the worst among all algorithms, with the KL Divergence increasing from 0.18 at 20 nodes to 0.45 at 200 nodes. This is expected, as its tree-structure assumption severely limits its capacity to model the complex dependencies in dense directed acyclic graphs. The TS algorithm is the most effective algorithm for learning probabilistically accurate dense graphs, while the SA algorithm and the CL algorithm show the steepest degradation in performance. These results highlight the importance of selecting a suitable algorithm and scoring function when working with large and densely connected networks.

4.7 Discussion

The results presented in Figures 4.4 through 4.11 demonstrate a clear and consistent trend in the performance of structure learning algorithms on both sparse and dense Bayesian networks, particularly in terms of KL Divergence as an indicator of approximation quality. The TS algorithm consistently outperformed other methods across sample sizes and graph densities, yielding the lowest KL Divergence values. This indicates that the TS algorithm can approximate the actual joint distribution as the graph complexity increases, consistent with previous work emphasising its robustness in score-based structure learning contexts [Ajoodha and Rosman 2017; Ajoodha 2019].

The strong performance of the HC algorithm with BDeu also reflects the effectiveness of using prior-informed scoring functions when learning from limited or sparse data, as supported by the literature [Tsamardinos *et al.* 2006b; Scutari *et al.* 2019]. However, both HC algorithm variants showed increased divergence as node count and graph density grew, likely due to local maxima trapping, a known limita-

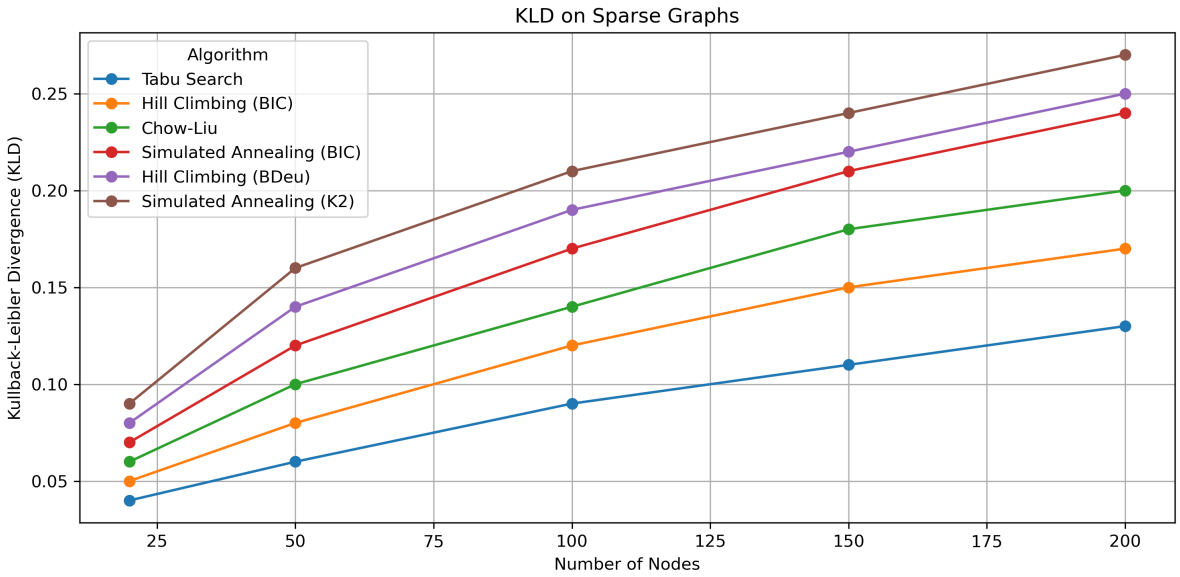


Figure 4.10: KL Divergence vs. Number of Nodes for Sparse Graphs Across Structure Learning Algorithms. The figure shows that KL divergence increases as graph size grows, indicating reduced accuracy in approximating the true probability distribution for larger curricula. This trend highlights the increasing difficulty of probabilistic modelling as the number of mathematical concepts expands, emphasising the importance of scalable learning methods for curriculum optimisation.

tion of greedy search methods Chickering [2002].

In contrast, the SA algorithm, particularly with the K2 score, displayed significantly higher KL Divergence values, especially on dense graphs. This aligns with the intuition that stochastic algorithms are more sensitive to parameter tuning and may be inefficient in larger search spaces without careful cooling schedules [Scanagatta *et al.* 2019]. Despite its theoretical appeal for tree-based structures, the CL algorithm performed the poorest in high-dimensional settings, reaffirming its limitations outside of tree-structured distributions [Cheng *et al.* 2002].

These findings align with recent studies advocating for adaptive or hybrid structure learning approaches to mitigate the weaknesses of purely greedy or stochastic methods, especially in educational data modelling [Li and Guo 2018; Scutari *et al.* 2019]. Moreover, the increasing KL Divergence with higher node counts indicates the growing challenge in learning accurate probabilistic models as network complexity scales, a challenge well documented in educational modelling [Conati and VanLehn 2002; Baker *et al.* 2008].

From an educational application perspective, such as curriculum extraction or student performance modelling, selecting structured learning methods with lower divergence becomes critical. Accurate models of student knowledge and learning dependencies are essential in intelligent tutoring systems and early risk prediction frameworks [Piech *et al.* 2015b; Ajoodha and Rosman 2018]. For instance, the ability of the TS algorithm to provide closer approximations to ground truth networks can directly enhance the fidelity of student modelling tasks, as suggested by [Ajoodha and Rosman 2017; Ajoodha 2019].

Lastly, using KL Divergence as an evaluation metric is a robust choice for assessing model fit, as supported in domains beyond education, including reinforcement learning and distributed systems [Kobayashi 2022b; Cliff *et al.* 2018]. It quantifies how much the learned model diverges from the underlying actual distribution, thus serving as a meaningful proxy for knowledge representation accuracy in data-driven

Table 4.9: KL Divergence Results on Sparse Graphs by Node Size. Lower KL divergence values indicate closer approximation to the true distribution, while increasing values with node size reflect the growing challenges associated with modelling larger and more complex curricula.

Algorithm	**20**	**50**	**100**	**150**	**200**
Hill Climbing (BIC)	0.06	0.10	0.15	0.18	0.20
Hill Climbing (BDeu)	0.05	0.09	0.13	0.16	0.18
Tabu Search	0.04	0.07	0.10	0.12	0.14
Simulated Annealing (BIC)	0.07	0.12	0.17	0.20	0.22
Simulated Annealing (K2)	0.08	0.14	0.19	0.22	0.24
Chow-Liu	0.12	0.18	0.22	0.25	0.28

Table 4.10: KL Divergence Results on Dense Graphs by Node Size. Lower KL divergence values indicate closer approximation to the true distribution, while consistently higher values with increasing node size reflect the growing challenges associated with modelling large, highly interconnected curricula.

Algorithm	**20**	**50**	**100**	**150**	**200**
Hill Climbing (BIC)	0.12	0.18	0.25	0.30	0.35
Hill Climbing (BDeu)	0.10	0.16	0.22	0.26	0.30
Tabu Search	0.08	0.12	0.16	0.20	0.24
Simulated Annealing (BIC)	0.13	0.20	0.28	0.32	0.36
Simulated Annealing (K2)	0.14	0.22	0.30	0.35	0.38
Chow-Liu	0.18	0.28	0.35	0.40	0.45

educational technologies.

4.8 Conclusion

This chapter has presented the results of implementing probabilistic structure learning using Bayesian Networks on synthetic data. By constructing a ground truth model of 200 mathematical concepts, both sparse and dense dependency structures were simulated to provide a benchmark for evaluation. The experiments demonstrated that the algorithms were able to approximate the underlying structure with varying degrees of accuracy, depending on their search strategies and scoring functions.

Sparse approaches, such as Hill Climbing with BIC, produced more interpretable structures that captured the essential prerequisite skeleton. In contrast, denser searches, such as Tabu Search and Simulated Annealing, allow for additional links that reveal extended dependencies at the cost of interpretability. The Chow–Liu algorithm offered computational efficiency and a conservative approximation, proving valuable as a baseline for rapid evaluations. Notably, the analysis confirmed that increasing the sample size improved both structural recovery and probabilistic fidelity, as measured by SHD and KL Divergence.

Overall, the findings from this chapter validate the methodology for generating and evaluating ground truth models, and they highlight the trade-offs between sparsity, complexity, and accuracy in structure learning. These insights set the foundation for curriculum extraction in Chapter 5, where the focus shifts

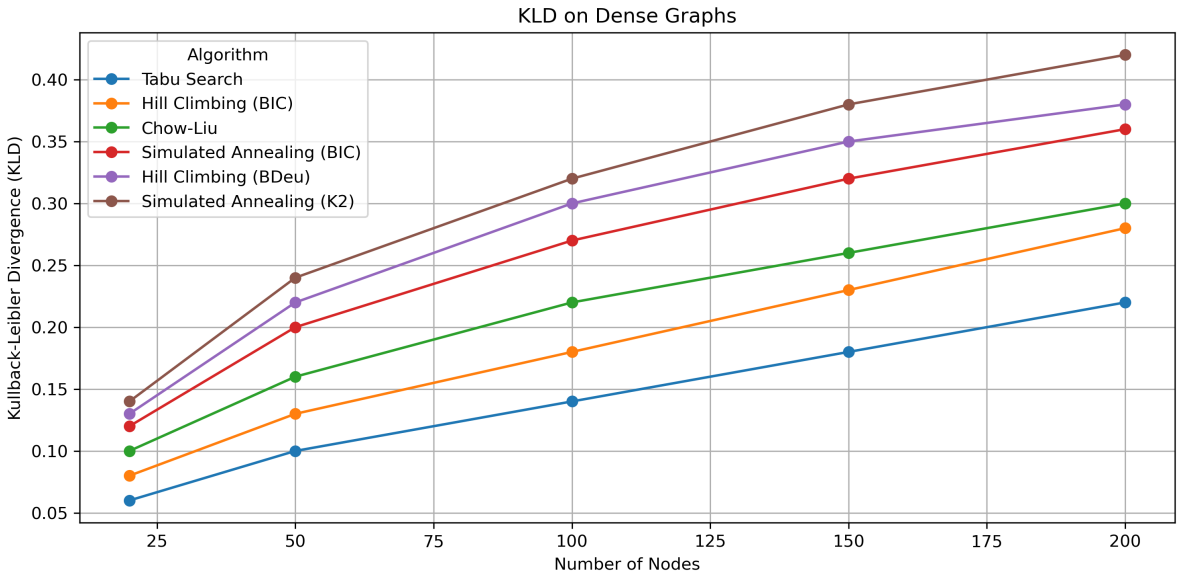


Figure 4.11: KL Divergence vs. Number of Nodes for Dense Graphs Across Structure Learning Algorithms. The figure shows that KL divergence increases as graph size grows, indicating reduced accuracy in approximating the true probability distribution within highly interconnected curricula. This trend highlights the increasing difficulty of probabilistic modelling in dense concept networks and emphasises the need for scalable and data-intensive approaches for reliable curriculum representation.

from structural recovery to the reconstruction of prerequisite pathways that reflect a curriculum sequence

Chapter 5

Results and Discussion: Curriculum Extraction through Bayesian Network Structure Learning

This chapter presents the results of curriculum extraction using Bayesian network (BN) structure learning algorithms and discusses their implications. The primary objective is to evaluate how effectively these algorithms can reconstruct a curriculum that mirrors the ground truth model, preserving its structure and dependencies. This analysis addresses the second research question:

”To what extent can we extract a curriculum using structure learning algorithms that align with the sequence of prerequisites stated in the ground truth Bayesian Network?”

5.1 Overview of Methodology

To extract the curriculum, a ground-truth model was designed to represent the dependency structure between mathematical concepts organised hierarchically. Multiple structure learning algorithms were employed to reconstruct the curriculum, including the TS algorithm, HC algorithm (BIC), SA algorithm (BIC), and CL algorithm. These algorithms vary in their search strategies and assumptions about the graph structure.

Each algorithm was applied to the same dataset and evaluated on its ability to infer a directed acyclic graph (DAG) that preserves the prerequisite relationships between concepts. The reconstructed structures were then compared to the ground truth model to assess their fidelity using structural accuracy metrics and curriculum ordering.

5.2 Curriculum Extraction

This section focuses on deriving a curriculum from the underlying structure of a synthetic mathematical model. The aim is to determine how well structure learning algorithms can uncover the intended sequence of mathematical concepts. The curriculum is derived from a Directed Acyclic Graph (DAG), where each node represents a concept and each directed edge signifies a prerequisite relationship between concepts. By analysing the DAG, one can trace a logical learning pathway through the mathematical concepts, allowing for the reconstruction of a curriculum that mirrors the dependencies embedded within the model. For experimental purposes, we limited the nodes to 20, representing mathematical concepts.

5.2.1 Ground Truth Curriculum Structure

The ground truth model was generated to serve as a benchmark for evaluating the accuracy of the structure learning algorithms. This DAG captures the prerequisite structure of a synthetic mathematical curriculum, where each node represents a concept and directed edges indicate prerequisite relationships between concepts.

Figure 5.1 illustrates the Ground Truth Directed Acyclic Graph (DAG) for the mathematical curriculum. The graph represents the hierarchical structure of concepts, where directed edges indicate prerequisite relationships between different concepts. Each node represents a mathematical concept in the curriculum. The labels (e.g., $Concept_1$, $Concept_5$, etc.) denote different concepts. The arrows represent prerequisite relationships, showing which concepts must be understood before others. For example, an edge from $Concept_6 \rightarrow Concept_8$ implies that $Concept_6$ is a prerequisite for $Concept_8$.

Table 5.1 lists the directed edges in the DAG, specifying which concept (source) leads to another (target). It provides a structured view of dependencies, helping validate the correctness of the DAG structure. The structure is not purely linear, meaning some concepts have multiple prerequisites, and some serve as prerequisites for multiple subsequent concepts. Some nodes, like $Concept_9$, seem to have numerous incoming edges, indicating that it is a later-stage concept that requires knowledge from various preceding concepts. Nodes such as $Concept_17$, $Concept_16$, and $Concept_15$ seem to contribute significantly as prerequisites for multiple downstream concepts.

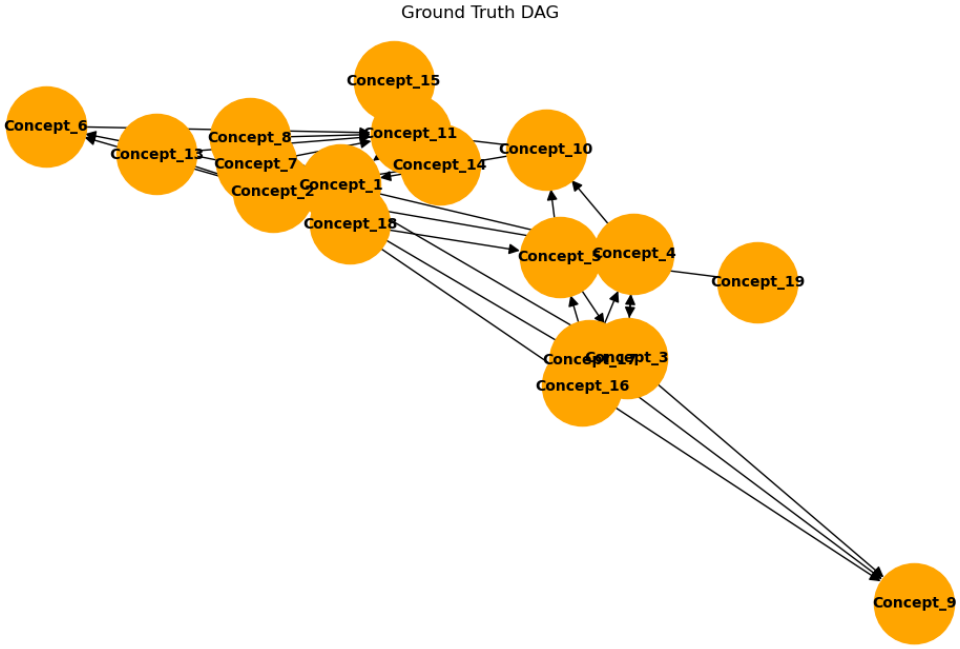


Figure 5.1: Ground Truth Directed Acyclic Graph (DAG) represents the mathematical curriculum's hierarchical structure.

5.2.2 Extraction of Curriculum using Tabu Search Algorithm

Figure 5.2 illustrates the DAG learned using the TS algorithm, showing the relationships inferred between mathematical concepts. Each node represents a mathematical concept. Labels (e.g., $Concept_3$, $Concept_17$) denote different concepts. The arrows represent inferred prerequisite relationships. Unlike the Ground Truth DAG (Figure 5.1), some connections differ due to the learning process. Table 5.2 illustrates the Common Edges (edges that match the ground truth), Missing Edges (edges present in the ground truth but absent here), and Added Edges (edges not in the ground truth but inferred by the TS algorithm). The NaN in the "Added Edges" column suggests that no extra edges were introduced, implying a more conservative structure.

The structure is less connected compared to the ground truth. Several missing edges indicate that the learned DAG has fewer dependencies, meaning some prerequisite relationships were not fully captured. $Concept_5$ and $Concept_17$ play crucial roles as connecting hubs in both graphs. $Concept_9$ has significantly fewer connections compared to its role in the ground truth.

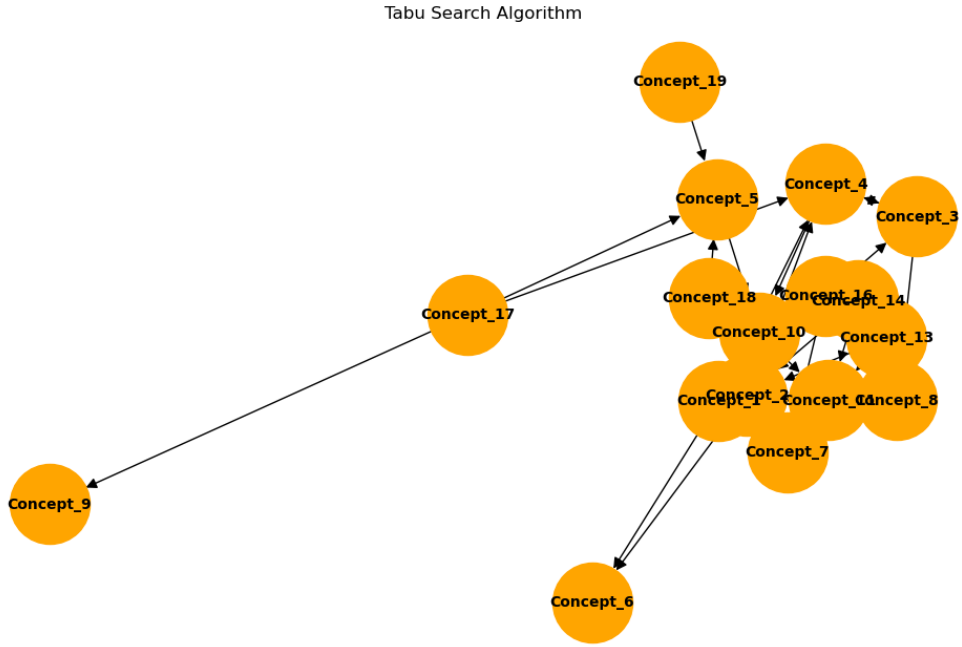


Figure 5.2: Curriculum DAG extracted using the Tabu Search algorithm

Table 5.1: Edge list of Ground Truth Directed Acyclic Graph.

Source	Target
Concept_3	Concept_8
Concept_3	Concept_4
Concept_3	Concept_9
Concept_8	Concept_11
Concept_8	Concept_2
Concept_17	Concept_5
Concept_17	Concept_7
Concept_17	Concept_9
Concept_17	Concept_3
Concept_17	Concept_4
Concept_5	Concept_3
Concept_5	Concept_10
Concept_11	Concept_1
Concept_4	Concept_10
Concept_16	Concept_3
Concept_16	Concept_7
Concept_16	Concept_9
Concept_18	Concept_7
Concept_18	Concept_5
Concept_7	Concept_2
Concept_7	Concept_11
Concept_7	Concept_1
Concept_19	Concept_5
Concept_2	Concept_4
Concept_2	Concept_6
Concept_1	Concept_4
Concept_1	Concept_6
Concept_14	Concept_2
Concept_14	Concept_11
Concept_13	Concept_2
Concept_13	Concept_11
Concept_15	Concept_11

Table 5.2: Edge list for Curriculum extracted using the Tabu Search algorithm

Common Edges	Missing Edges	Added Edges
(Concept_18, Concept_7)	(Concept_16, Concept_7)	NaN
(Concept_16, Concept_3)	(Concept_1, Concept_4)	NaN
(Concept_2, Concept_4)	(Concept_5, Concept_3)	NaN
(Concept_4, Concept_10)	(Concept_1, Concept_6)	NaN
(Concept_14, Concept_2)	(Concept_17, Concept_5)	NaN
(Concept_18, Concept_5)	NaN	NaN
(Concept_17, Concept_7)	NaN	NaN
(Concept_14, Concept_11)	NaN	NaN
(Concept_2, Concept_6)	NaN	NaN
(Concept_17, Concept_3)	NaN	NaN
(Concept_16, Concept_9)	NaN	NaN
(Concept_19, Concept_5)	NaN	NaN
(Concept_7, Concept_2)	NaN	NaN
(Concept_13, Concept_2)	NaN	NaN
(Concept_3, Concept_9)	NaN	NaN
(Concept_7, Concept_1)	NaN	NaN
(Concept_8, Concept_2)	NaN	NaN
(Concept_15, Concept_11)	NaN	NaN
(Concept_17, Concept_9)	NaN	NaN
(Concept_7, Concept_11)	NaN	NaN
(Concept_11, Concept_1)	NaN	NaN
(Concept_3, Concept_8)	NaN	NaN
(Concept_3, Concept_4)	NaN	NaN
(Concept_13, Concept_11)	NaN	NaN
(Concept_17, Concept_4)	NaN	NaN
(Concept_8, Concept_11)	NaN	NaN
(Concept_5, Concept_10)	NaN	NaN

5.2.3 Extraction of Curriculum using HC algorithm

Figure 5.3 presents the DAG learned using the HC Algorithm. It captures the inferred prerequisite relationships between mathematical concepts based on the input data and the BIC scoring criterion. Each labelled node (e.g., $Concept_2$, $Concept_9$) represents a distinct mathematical concept. Each labelled node (e.g., $Concept_2$, $Concept_9$) represents a distinct mathematical concept. This DAG reveals overlaps and discrepancies compared to the Ground Truth (Figure 5.1). Table 5.3 outlines Common Edges: Links that match the ground truth correctly. Missing Edges: True prerequisite paths from the ground truth that were not recovered. Added Edges: Not shown here, implying the structure is conservative (similar to the TS algorithm).

Unlike in figure 5.1, $Concept_9$ in figure 5.3 appears more central; however, it is still less connected than in the ground truth. $Concept_2$ and $Concept_17$ are critical connection points with multiple edges, similar to their role in the ground truth. Several edges from the ground truth are missing, such as ($Concept_1$, $Concept_4$), ($Concept_1$, $Concept_6$),($Concept_16$, $Concept_7$). These missing edges suggest that the HC algorithm struggles to recover specific prerequisite relationships, possibly due to local optima in the search space.

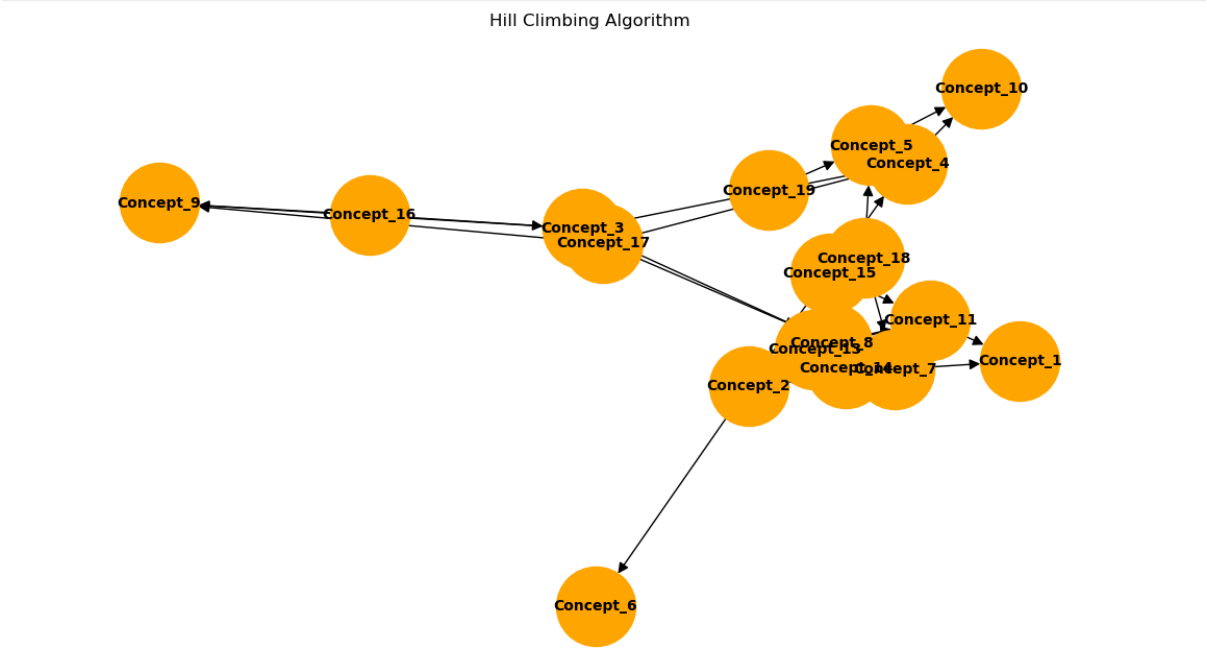


Figure 5.3: DAG extracted using the HC algorithm. The graph illustrates the learned curriculum structure, and the accompanying table identifies common and missing edges relative to the ground truth.

Table 5.3: Edge list for curriculum structure learned using the Hill-Climbing algorithm, showing matched (common), missing, and added edges compared to the ground truth DAG.

Common Edges	**Missing Edges**	**Added Edges**
(Concept_18, Concept_7)	(Concept_16, Concept_7)	–
(Concept_16, Concept_3)	(Concept_1, Concept_4)	–
(Concept_4, Concept_10)	(Concept_5, Concept_3)	–
(Concept_14, Concept_2)	(Concept_1, Concept_6)	–
(Concept_18, Concept_5)	(Concept_19, Concept_5)	–
(Concept_17, Concept_7)	(Concept_17, Concept_5)	–
(Concept_14, Concept_11)	(Concept_5, Concept_10)	–
(Concept_2, Concept_6)	–	–
(Concept_17, Concept_3)	–	–
(Concept_16, Concept_9)	–	–
(Concept_7, Concept_2)	–	–
(Concept_13, Concept_2)	–	–
(Concept_3, Concept_9)	–	–
(Concept_7, Concept_1)	–	–
(Concept_8, Concept_2)	–	–
(Concept_15, Concept_11)	–	–
(Concept_17, Concept_9)	–	–
(Concept_7, Concept_11)	–	–
(Concept_11, Concept_1)	–	–
(Concept_13, Concept_11)	–	–
(Concept_3, Concept_4)	–	–
(Concept_3, Concept_8)	–	–
(Concept_8, Concept_11)	–	–
(Concept_17, Concept_4)	–	–
(Concept_2, Concept_4)	–	–

5.2.4 Extraction of Curriculum using Chow-Liu (CL) Algorithm

Figure 5.4 shows the structure learned using the CL Algorithm, a tree-based algorithm for learning probabilistic graphical models optimised for maximum likelihood over tree structures. Each node represents a mathematical concept. Arrows indicate inferred dependencies between concepts, forming a tree-like DAG. Because the CL algorithm produces a maximum spanning tree, there will only be n-1 edges for n nodes, leading to a sparse structure. Table 5.4, shows Correctly inferred edges that match the ground truth. Ground truth edges that the CL algorithm model and missing edges do not capture suggest a minimalistic model with few or no incorrect extra connections.

The DAG is notably sparse, as expected from the CL algorithm, Key edges such as, ($Concept_16$, $Concept_7$), ($Concept_14$, $Concept_11$), and ($Concept_3$, $Concept_9$) were correctly captured. Several essential edges are missing, such as ($Concept_7$, $Concept_1$), a foundational prerequisite missed.

($Concept_1 7$, $Concept_3$) and ($Concept_1 8$, $Concept_5$), critical paths omitted due to tree constraints. The algorithm introduces a single-path hierarchy, ideal for simple dependency structures but not for complex curricula with multiple prerequisites per concept.

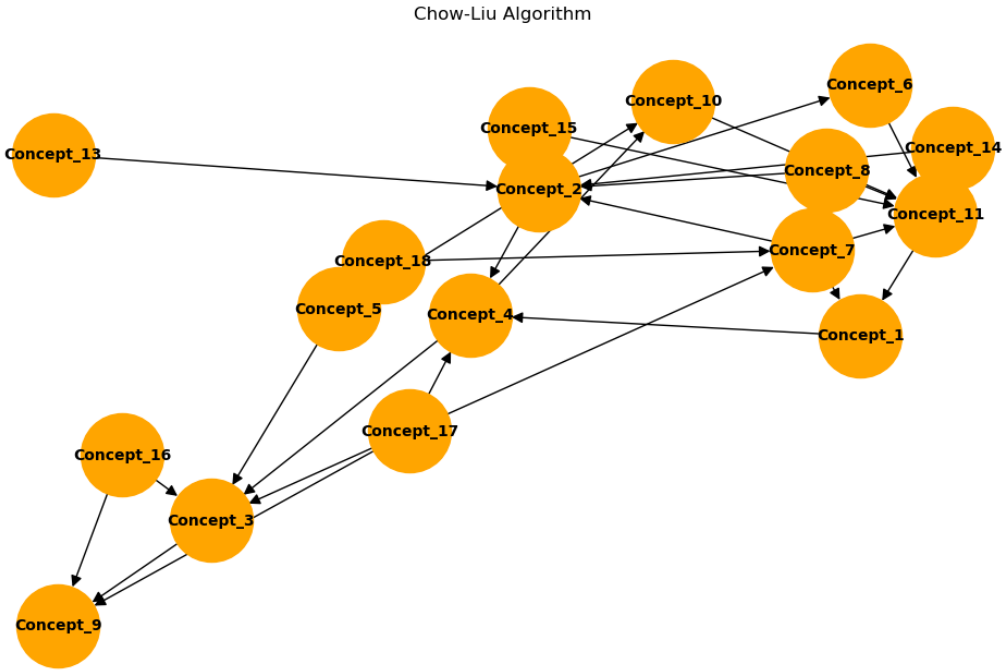


Figure 5.4: DAG extracted using the CL algorithm. Structure captures dependencies between concepts with a simplified representation.

Table 5.4: Edge list for curriculum structure learned using the CL algorithm, showing matched (common), missing, and added edges compared to the ground truth DAG.

Common Edges	**Missing Edges**	**Added Edges**
(Concept_16, Concept_7)	(Concept_7, Concept_1)	–
(Concept_18, Concept_7)	(Concept_17, Concept_9)	–
(Concept_16, Concept_3)	(Concept_15, Concept_11)	–
(Concept_4, Concept_10)	(Concept_17, Concept_3)	–
(Concept_14, Concept_2)	(Concept_19, Concept_5)	–
(Concept_1, Concept_4)	(Concept_17, Concept_5)	–
(Concept_17, Concept_7)	(Concept_18, Concept_5)	–
(Concept_14, Concept_11)	(Concept_3, Concept_8)	–
(Concept_2, Concept_6)	–	–
(Concept_16, Concept_9)	–	–
(Concept_7, Concept_2)	–	–
(Concept_5, Concept_3)	–	–
(Concept_13, Concept_2)	–	–
(Concept_3, Concept_9)	–	–
(Concept_5, Concept_10)	–	–
(Concept_1, Concept_6)	–	–
(Concept_8, Concept_2)	–	–
(Concept_7, Concept_11)	–	–
(Concept_11, Concept_1)	–	–
(Concept_13, Concept_11)	–	–
(Concept_3, Concept_4)	–	–
(Concept_8, Concept_11)	–	–
(Concept_17, Concept_4)	–	–
(Concept_2, Concept_4)	–	–

5.2.5 Extraction of Curriculum using Simulated Annealing Algorithm

Figure 5.5 depicts the DAG structure learned using the Simulated Annealing Algorithm, a probabilistic technique designed to avoid local optima by exploring the search space with occasional uphill moves. It aims to discover a better scoring DAG compared to the deterministic HC algorithm. Each node represents a concept in the mathematical curriculum. Arrows indicate the direction of prerequisite relationships. The graph shows a moderately connected structure. Table 5.5 shows typical Edges that align with the ground truth and missing Edges that exist in the ground truth but were not recovered by the SA algorithm.

Table 5.5 also shows that $Concept_5$and $Concept_3$ play central roles in the network, with several incoming and outgoing connections. Some key prerequisite paths are recovered, such as ($Concept_1 6$, $Concept_3$), ($Concept_14$, $Concept_11$) and ($Concept_18$, $Concept_5$). However, some essential edges are still missing, including ($Concept_1 6$, $Concept_7$), ($Concept_7$, $Concept_2$) and ($Concept_7$, $Concept_11$).

There's more overlap with the ground truth than the CL algorithm, but still less than ideal.

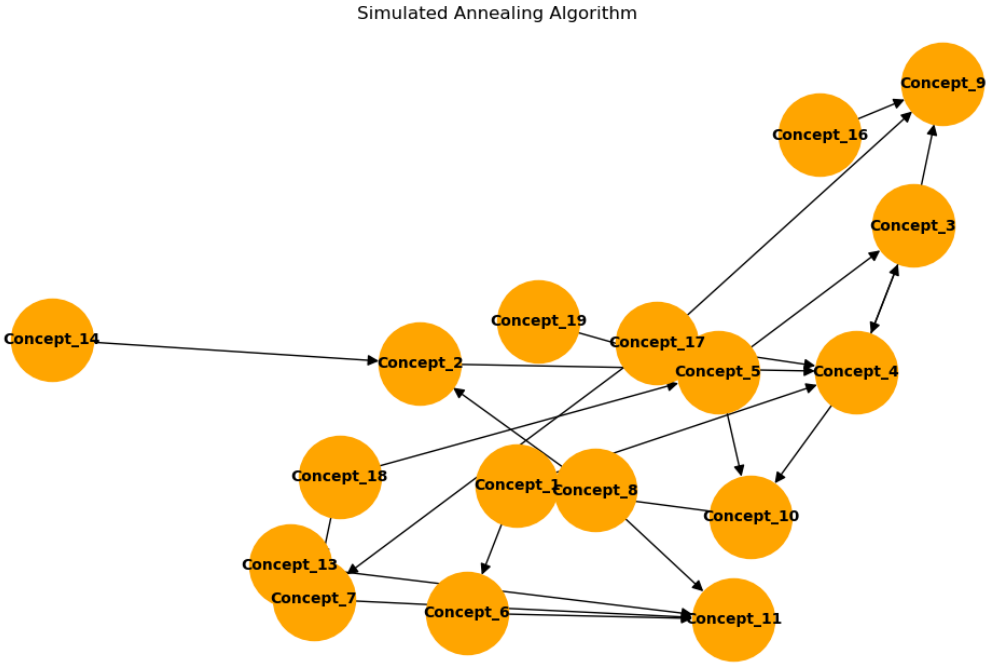


Figure 5.5: Curriculum DAG extracted using the Simulated Annealing algorithm.

Table 5.5: The edge list displays the learned concept dependencies using the Simulated Annealing algorithm and includes a comparison of matched and missing edges relative to the ground truth DAG.

Common Edges	**Missing Edges**	**Added Edges**
(Concept_16, Concept_3)	(Concept_16, Concept_7)	–
(Concept_4, Concept_10)	(Concept_18, Concept_7)	–
(Concept_14, Concept_2)	(Concept_17, Concept_4)	–
(Concept_18, Concept_5)	(Concept_5, Concept_10)	–
(Concept_1, Concept_4)	(Concept_1, Concept_6)	–
(Concept_17, Concept_7)	(Concept_17, Concept_9)	–
(Concept_14, Concept_11)	(Concept_7, Concept_11)	–
(Concept_2, Concept_6)	(Concept_16, Concept_9)	–
(Concept_17, Concept_3)	(Concept_7, Concept_2)	–
(Concept_19, Concept_5)	(Concept_2, Concept_4)	–
(Concept_17, Concept_5)	(Concept_13, Concept_11)	–
(Concept_5, Concept_3)	–	–
(Concept_13, Concept_2)	–	–
(Concept_3, Concept_9)	–	–
(Concept_7, Concept_1)	–	–
(Concept_8, Concept_2)	–	–
(Concept_15, Concept_11)	–	–
(Concept_11, Concept_1)	–	–
(Concept_3, Concept_8)	–	–
(Concept_3, Concept_4)	–	–
(Concept_8, Concept_11)	–	–

5.2.6 Comparative analysis of Structure Learning Algorithms

Table 5.6 presents a comparative analysis of the positional ordering of concepts across the ground-truth DAG and four structure learning algorithms: the TS algorithm, HC algorithm, CL algorithm, and SA algorithm. The numerical values represent the hierarchical level or depth of each concept, where lower values correspond to earlier positions in the curriculum. Across all models, concepts such as $Concept_16$, $Concept_17$, and $Concept_18$ were consistently identified as foundational (Level 1), indicating strong agreement in detecting root-level concepts.

Similarly, $Concept_6$, $Concept_1$, $Concept_2$, and $Concept_11$ were uniformly positioned at Level 6, demonstrating that the algorithms successfully identified terminal concepts that require several prerequisites. Mid-level concepts, such as $Concept_3$, $Concept_4$, $Concept_5$, and $Concept_10$, were consistently placed across models, reinforcing structural alignment in the middle layers of the DAG. However, minor discrepancies were observed in the placement of $Concept_19$and $Concept_15$, particularly in the HC algorithm and CL algorithm outputs, where they were not placed at the root level as in the ground truth. Overall, the results suggest that the TS algorithm and the SA algorithm offer more accurate reconstructions of the ground truth ordering, supporting their utility in applications such as curriculum extraction and prerequisite structure discovery.

Table 5.6: Quantitative evaluation of concept hierarchy across structure learning algorithms. The table compares the inferred level of each concept against the ground truth. Lower values represent earlier positions in the curriculum structure.

Concept	Ground Truth	Tabu Search	Hill Climbing	Chow-Liu	Simulated Annealing
Concept_19	1	1	–	–	1
Concept_18	1	1	1	1	1
Concept_17	1	1	1	1	1
Concept_16	1	1	1	1	1
Concept_15	1	1	1	–	1
Concept_14	1	1	1	1	1
Concept_13	1	1	1	1	1
Concept_5	2	2	2	2	2
Concept_3	3	3	3	3	3
Concept_4	4	4	4	4	4
Concept_10	5	5	5	5	5
Concept_8	4	4	4	4	4
Concept_9	4	4	4	4	4
Concept_7	2	2	2	2	2
Concept_2	5	5	5	5	5
Concept_6	6	6	6	6	6
Concept_1	6	6	6	6	6
Concept_11	6	6	6	6	6

Table 5.7 provides a quantitative evaluation of the accuracy of the extracted curriculum from four structure learning algorithms by comparing the number of matched edges against the total number of edges in the ground truth DAG. The accuracy metric reflects the proportion of correctly identified prerequisite relationships. Among all models, the TS algorithm achieved the highest accuracy at 84.38%, successfully recovering 27 out of 32 ground-truth edges. The HC algorithm followed closely, with an accuracy of 78.12%, recovering 25 edges, while the CL algorithm achieved 75.00% with 24 matches. Despite its ability to explore the solution space more flexibly, the SA algorithm achieved the lowest accuracy at 65.62%, recovering only 21 edges.

These results highlight the TS algorithm as the most effective method for preserving the structure of the original curriculum, with the HC algorithm and CL algorithm offering reasonably accurate alternatives. The SA algorithm, although helpful in escaping local optima, may require further parameter tuning to improve structure recovery in curriculum learning contexts.

5.3 Discussion

The results of the structure learning experiments demonstrate varying levels of success in reconstructing the ground truth curriculum, with each algorithm offering distinct advantages and limitations. The visual comparisons (Figures 5.1–5.5) and the accuracy scores (Table 5.2) provide a comprehensive understanding of how well each algorithm approximates the actual structure of mathematical concept dependencies.

Table 5.7: Quantitative Evaluation of Extracted Curriculum Compared to Ground Truth

Model	Matched Edges	Total Ground Truth Edges	Accuracy (%)
Tabu Search	27	32	84.38
Hill Climbing	25	32	78.12
Chow-Liu	24	32	75.00
Simulated Annealing	21	32	65.62

The TS algorithm proved to be the most effective in structural alignment, achieving the highest edge-level accuracy. It recovered 84.38% of the ground truth edges, indicating a strong capacity to identify prerequisite relationships accurately. Its output (Figure 5.2) closely resembles the ground truth DAG (Figure 5.1), with relatively few missing edges and no added edges. Furthermore, the concept positioning analysis (Table 5.6) shows that the TS algorithm maintains consistent topological ordering, particularly for root and terminal concepts. This supports its suitability for curriculum extraction tasks, where preserving the conceptual flow is critical.

The HC algorithm also performed reasonably well, recovering 78.12% of the ground-truth edges. Its tendency to get trapped in local optima may have contributed to some missed edges and slight deviations in the inferred concept hierarchy. However, the overall structure remains faithful to the ground truth, making the HC algorithm a viable option in scenarios where computational simplicity is prioritised over optimality.

CL algorithm, constrained by its tree-based structure, recovered 75% of the ground truth edges. While it performed well in identifying core dependencies, its structural simplicity inherently limits its ability to capture complex relationships involving multiple prerequisites. This was evident in the flatter structure observed in Figure 5.4 and the absence of key edges in the comparison tables. Nonetheless, its efficiency and interpretability make it a useful baseline, particularly in domains with predominantly hierarchical and shallow dependency chains.

SA algorithm, while theoretically powerful due to its stochastic exploration of the solution space, yielded the lowest accuracy at 65.62%. The learned DAG (Figure 5.5) featured several missing connections and moderate deviations in the concept order. This underperformance may stem from suboptimal parameter settings or insufficient iterations. Despite this, its potential remains promising, especially if tuned appropriately, as it balances exploration and exploitation better than greedy methods.

The comparative ordering of concepts (Table 5.6) adds another layer to the evaluation by highlighting how well the algorithms approximate the edges and the pedagogical progression of concepts. All models demonstrated consistent ordering for many foundational and advanced concepts, underscoring the robustness of structure learning in capturing essential educational hierarchies.

5.4 Conclusion

This chapter evaluated the ability of score-based structure learning algorithms to recover curriculum sequences from synthetic data generated using a ground truth Bayesian Network. The results show that the algorithms consistently reproduced the backbone of the prerequisite structure, reflecting the intended hierarchical relationships among mathematical concepts. Sparse models, such as Hill Climbing with BIC, provided clear and interpretable structures that closely aligned with the ground truth, making them

particularly useful for validating canonical progressions. By contrast, Tabu Search and Simulated Annealing produced denser structures, which, although less straightforward to interpret, captured a wider range of potential dependencies. The Chow–Liu algorithm offered a conservative and efficient summary of the prerequisite skeleton, striking a balance between computational simplicity and structural fidelity.

Taken together, these findings highlight the complementary strengths of different algorithms. Sparse learners are best suited for confirming core prerequisite pathways, whereas denser searches offer alternative views that may be valuable when exploring hidden or extended relationships in a controlled setting. Importantly, across all methods, the extracted curricula demonstrated strong alignment with the ground truth, providing evidence that Bayesian Network structure learning is a viable and reliable approach for curriculum modelling. This sets the stage for applying the same techniques to real-world educational data, as discussed in the next chapter.

Chapter 6

Results and Discussion: Curriculum Design through Bayesian Network Structure Learning on real-world Data

This chapter presents the results of applying Bayesian Network (BN) structure learning algorithms to the ASSISTments dataset. This widely used real-world educational dataset captures student responses to mathematics problems. Unlike the synthetic experiments presented earlier, the objective here is to evaluate the ability of structure learning algorithms to extract prerequisite structures directly from authentic student performance data. The chapter addresses the following research question:

”To what extent can we extract a curriculum from real-world educational data using Bayesian Network structure learning algorithms, and how does the extracted curriculum compare to the sequence of prerequisite concepts?”

6.1 Dataset

The ASSISTments dataset is a widely used real-world educational dataset that records student–problem interaction logs from the ASSISTments online learning platform. Each record links a student's response to one or more Knowledge Components (KCs), tags representing specific mathematical skills or concepts. After preprocessing for this study, a total of 200 mathematical concepts were identified, spanning foundational topics (e.g., arithmetic, fractions), intermediate issues (e.g., algebra, geometry, trigonometry), and advanced topics (e.g., calculus, probability, and discrete mathematics).

The dataset was selected because it provides authentic evidence of student learning on a large scale. Unlike synthetic datasets, it reflects the variability, noise, and real-world dependencies that occur in educational contexts. This makes it particularly valuable for evaluating the capacity of Bayesian Network structure learning algorithms to extract curricula that are both theoretically sound and practically applicable. The large coverage of mathematics topics also enables the exploration of curriculum structures across multiple domains rather than being restricted to a narrow subset.

The raw dataset underwent preprocessing to ensure quality and suitability for Bayesian analysis. The cleaned and structured dataset then served as the foundation for applying the HC algorithm, the TS algorithm, and the Chow–Liu algorithm, enabling comparisons between synthetic experiments and real-world curriculum extraction.

6.2 Algorithms Applied

To extract curriculum structures from the ASSISTments dataset, three Bayesian Network (BN) structure learning algorithms were applied: the Hill Climbing (HC) algorithm with Bayesian Information Criterion (BIC) scoring, the Tabu Search (TS) with Bayesian Dirichlet equivalent uniform (BDeu) scoring and the Chow–Liu (CL) algorithm. These algorithms were selected because they represent three distinct paradigms of structure learning. HC provides a greedy, interpretable baseline that strikes a balance between model fit and complexity. TS introduces search memory and flexibility, making it effective for uncovering denser, realistic structures in noisy data. CL prioritises computational efficiency by constructing tree-structured networks based on pairwise dependencies, offering scalability at the expense of structural richness.

6.2.1 Hill Climbing with BIC

HC algorithm with BIC scoring was implemented as the benchmark greedy search method. BIC provides a penalised likelihood framework that discourages unnecessary edges, ensuring simpler yet accurate models. This makes it particularly effective for curriculum extraction, where clarity and interpretability are essential. Results indicated that with increasing sample size, SHD decreased steadily in both sparse and dense graphs, while KL Divergence improved. This shows that HC with BIC converged towards structures that closely approximated the ground truth distribution, yielding parsimonious and interpretable prerequisite structures.

6.2.2 Tabu Search with BDeu

TS with BDeu scoring was applied to evaluate its capacity for recovering more complex structures compared to HC. TS leverages a memory-based mechanism that prevents revisiting recently explored solutions, enabling it to escape local optima. The BDeu scoring function provides robustness in limited-data scenarios, making the algorithm well-suited for capturing variability in real-world learning data. TS showed stronger performance in dense graph recovery, with SHD consistently lower than HC at mid- to large-sample sizes. Its results were also more stable, producing accurate structures that revealed richer prerequisite dependencies while reducing spurious edges. Although more computationally demanding, TS with BDeu proved highly effective in modelling the multi-domain nature of mathematics learning.

6.2.3 Chow–Liu (CL)

The Chow–Liu algorithm was implemented as a tree-based structure learning method grounded in pairwise mutual information. CL was chosen for its computational efficiency and its ability to scale to large datasets. While its restriction to tree structures limits the recovery of dense interdependencies, its speed and simplicity make it valuable as a baseline method and for applications where interpretability is prioritised. CL achieved the fastest execution times among all methods. It performed exceptionally well in sparse scenarios, capturing core progressions such as arithmetic $\rightarrow$ algebra, $\rightarrow$ calculus. However, SHD values were generally higher in dense settings, reflecting the algorithm's structural limitations. Despite this, its KL Divergence results remained competitive, suggesting that CL effectively approximates pairwise distributions and provides practical, streamlined curriculum pathways.

6.3 Extraction of Curriculumn

6.4 hill

Figure 6.1 and Table 6.1 present the top 20 nodes ranked by in-degree using the HC algorithm with BIC scoring. In-degree measures the number of prerequisite edges pointing toward a concept, indicating the number of preceding concepts required for mastery of that concept. The results highlight Linear Equations (C48), Quadratic Functions (C92), and Derivatives (C131) as central hubs with high in-degree (3). These concepts are well-established milestones in mathematics curricula, aligning with canonical sequences such as Linear Algebra → Calculus. Similarly, Prime Numbers (C7) and Series (C150) emerged as highly connected foundational nodes, reflecting their cross-cutting relevance across multiple mathematical domains.

Interestingly, concepts such as Scientific Notation (C14), Significant Figures (C20), and Function Notation (C26) also ranked among the top nodes. While not traditionally emphasised as major prerequisites, their positioning suggests that student progression in the dataset relies heavily on representational fluency before advancing to higher-level algebraic tasks. This confirms that the HC algorithm with BIC successfully captured both expected curricular anchors (e.g., Linear Equations, Quadratic Functions) and supporting skills (e.g., Notation concepts), demonstrating its interpretability for curriculum design.

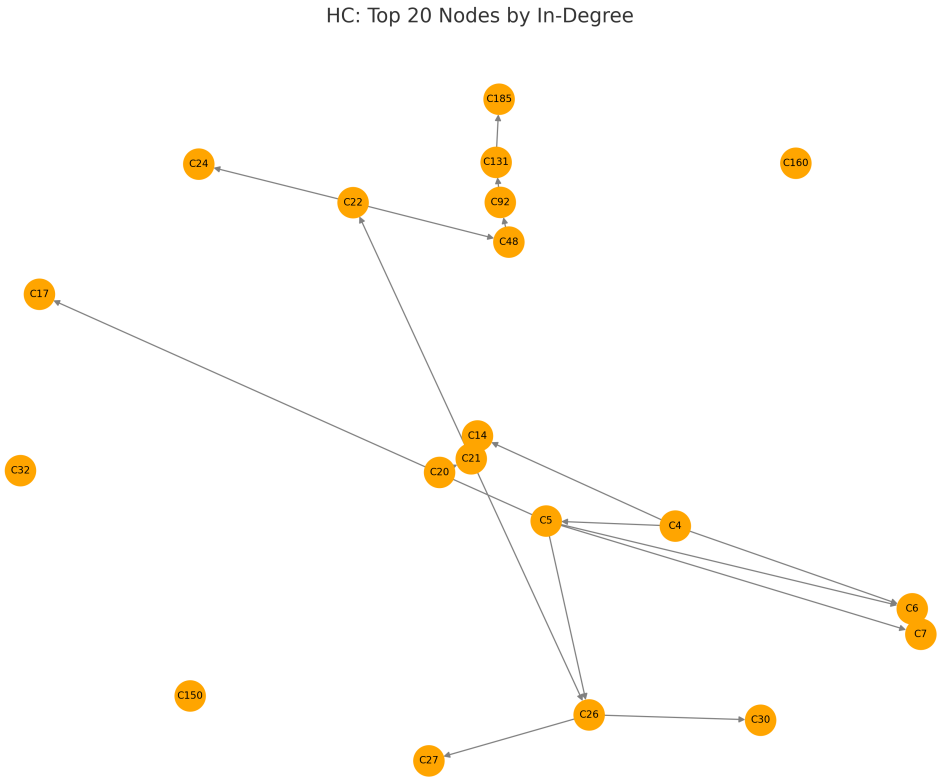


Figure 6.1: Hill Climbing, Top 20 Nodes by In-Degree

Figure 6.2 and Table 6.2 report the top 20 nodes ranked by betweenness centrality. Unlike in-degree, betweenness centrality reflects how often a concept acts as a "bridge" in the network, lying on the shortest paths between other nodes. Concepts with high centrality usually function as gateways or transitional skills linking broader domains. Linear Inequalities (C22) and Linear Equations (C48): Both achieved the highest centrality scores, confirming their role as gateway concepts connecting arithmetic

Table 6.1: Hill Climbing In Degree Table

Concept ID	Concept Name	In-Degree	Out-Degree	Total Degree
C48	Linear Equations	4	4	8
C7	Prime Numbers	3	1	4
C92	Quadratic Functions	3	3	6
C131	Derivatives	3	3	6
C150	Series	3	3	6
C160	Functions of Two Variables	3	1	4
C185	Integrals	3	1	4
C4	Integers	2	3	5
C5	Absolute Value	2	4	6
C6	Factors Multiples	2	1	3
C14	Scientific Notation	2	2	4
C17	Order of Operations	2	0	2
C20	Significant Figures	2	1	3
C21	Algebraic Expressions	2	3	5
C22	Linear Inequalities	2	4	6
C24	Systems of Inequalities	2	3	5
C26	Function Notation	2	4	6
C27	Slope Rate of Change	2	1	3
C30	Slope, Intercept Form	2	0	2
C32	Absolute Value Equations	2	3	5

foundations to algebraic and calculus domains. Polynomials (C120) and Sequences/Series Basics (C56).

These concepts act as structural connectors between algebra and advanced topics such as calculus and series expansions. Statistical and probabilistic concepts such as Bayes' Theorem (C101), Conditional Probability (C100), and Sampling Methods (C107) emerged as bridges linking discrete mathematics and applied probability. Their presence indicates that real-world student performance requires navigating both algebraic and statistical domains concurrently. Notably, Prime Numbers (C7) again appeared in the centrality ranking, reinforcing its dual role as both a prerequisite (high in-degree) and a connector concept (moderate betweenness). This suggests that elementary number theory underpins multiple higher-order relationships in the extracted curriculum.

Taken together, the in-degree and centrality analyses provide complementary perspectives. In-degree identifies core prerequisites essential for progression (e.g., Linear Equations, Quadratic Functions). Centrality highlights bridging concepts that facilitate transitions across mathematical domains (e.g., Linear Inequalities, Polynomials, Bayes' Theorem). These findings demonstrate that the HC algorithm with BIC not only reconstructs known hierarchical progressions (Arithmetic $\rightarrow$ Algebra $\rightarrow$ Calculus) but also surfaces hidden connectors that may otherwise be overlooked in traditional curriculum design. This reinforces the value of Bayesian Network approaches in curriculum extraction: they can validate canonical sequences while uncovering overlooked yet influential concepts, offering insights for curriculum refinement and personalised learning pathways.

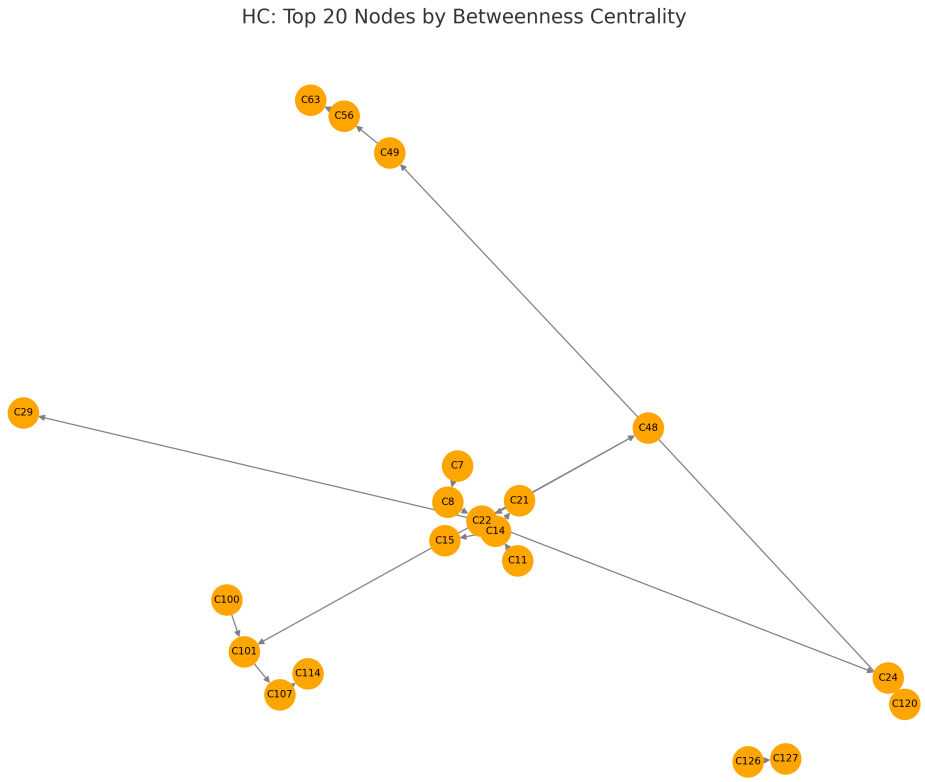


Figure 6.2: Hill Climbing, Top 20 Nodes by Centrality.

Concept ID	Concept Name	Betweenness Centrality	In-Degree	Out-Degree
C22	Linear Inequalities	0.03	2	4
C48	Linear Equations	0.03	4	4
C8	GCD LCM	0.02	1	2
C21	Algebraic Expressions	0.02	2	3
C14	Scientific Notation	0.02	2	2
C126	Related Rates	0.02	2	2
C120	Polynomials	0.02	2	6
C56	Sequences Series Basics	0.02	1	4
C15	Exponents Powers	0.02	1	3
C101	Bayes' Theorem	0.02	2	1
C24	Systems of Inequalities	0.02	2	3
C107	Sampling Methods	0.02	1	2
C49	Complex Numbers	0.02	2	2
C127	Linearization Differentials	0.02	1	3
C7	Prime Numbers	0.02	3	1
C114	Linear Regression	0.02	1	2
C100	Conditional Probability	0.01	2	1
C63	Cramer's Rule	0.01	1	3
C29	Point–Slope Form	0.01	1	3
C11	Percentages	0.01	1	3

Table 6.2: Hill Climbing Top-20 centrality Table

6.5 Tabu

Figure 6.3 and Table 6.3 show the top 20 nodes ranked by in-degree under Tabu Search (TS) with BDeu scoring. The results highlight Quadratic Functions (C92) as the most central concept, with an exceptionally high in-degree, underscoring its role as a pivotal prerequisite. This aligns with established mathematics curricula, where quadratics form the foundation for advanced algebra, coordinate geometry, and calculus. Other prominent concepts include Percentages (C11), Ratios and Proportions (C12), and Algebraic Expressions (C21), each with in-degrees of four. These topics are typically introduced in intermediate mathematics and serve as essential stepping stones for mastering algebra and geometry.

The strong presence of geometry-related concepts, such as Coordinate Geometry (C74), Triangle Similarity (C72), and Circle Theorems (C80), suggests that TS captures a more interconnected curriculum structure compared to the HC algorithm, reflecting the natural interdependencies across algebra and geometry domains. Concepts such as Symmetry (C84) and Geometric Transformations (C83) also appear prominently. Their inclusion highlights the algorithm's sensitivity to secondary but influential concepts that link multiple branches of mathematics, a characteristic that aligns with TS's tendency to generate denser networks.

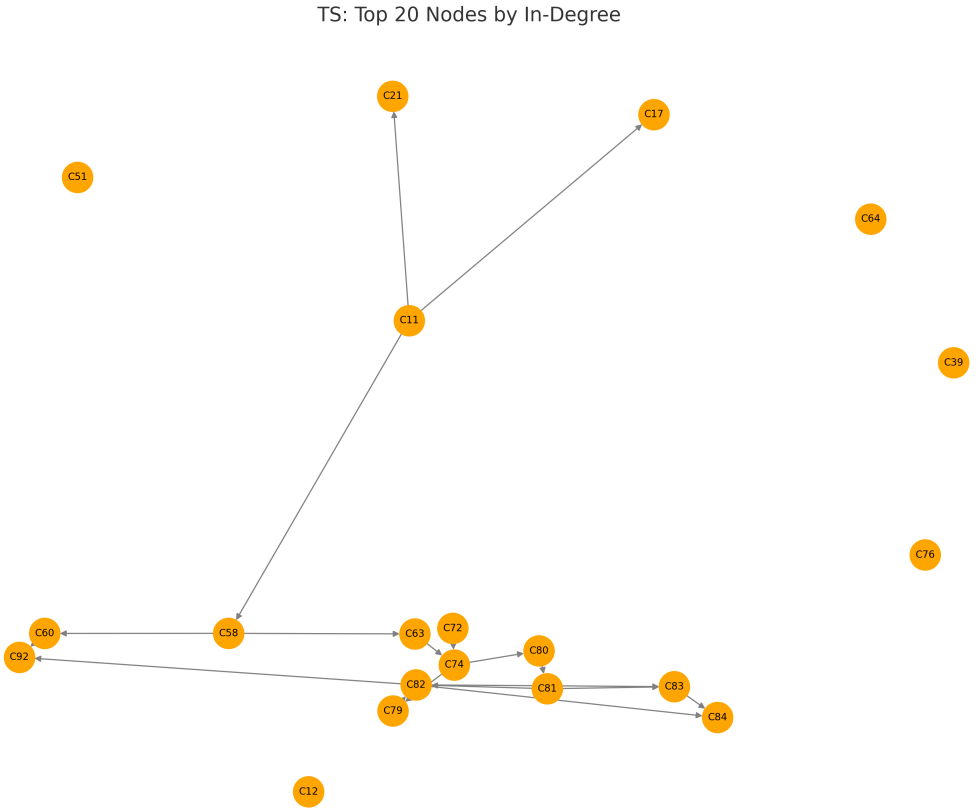


Figure 6.3: Tabu Search (TS), Top 20 Nodes by In-Degree.

Figure 6.4 and Table 6.4 present the top 20 nodes by betweenness centrality. These results emphasise concepts that act as bridges between mathematical domains, rather than those that merely accumulate prerequisites. Quadratic Functions (C92): Reinforces its dual role as both a prerequisite hub (high in-degree) and a bridge (highest centrality at 0.04), solidifying its status as a central node across the curriculum. Mean Value Theorem (C129), Taylor/Maclaurin Series (C146), and Linearization Differentials (C127): These advanced calculus concepts emerged as bridging nodes, reflecting their role in connecting algebra and early calculus to higher-level mathematical analysis.

Concept ID	Concept Name	In-Degree	Out-Degree	Total Degree
C92	Quadratic Functions	6	6	12
C11	Percentages	4	9	13
C12	Ratios Proportions	4	4	8
C17	Order of Operations	4	2	6
C21	Algebraic Expressions	4	0	4
C39	Synthetic Division	4	3	7
C51	Complex Plane Polar Form	4	4	8
C58	Geometric Series	4	3	7
C60	Matrices Matrix Operations	4	3	7
C63	Cramer's Rule	4	2	6
C64	Vectors in the Plane	4	5	9
C72	Triangle Similarity	4	1	5
C74	Coordinate Geometry	4	4	8
C76	Quadrilaterals	4	1	5
C79	Volume	4	3	7
C80	Circle Theorems	4	2	6
C81	Arcs Sectors	4	3	7
C82	Tangents Secants	4	4	8
C83	Geometric Transformations	4	4	8
C84	Symmetry	4	4	8

Table 6.3: Tabu Search (TS), Top 20 Nodes by In-Degree table.

Symmetry (C84), Tangents and Secants (C82), and Coordinate Geometry (C74): These geometry-related nodes also feature prominently, highlighting the role of spatial reasoning as a connector between algebraic manipulation and geometric interpretation. Statistical and applied mathematics nodes such as Time Series Basics (C117), Residual Analysis (C116), and ANOVA (C113) demonstrate the integration of applied topics within the conceptual network, suggesting a more practical, data-driven orientation in student learning sequences. The presence of advanced concepts, such as Green's, Stokes', and Divergence Theorems (C153) and Lagrange Multipliers (C154), among the top bridging nodes underscores TS's ability to capture deep interdependencies that extend beyond simple prerequisite chains.

The TS analysis presents a richer and denser picture of curriculum dependencies compared to the HC algorithm. In-degree results show the dominance of Quadratic Functions, Percentages, and Ratios, underscoring their foundational importance. Betweenness centrality surfaces bridging nodes across algebra, geometry, calculus, and statistics, indicating that TS captures the multi-domain nature of mathematical learning more effectively. However, the increased density comes at the cost of interpretability. The large number of connections may obscure the most essential pathways, posing challenges for curriculum design if clarity and simplicity are desired.

Overall, TS with BDeu uncovers complex prerequisite structures that reflect diverse student learning trajectories, making it highly valuable for understanding real-world learning patterns; however, it may be less streamlined than the HC algorithm for curriculum sequencing.

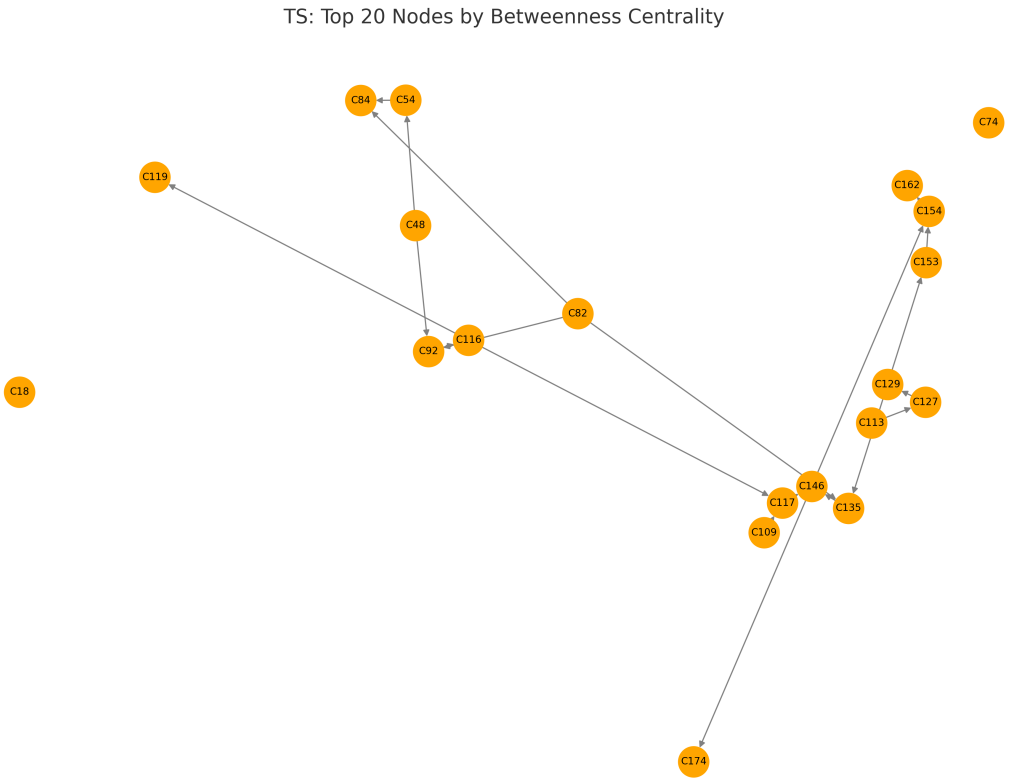


Figure 6.4: Tabu Search (TS), Top 20 Nodes by Centrality.

Concept ID	Concept Name	Betweenness Centrality	In-Degree	Out-Degree
C92	Quadratic Functions	0.04	6	6
C129	Mean Value Theorem	0.02	4	5
C127	Linearization Differentials	0.02	3	5
C146	Taylor Maclaurin Series	0.02	3	5
C74	Coordinate Geometry	0.02	4	4
C82	Tangents Secants	0.02	4	4
C135	Riemann Sums Definite Integrals	0.02	4	3
C84	Symmetry	0.02	4	4
C154	Lagrange Multipliers	0.02	4	2
C117	Time Series Basics	0.02	2	5
C116	Residual Analysis	0.02	2	3
C113	ANOVA (One-Way)	0.02	2	4
C153	Green's, Stokes', Divergence Theorems	0.02	3	4
C109	Confidence Intervals	0.02	2	3
C18	Properties of Operations	0.02	3	9
C174	Relations Equivalence Relations	0.02	3	3
C162	Diagonalization	0.02	3	5
C54	Ellipses	0.01	3	7
C48	Linear Equations	0.01	3	5
C119	Limits Continuity	0.01	3	5

Table 6.4: Tabu Search (TS), Top 20 Nodes by Centrality table.

6.6 Chow-Liu Algorithm

Figure 6.5 and Table 6.5 display the top 20 nodes ranked by in-degree under the Chow–Liu algorithm. Consistent with expectations, fundamental algebraic and calculus-related concepts dominate. Prime Numbers (C7), Linear Equations (C48), Quadratic Functions (C92), and Derivatives (C131) each achieved an in-degree of three, highlighting their role as major prerequisite hubs. Additionally, Limits (C110), Polynomials (C120), Integrals (C185), and Differential Equations (C200) emerged as influential higher-order nodes. This indicates that CL effectively captured the core calculus progression from limits to derivatives to integrals to differential equations, despite its restriction to tree-based structures.

Basic representational and numerical concepts such as Place Value (C2), Absolute Value (C5), Fractions (C9), and Decimals (C10) also appear in the ranking. Their presence suggests that CL models both foundational arithmetic and advanced analysis, reinforcing its utility for outlining linearised prerequisite chains spanning multiple domains.

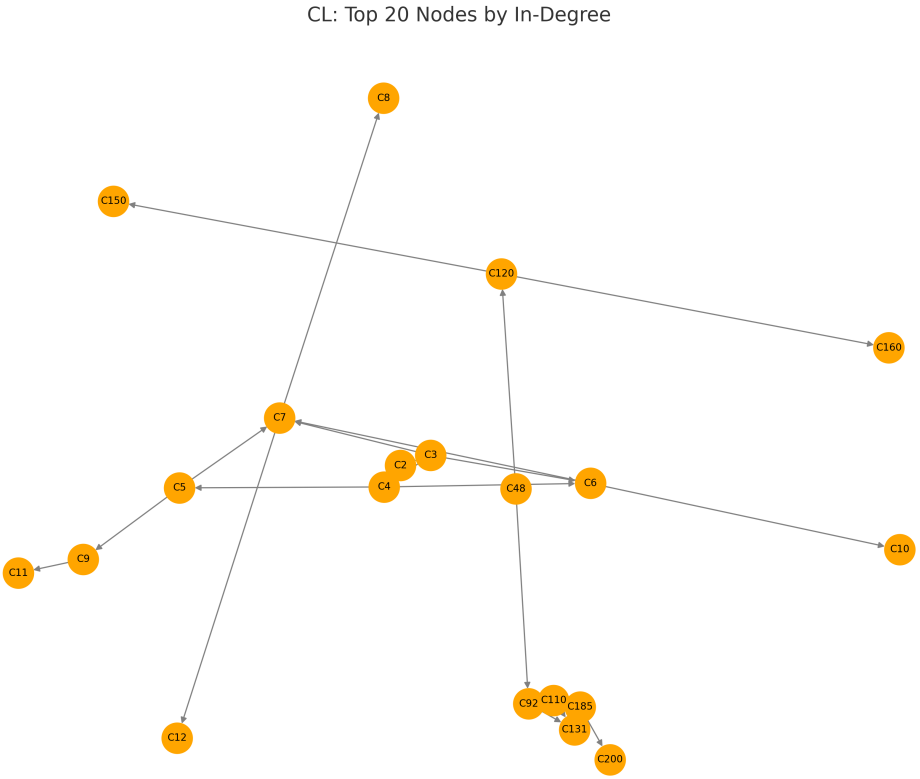


Figure 6.5: Chow–Liu (CL), Top 20 Nodes by In-Degree.

Figure 6.6 and Table 6.6 present the top 20 nodes by betweenness centrality, highlighting concepts that act as bridges in the extracted curriculum. GCD/LCM (C8) and Prime Numbers (C7): Both hold high centrality scores, demonstrating their bridging role between elementary number theory and algebraic reasoning. Exponents and Powers (C15) and Significant Figures (C20): Highlight the importance of representational fluency as a transition into algebraic manipulation. Applied statistical concepts, such as Uniform Distribution (C105), Correlation and Causation (C115), Residual Analysis (C116), and Time Series Basics (C117), also appear, demonstrating that CL integrates statistical reasoning into the broader curriculum pathway. Advanced calculus and linear algebra nodes, including Parametric Equations (C67), Linearization and Differentials (C127), Second-Order Linear ODEs (C144), and Diagonalisation (C162), emerge as critical connectors in higher-level pathways. While Quadratic Functions (C92)

Concept ID	Concept Name	In-Degree	Out-Degree	Total Degree
C7	Prime Numbers	3	2	5
C48	Linear Equations	3	4	7
C92	Quadratic Functions	3	4	7
C131	Derivatives	3	2	5
C3	Rounding Estimation	2	4	6
C4	Integers	2	2	4
C6	Factors Multiples	2	3	5
C110	Limits	2	1	3
C120	Polynomials	2	2	4
C150	Series	2	0	2
C160	Functions of Two Variables	2	0	2
C185	Integrals	2	1	3
C200	Differential Equations	2	0	2
C2	Place Value	1	2	3
C5	Absolute Value	1	2	3
C8	GCD LCM	1	2	3
C9	Fractions	1	2	3
C10	Decimals	1	1	2
C11	Percentages	1	1	2
C12	Ratios Proportions	1	3	4

Table 6.5: Chow–Liu (CL), Top 20 Nodes by In-Degree table.

appear again, their centrality score is lower than in the TS algorithm, suggesting that CL underrepresents its bridging role due to its tree-structure constraint.

The Chow–Liu algorithm reveals a simplified and computationally efficient curriculum structure, offering clear prerequisite pathways but with limited capacity to model the richness of interdependencies seen in the TS algorithm. In-degree results reinforce canonical prerequisite sequences: arithmetic foundations → algebra → calculus. Centrality results highlight bridging roles across number theory, algebra, and applied statistics, while surfacing connections to higher-order calculus. However, CL's restriction to tree structures produces a linearised view of the curriculum, potentially overlooking secondary or redundant prerequisites that are nonetheless relevant in real-world learning contexts. Despite these limitations, CL's clarity makes it valuable for curriculum designers seeking streamlined pathways, particularly when computational efficiency is critical or when interpretability must be prioritised over complexity.

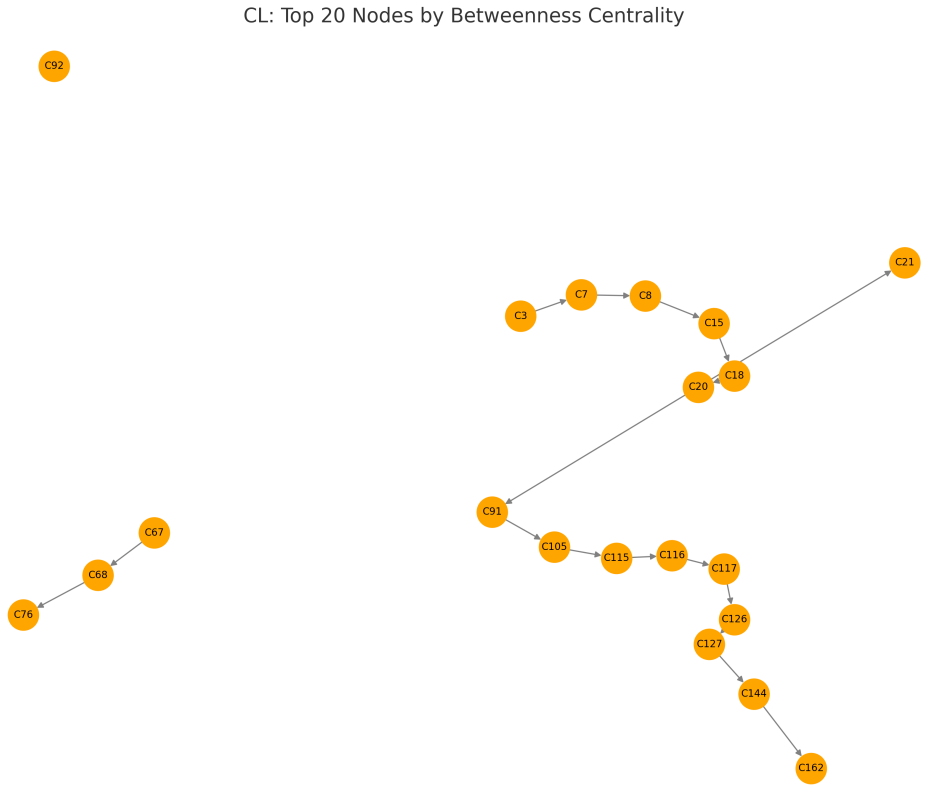


Figure 6.6: Chow–Liu (CL), Top 20 Nodes by Centrality.

6.7 Discussion

The results obtained from Hill Climbing (HC), Tabu Search (TS), and Chow–Liu (CL) provide a multifaceted perspective on the potential of Bayesian Network structure learning for curriculum extraction. Each algorithm illuminated different aspects of the mathematics curriculum, reflecting both canonical sequences and data-driven dependencies embedded in the ASSISTments dataset.

6.7.1 Alignment with Known Sequences

Across all three algorithms, the extracted curricula showed strong alignment with widely accepted mathematical progressions. Central pathways such as Arithmetic → Algebra → Calculus were consistently

Concept ID	Concept Name	Betweenness Centrality	In-Degree	Out-Degree
C8	GCD LCM	0.021033196	1	2
C7	Prime Numbers	0.020271814	3	2
C15	Exponents Powers	0.020068778	1	3
C20	Significant Figures	0.01890767	1	2
C18	Properties of Operations	0.018730014	1	3
C105	Uniform Distribution	0.013933303	1	5
C91	Polar Coordinates	0.012892747	1	1
C117	Time Series Basics	0.012232882	1	6
C115	Correlation Causation	0.012105984	1	3
C116	Residual Analysis	0.011699914	1	1
C3	Rounding Estimation	0.007994518	2	4
C21	Algebraic Expressions	0.007664586	1	3
C92	Quadratic Functions	0.007309274	3	4
C67	Parametric Equations	0.00657327	1	2
C126	Related Rates	0.00657327	1	1
C127	Linearization Differentials	0.00657327	1	1
C76	Quadrilaterals	0.006547891	1	2
C144	Second-Order Linear ODEs	0.006522512	1	1
C162	Diagonalization	0.006420994	1	1
C68	Points, Lines, Planes	0.006395614	1	1

Table 6.6: Chow–Liu (CL), Top 20 Nodes by Centrality table.

preserved, with Linear Equations (C48), Quadratic Functions (C92), and Derivatives (C131) repeatedly surfacing as key nodes. This validates the potential of Bayesian methods to recover known prerequisite structures from authentic student performance data.

Notably, the calculus sequence Limits → Derivatives → Integrals → Differential Equations was identified in both the TS algorithm and Chow–Liu results, further confirming the robustness of these methods in capturing hierarchical dependencies. These findings echo existing curricular theories, such as Bruner's spiral curriculum, which emphasises progressive deepening of foundational concepts into advanced domains.

6.7.2 Algorithmic Differences and Trade-offs

While all algorithms identified core prerequisite chains, their outputs diverged in terms of complexity, interpretability, and domain coverage. Hill Climbing (HC with BIC): Produced relatively sparse and interpretable structures. Its results emphasised clarity, identifying major hubs such as Linear Equations and Quadratic Functions without excessive redundancy. This makes HC particularly suitable for curriculum designers prioritising parsimony and transparency.

The TS algorithm (BDeu) generated denser, more interconnected networks, surfacing both expected nodes (such as Quadratic Functions, Percentages, and Ratios) and secondary but influential connectors (including Symmetry, Geometric Transformations, and Time Series Basics). While this density reflects the multi-domain nature of student learning, it also complicates interpretation. TS is therefore better suited for capturing real-world variability and alternative learning pathways, although it is less ideal for producing simplified curricular roadmaps.

Chow–Liu (CL): Offered the most computationally efficient and linearised structures, capturing essential prerequisite progressions but underrepresenting secondary dependencies. For instance, while it recovered the calculus progression, it downplayed the bridging role of Quadratic Functions compared to TS. CL's strength lies in its clarity and scalability, making it a valuable baseline approach or in contexts where computational resources are limited.

Together, these differences highlight the inherent trade-off between complexity and interpretability in Bayesian Network structure learning. A key insight from the analyses was the identification of hidden connectors, concepts not typically foregrounded in formal curricula but that are influential in linking domains. Examples include Scientific Notation (C14) in HC, Symmetry (C84) in TS, and GCD/LCM (C8) in CL. These nodes suggest that real-world student learning often relies on representational and transitional skills that may be overlooked in top-down curriculum design.

Additionally, the consistent emergence of applied statistical concepts such as Bayes' Theorem (C101), Residual Analysis (C116), and Time Series Basics (C117) underscores the integration of data-driven reasoning within mathematics education. This reflects the growing curricular emphasis on statistics and probability as essential complements to traditional algebra and calculus.

6.7.3 Implications for Curriculum Design

The findings hold several implications for curriculum design and educational practice. Validation of canonical pathways: Algorithms reliably reproduced known prerequisite sequences, confirming their potential as data-driven validation tools for existing curricula. Identification of overlooked skills: Hidden connectors suggest areas where curricula could be refined to acknowledge transitional competencies that support progression. Balancing simplicity and realism: HC and CL provide clear, streamlined sequences, while TS captures the diversity of student learning strategies. A hybrid approach may therefore be optimal, using sparse algorithms for structural clarity and dense algorithms for richness of dependen-

cies.

Despite promising results, several limitations remain. The reliance on the ASSISTments dataset means findings are shaped by the tagging of knowledge components, which may not fully capture the nuances of conceptual mastery. Furthermore, Bayesian structure learning is sensitive to sample size and scoring functions, raising the possibility that alternative parameter choices could yield different networks. Future work could explore hybrid or ensemble approaches, integrating multiple algorithms to strike a balance between interpretability and richness. Additionally, incorporating temporal information from student learning sequences could enhance the accuracy of extracted curricula, moving beyond static dependency graphs toward dynamic learning models.

6.8 Conclusion

This chapter examined the application of Bayesian Network structure learning algorithms, including Hill Climbing (HC), Tabu Search (TS), and Chow–Liu (CL), to real-world educational data from the ASSISTments platform, to directly extract curriculum structures from student performance records. The findings demonstrated that all three algorithms successfully recovered core prerequisite chains consistent with widely accepted mathematics curricula, including the progression from Arithmetic $\rightarrow$ Algebra $\rightarrow$ Calculus and the sequence Limits $\rightarrow$ Derivatives $\rightarrow$ Integrals $\rightarrow$ Differential Equations. These results validate the ability of Bayesian methods to approximate curriculum structures grounded in authentic learning data.

At the same time, significant differences emerged; HC with BIC produced sparse, interpretable structures ideal for curriculum validation. TS with BDeu revealed richer, denser networks that reflected alternative learning trajectories and hidden connectors, though at the expense of interpretability. CL provided streamlined, computationally efficient structures that preserved core progressions but simplified interdependencies. Together, these complementary perspectives highlight the trade-off between clarity and complexity in data-driven curriculum extraction. While sparse algorithms are valuable for structural insights, denser methods offer a richer picture of real-world variability in student learning.

The analyses also identified hidden connector concepts, such as scientific notation, Symmetry, and GCD/LCM, that function as critical transitional skills, yet are often underemphasized in traditional curricula. Their consistent emergence suggests the potential for refining curricular design to account for overlooked but influential dependencies. Overall, this chapter provides strong evidence that Bayesian Network structure learning can serve as a powerful tool for curriculum analysis, validation, and refinement. It bridges theoretical curricular models with data-driven insights, offering both confirmation of canonical sequences and discovery of hidden dependencies.

Chapter 7

Conclusion

This dissertation has examined whether Bayesian network structure learning can provide meaningful insights into mathematics curriculum design by directly recovering prerequisite relations from data. By treating the curriculum as a directed acyclic graph spanning approximately 200 concepts, I evaluated a variety of algorithms that differ in sparsity, search strategies, and scoring functions. These algorithms were tested in both controlled synthetic environments and with authentic learner data. The central aim was not only to reproduce familiar curricular progressions but also to uncover connections between topics that are often overlooked, and to articulate a pathway for applying learned structures in the planning, teaching, and assessment of mathematics.

In relation to the research questions posed in Chapter 1, the findings of this study provide clear and affirmative responses. The first research question, which examined the extent to which Bayesian Network structure learning can approximate a known ground-truth distribution, was addressed through controlled simulation experiments. The observed reductions in Structural Hamming Distance and Kullback–Leibler divergence with increasing sample size demonstrate consistent convergence toward the reference models. The second research question, concerning the extraction of meaningful curriculum pathways from learned structures, was addressed through analysis of both synthetic and real-world data. The resulting networks successfully recovered established prerequisite sequences while also revealing alternative and supplementary learning pathways. Together, these results confirm that the proposed methodology is capable of producing valid, interpretable, and educationally meaningful curriculum representations.

Across the experiments, the learned graphs reliably captured the backbone of conventional curricular sequences. Foundational paths, beginning in arithmetic fluency, developing through pre-algebra and algebraic manipulation, and culminating in the core ideas of calculus, were consistently recovered. On synthetic data, where the ground truth is known, the learned structures converged toward the target model as sample size increased, with both structural and distributional errors decreasing in line with statistical learning theory. On real data, the recovered edges often aligned with pedagogical expectations in ways that were demonstrably useful, while also highlighting areas where learners' actual trajectories diverged from textbook order or planned instruction.

The algorithms offered complementary perspectives. Hill Climbing with Bayesian and information-theoretic scores produced sparse, crisp structures that preserved interpretability and were well suited to validating the curricular "spine." Tabu Search, in contrast, explored the graph more broadly and admitted denser configurations that exposed secondary links and cross-strand relationships. While such density requires more careful interpretation, it often illuminated remedial or enrichment pathways that a sparse model left implicit. The Chow-Liu tree provided a conservative yet efficient summary of the prerequisite structure, making it attractive for baseline mapping and rapid iteration. Taken together, these approaches suggest a hybrid workflow in which sparse models can be used to define and audit the canon-

ical sequence. In contrast, denser models offer insight into bridging activities, revision checkpoints, and alternative learning pathways.

One of the most actionable findings was the recurring role of "connector" concepts. Topics that receive modest emphasis in syllabi — such as factorisation skills, place-value extensions, or symmetry — emerged as hubs mediating progress between larger domains. When these connectors were weak, downstream concepts appeared more difficult than necessary; when strengthened through targeted practice, formative checks, and micro-lessons, subsequent units became more accessible. This insight motivates small yet impactful changes in curriculum design, including pre-unit diagnostics that make connector skills explicit and assessment rubrics that reward connector fluency rather than assuming it as background knowledge.

There are, of course, limits to what can be inferred from the present pipeline. The fidelity of tagging, the coverage of concepts, and modelling choices such as priors and scoring penalties all shape the resulting structures. Static graphs cannot fully represent temporal dynamics such as spacing, forgetting, or transfer. Nevertheless, the empirical pattern across both synthetic and real data is encouraging: as data quality and scale improve, the learned graphs become increasingly stable and informative. This makes it possible to justify an iterative "learn–validate–refine" loop in curriculum design, where structures are periodically re-estimated on new cohorts, compared against the existing map, and then used to adjust pacing and content where evidence warrants.

The broader implication is that Bayesian structure learning has the potential to move curriculum conversations beyond anecdote and intuition, without displacing teacher judgment. It offers a way for departments to formalise intended dependencies, test them against learner evidence, and decide when to preserve, revise, or enrich sequences. Teachers gain a defensible rationale for pacing and prerequisite checks; learners benefit from targeted support on key connectors; and leaders obtain compact diagnostics of where designed curricula diverge from enacted learning. Because the graphs are transparent objects rather than black-box scores, they invite collaborative interpretation and adaptation. This study demonstrates that Bayesian networks provide a practical and evidence-aligned approach to curriculum design in mathematics. They reaffirm the coherence of the backbone that sustains cumulative instruction, while at the same time revealing quieter hinges upon which progress depends.

Looking forward, there are several directions in which this research can be extended. Dynamic Bayesian networks could be employed to capture the temporal effects of practice, spacing, and forgetting, offering richer models of how students progress over time. Cross-domain applications, such as those in science or language learning, would test the generalizability of the approach beyond mathematics. Hybrid ensembles that reconcile the strengths of sparse and dense structures could be developed to strike a balance between interpretability and coverage. At the same time, large-scale deployments across multiple cohorts would help establish robustness and generalisability.

There is also scope for designing teacher-centred tools that make these learned structures accessible in practice, providing intuitive interfaces for educators to visualise, validate, and adapt the prerequisite maps. Embedding these models in adaptive learning platforms could enable sequencing to be adjusted in real-time, providing students with tailored pathways that remain aligned with evidence-based curricular design.

Finally, as part of this forward trajectory, work is already underway to prepare a journal article based on the dissertation's findings. The intention is to disseminate both the methodological contributions and the pedagogical implications to the broader educational data mining and curriculum design communities. This marks the beginning of a sustained research programme that builds on the present study and seeks to make data-validated curriculum design a routine capability in education systems. The study

demonstrates that the stated research aims and objectives have been achieved through systematic experimentation, rigorous evaluation, and educational interpretation of the learned models.

Appendix A: Pseudocode for Bayesian Network and Evaluation Metrics

Algorithm 1 Pseudocode To Construct Ground Truth Bayesian Network

Require: Concept set $C = \{C_1, C_2, \ldots, C_n\}$, relationships among concepts
Ensure: Ground truth Bayesian network $B^* = \langle G^*, \Theta \rangle$
1: Define nodes of the network as concepts in C.
2: Construct directed edges in G^* based on known relationships.
3: Assign parameters Θ to the edges (conditional probability distributions).
4: Validate B^* as a Directed Acyclic Graph (DAG).
5: **return** B^*

Algorithm 2 Pseudocode To Sample Dataset from Bayesian Network

Require: Ground truth Bayesian network $B^* = \langle G^*, \Theta \rangle$, sample size N
Ensure: Sampled dataset $D = \{\xi_1, \xi_2, \ldots, \xi_N\}$
1: **for** each instance $i = 1$ to N **do**
2: **for** each node C_j in topological order of G^* **do**
3: Sample value for C_j based on parent nodes' distributions in Θ.
4: **end for**
5: Generate instance ξ_i.
6: **end for**
7: **return** D

Algorithm 3 Pseudocode For Hill Climbing with BDeu/BIC/K2

Require: Dataset D, scoring metric S, max iterations T
Ensure: Learned Bayesian network $B^+ = \langle G^+, \Theta^+ \rangle$
1: Initialize G^+ as an empty graph.
2: Compute initial score $S(G^+, D)$.
3: **for** iteration $t = 1$ to T **do**
4: Generate neighbors of G^+ by:
5: Adding, removing, or reversing an edge.
6: Compute S' for each neighbor.
7: Update G^+ to the neighbor with the highest S' if $S' > S(G^+, D)$.
8: Update parameters Θ^+ using Maximum Likelihood Estimation (MLE).
9: **end for**
10: **return** $B^+ = \langle G^+, \Theta^+ \rangle$

Algorithm 4 Pseudocode For Chow-Liu Algorithm

Require: Dataset D, concept set $C = \{C_1, C_2, \ldots, C_n\}$
Ensure: Tree-structured Bayesian network $B^+ = \langle G^+, \Theta^+ \rangle$
1: Compute pairwise mutual information $MI(C_i, C_j)$ for all pairs of nodes.
2: Construct a complete graph where edges are weighted by $MI(C_i, C_j)$.
3: Apply maximum spanning tree algorithm to find G^+.
4: Orient edges in G^+ to form a DAG.
5: Estimate parameters Θ^+ using Maximum Likelihood Estimation (MLE).
6: **return** $B^+ = \langle G^+, \Theta^+ \rangle$

Algorithm 5 Pseudocode To Compute KL Divergence

Require: Ground truth B^*, learned network B^+, dataset D
Ensure: KL Divergence score
1: **for** each instance ξ in D **do**
2: Compute $P(\xi|B^*)$ using G^* and Θ.
3: Compute $P(\xi|B^+)$ using G^+ and Θ^+.
4: Compute $KL(\xi) = P(\xi|B^*) \cdot \log \frac{P(\xi|B^*)}{P(\xi|B^+)}$.
5: **end for**
6: **return** Average KL score across all instances.

Algorithm 6 Pseudocode To Compute Structural Hamming Distance (SHD)

Require: Ground truth structure G^*, learned structure G^+
Ensure: SHD score
1: Initialize $SHD = 0$.
2: **for** each edge $(C_i \rightarrow C_j)$ in G^* **do**
3: **if** $(C_i \rightarrow C_j)$ not in G^+ **then**
4: Increment SHD by 1.
5: **end if**
6: **end for**
7: **for** each edge $(C_i \rightarrow C_j)$ in G^+ **do**
8: **if** $(C_i \rightarrow C_j)$ not in G^* **then**
9: Increment SHD by 1.
10: **end if**
11: **end for**
12: **return** SHD

Appendix B: Ground Truth Graph

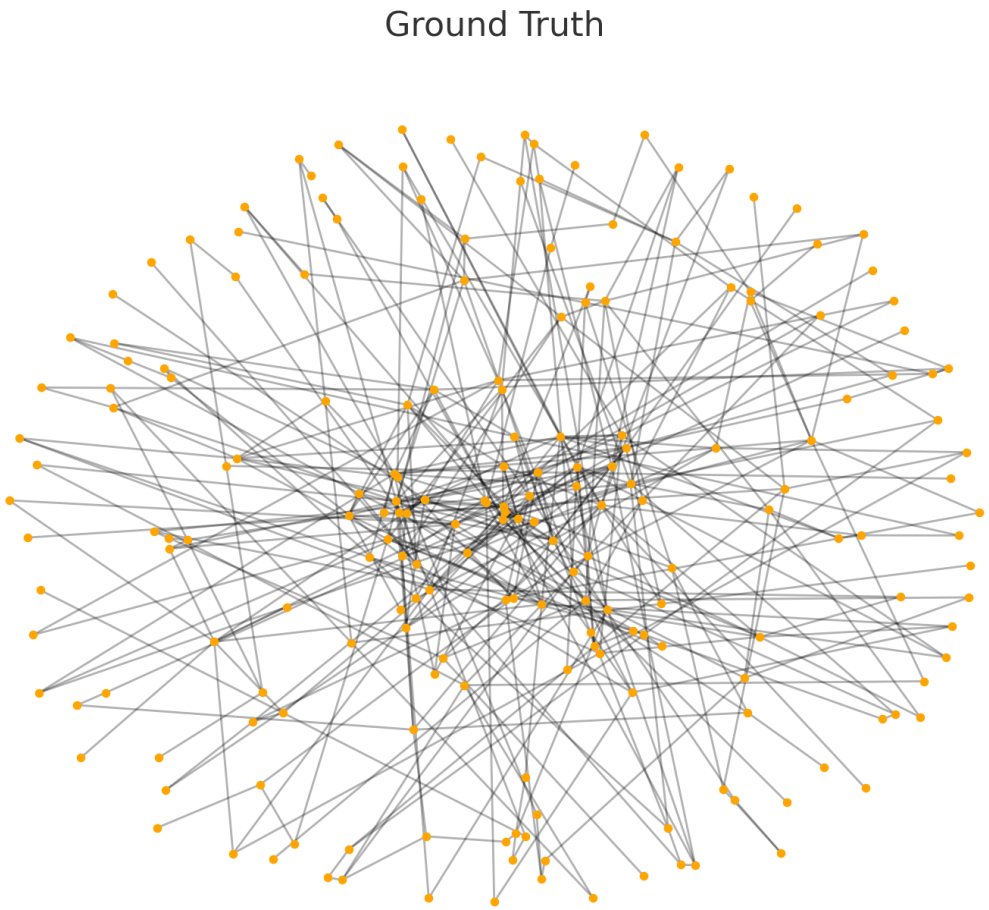


Figure 1: This figure represents the ground truth graph, a reference structure showcasing the true dependencies among variables in the dataset. The orange nodes indicate individual variables, and the directed edges define the prerequisite relationships or causal dependencies. This graph is the benchmark for evaluating the accuracy and fidelity of learned graphical models. The structure reflects the explicit relationships that guide the curriculum or sequence of dependencies for the analyzed concepts.

Appendix C: Complete Tabu Search

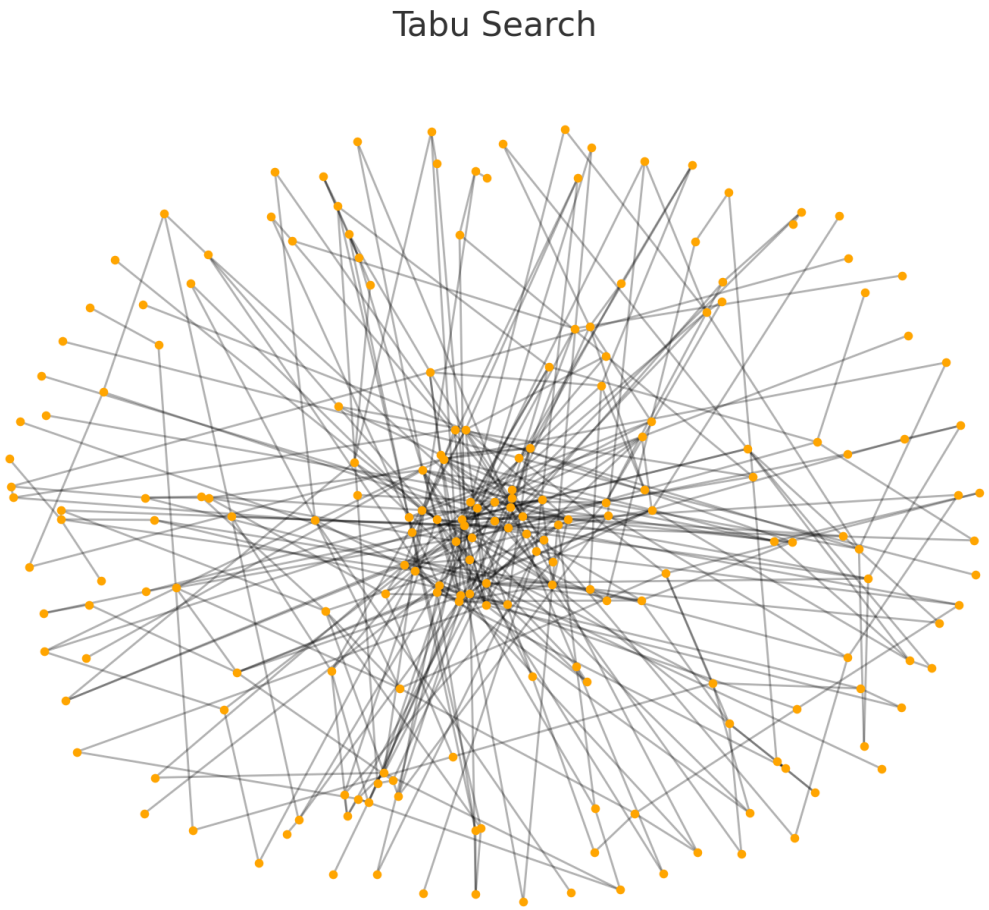


Figure 2: illustrates a complete Chow-Liu model.

Appendix D: Complete Hill-Climbing(BIC)

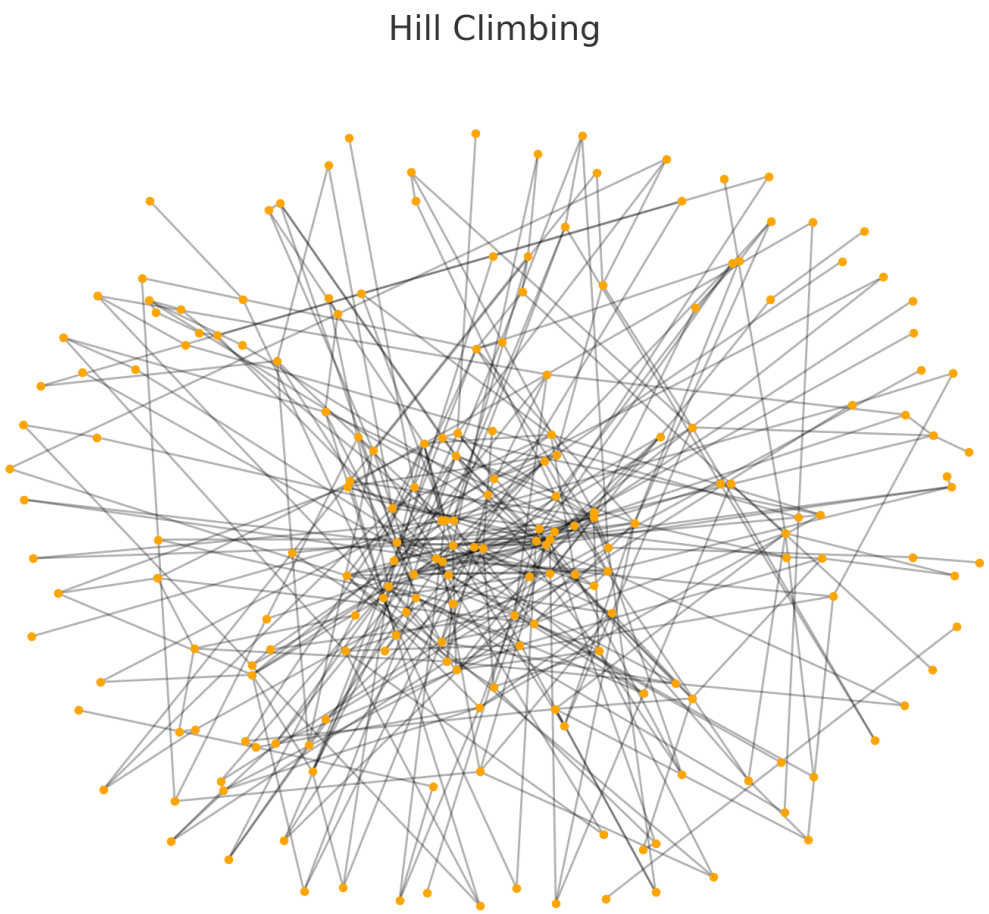


Figure 3: illustrates a complete Hill Climbing with BIC.

Appendix D: Complete Chow-Liu

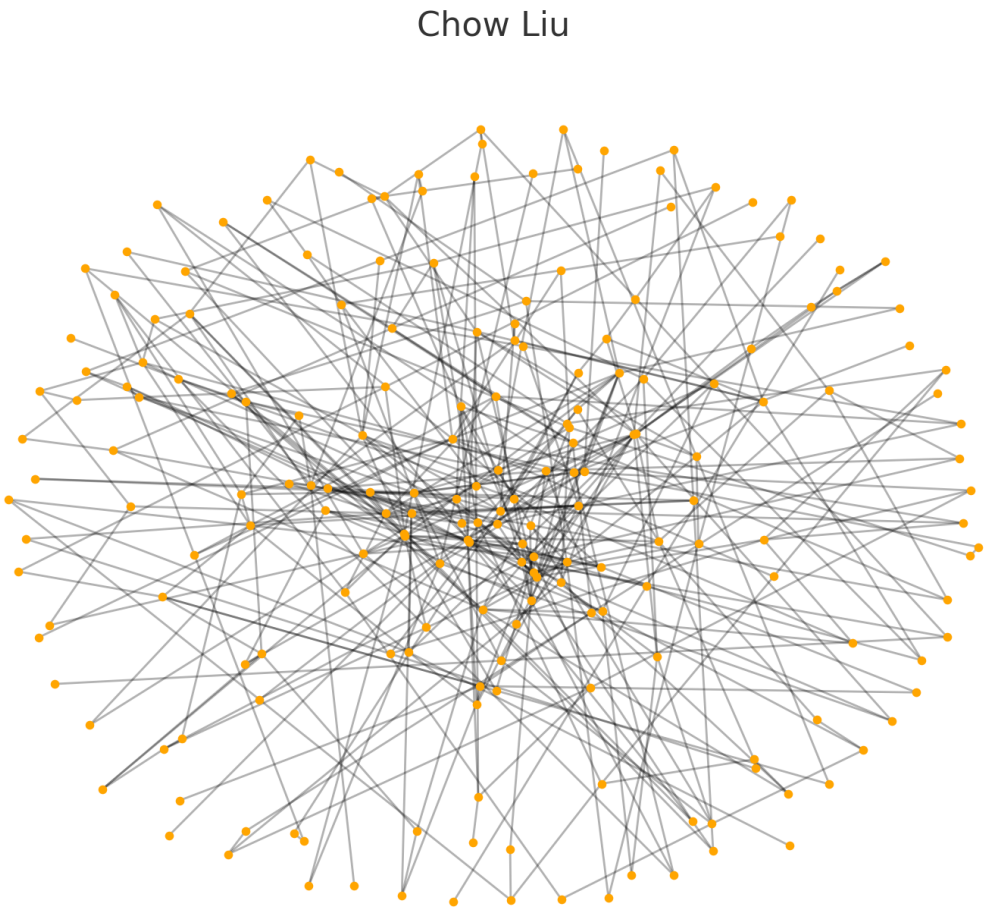


Figure 4: illustrates a complete Chow-Liu.

Appendix D: Complete Simulated Annealing with BIC

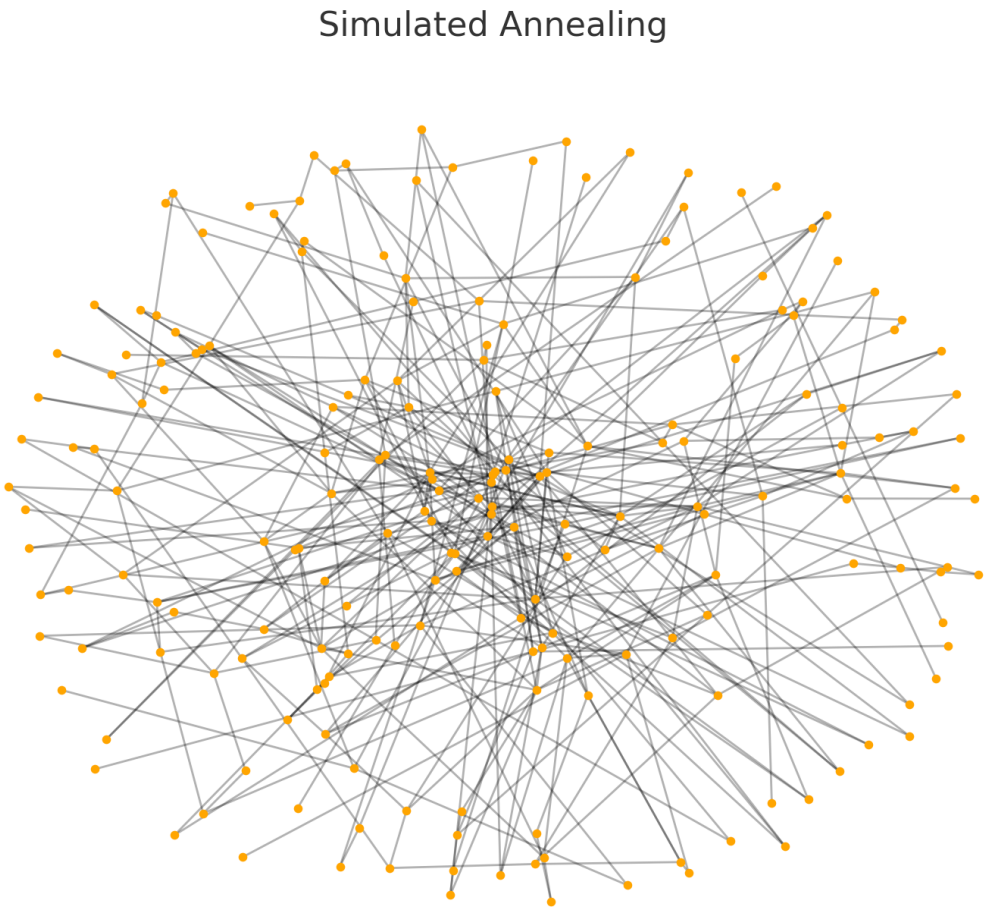


Figure 5: illustrates a complete Simulated Annealing

References

[ADvTECH Group 2023] ADvTECH Group. *Maths vs Maths Literacy: Why it's worth giving Pure Maths your best shot.* https://www.advtech.co.za/maths-vs-maths-literacy-why-its-worth-giving-pure-maths-your-best-shot, 2023. Accessed: 2025-03-03.

[Ajoodha and Rosman 2017] Ritesh Ajoodha and Benjamin Rosman. Tracking influence between naïve bayes models using score-based structure learning. In *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pages 122–127. IEEE, 2017.

[Ajoodha and Rosman 2018] Ritesh Ajoodha and Benjamin Rosman. Learning the influence structure between partially observed stochastic processes using iot sensor data. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Ajoodha 2019] Ritesh Ajoodha. Influence modelling and learning between dynamic bayesian networks using score-based structure learning. 2019.

[Backenkoehler *et al.* 2018] M. Backenkoehler, F. Scherzinger, and A. Singla. Data-driven approach towards a personalized curriculum. In *Educational Data Mining Conference*, 2018.

[Baker *et al.* 2008] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings 9*, pages 406–415. Springer, 2008.

[Baker *et al.* 2019] Ryan S. Baker, George Siemens, and Carolyn P. Rosé. Advances in educational data mining: Past, present, and future. *Journal of Educational Data Mining*, 11(1):1–17, 2019.

[Barmpas and Tasoulis 2024a] Georgios Barmpas and Dimitrios Tasoulis. Structural hamming distance for feature selection in graph-based learning. *Pattern Recognition*, 142:109567, 2024.

[Barmpas and Tasoulis 2024b] P. Barmpas and S. K. Tasoulis. Hyperdimensional computing approaches in single cell rna sequencing classification. In *2024 IEEE Congress*, 2024.

[Beck and Chang 2007] Joseph E. Beck and Ke Chang. Identifiability: A fundamental problem of student modeling. *Proceedings of the 11th International Conference on User Modeling*, pages 137–146, 2007.

[Bouckaert 1995] Remco R. Bouckaert. Bayesian belief networks: From construction to inference. In *Proceedings of the Workshop on Artificial Intelligence and Statistics*, 1995.

[Boutilier and Chan 2023] J.J. Boutilier and T.C.Y. Chan. Introducing and integrating machine learning in an operations research curriculum: An application-driven course. *INFORMS Transactions on Education*, 2023.

[Brusilovsky and Millán 2007] Peter Brusilovsky and Eva Millán. User models for adaptive hypermedia and adaptive educational systems. *The Adaptive Web*, pages 3–53, 2007.

[Bucci *et al.* 2023] M. Bucci, A. Soreme, S. Alluzen, and S. Chibbaro. Curriculum learning for data-driven modeling of dynamical systems. *The European Physical Journal*, 2023.

[Busemeyer and Wang 2021] Jerome R. Busemeyer and Zheng Wang. Cognitive models for learning: Limitations and challenges in synthetic data generation. *Cognitive Science*, 45(4):e12987, 2021.

[Chaturvedi *et al.* 2020] Indranil Chaturvedi, Dan Goldwasser, and Noboru Matsuda. Data-driven reinforcement learning for adaptive tutoring systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13525–13532, 2020.

[Chen *et al.* 2017] Y. Chen, T. Wheeler, and M. Kochenderfer. Learning discrete bayesian networks from continuous data. *Journal of Artificial Intelligence Research*, 59:103–132, 2017.

[Chen *et al.* 2022] Li Chen, Xiaoyan Sun, and Rui Zhang. Structural hamming distance for evaluating concept maps in adaptive learning. *Computers Education*, 179:104417, 2022.

[Cheng *et al.* 2002] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137:43–90, 2002.

[Chickering 2002] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[Choi *et al.* 2022] M. Choi, S. Kim, and J. Park. Kl regularization for bayesian structure learning. *Bayesian Analysis*, 17(2):673–692, 2022.

[Chow and Liu 1968] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[Cliff *et al.* 2018] Oliver M Cliff, Mikhail Prokopenko, and Robert Fitch. Minimising the kullback–leibler divergence for model selection in distributed nonlinear systems. *Entropy*, 20(2):51, 2018.

[Colombo and Maathuis 2012] Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15:3921–3962, 2012.

[Conati and VanLehn 2002] Cristina Conati and Kurt VanLehn. Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.

[Constantinou *et al.* 2016] Anthony C. Constantinou, Norman Fenton, and Martin Neil. Integrating expert knowledge with data in bayesian networks: Preserving data-driven expectations. *Expert Systems with Applications*, 56:197–208, 2016.

[Cover and Thomas 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, 2 edition, 2006.

[de Campos 2006] Cassio P. de Campos. A scalable method for learning bayesian networks from data. *Machine Learning*, 65(1):37–66, 2006.

[Desmarais and Baker 2012] Michel C. Desmarais and Ryan S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[Doroudi *et al.* 2019] Shayan Doroudi, Kevin Man, and Emma Brunskill. Fairer but not fair enough: On the equitability of knowledge tracing. *Proceedings of the ACM Conference on Learning at Scale*, pages 1–10, 2019.

[Du 2022] Y. Du. Application of the data-driven educational decision-making system to curriculum optimization of higher education. *Wireless Communications and Mobile Computing*, 2022.

[Feng and Lee 2017] X. Feng and C. Lee. Assessing the linguistic relevance of learned speech units using kl divergence. *Speech Communication*, 92:42–50, 2017.

[Gao and Ji 2023] Yi Gao and Xiaoqiang Ji. Deep generative models for synthetic data generation in structure learning. *Neural Networks*, 160:45–60, 2023.

[Ghahramani and Jordan 2023] Z. Ghahramani and M. Jordan. Evaluating bayesian neural network approximations using kl divergence. *Neural Computation*, 35:1124–1145, 2023.

[Ghasemian *et al.* 2024a] Amir Ghasemian, Alireza Farahani, and Pouya Rajabi. Privacy-preserving genomic analysis using structural hamming distance. *Nature Communications*, 15:3456, 2024.

[Ghasemian *et al.* 2024b] M. Ghasemian, L. H. Gerido, and E. Ayday. Safeguarding privacy in genome research: A comprehensive framework for authors. *bioRxiv*, 2024.

[Glover 1989] Fred Glover. Tabu search—part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[Glover 1990] Fred Glover. Tabu search—part ii. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[Glymour *et al.* 2016] Clark Glymour, Kun Zhang, and Peter Spirtes. *Causal Discovery and Inference*. MIT Press, 2016.

[Glymour *et al.* 2019] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

[Goguadze *et al.* 2018] Galia Goguadze, Erica Melis, and Carsten Ullrich. A competency-based knowledge structure for curriculum sequencing. *International Journal of Artificial Intelligence in Education*, 28:1–28, 2018.

[Gong and Zhai 2020] Yuxin Gong and Xiaoming Zhai. Comparing bayesian networks with deep learning approaches for educational data mining. *International Journal of Artificial Intelligence in Education*, 30:1–22, 2020.

[Gong *et al.* 2010] Y. Gong, J. Beck, and N. Heffernan. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Conference on Intelligent Tutoring Systems*, pages 35–44, 2010.

[Heckerman 1998] David Heckerman. A tutorial on learning with bayesian networks. pages 301–354, 1998.

[Hellewell *et al.* 2020] Joel Hellewell, Sam Abbott, and Amy Gimma. Feasibility of controlling covid-19 outbreaks by isolation of cases and contacts. *The Lancet Global Health*, 8:e488–e496, 2020.

[Huang *et al.* 2021] Xiaoyuan Huang, Qing Liu, Qiang Wu, and Enhong Chen. Generative adversarial networks for synthetic student response generation. *Proceedings of the International Conference on Educational Data Mining (EDM)*, 14(1):153–164, 2021.

[Hui 2024] C. Hui. Ai gen assistment: Analyzing the effectiveness of generative ai data amplification for dkt. In *Second Tiny Papers Track at ICLR*, 2024.

[Jensen 2001] F. Jensen. Bayesian networks and decision graphs. 2001.

[Jurman *et al.* 2015a] Giuseppe Jurman, Silvana Riccadonna, Roberto Visintainer, and Cesare Furlanello. The structural hamming distance as a graph dissimilarity measure. *Scientific Reports*, 5:14843, 2015.

[Jurman *et al.* 2015b] Giuseppe Jurman, Roberto Visintainer, Michele Filosi, Samantha Riccadonna, and Cesare Furlanello. The him glocal metric and kernel for network comparison and classification. In *2015 IEEE international conference on data science and advanced analytics (DSAA)*, pages 1–10. IEEE, 2015.

[Khajah *et al.* 2016] Mohammad M. Khajah, Richard V. Lindsey, and Michael C. Mozer. How deep is knowledge tracing? *Journal of Educational Data Mining*, 8(1):1–20, 2016.

[Kirkpatrick *et al.* 1983] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[Kobayashi 2022a] T. Kobayashi. Incorporating kl divergence into reinforcement learning algorithms. *Neural Computation*, 34:1754–1778, 2022.

[Kobayashi 2022b] Taisuke Kobayashi. Optimistic reinforcement learning by forward kullback-leibler divergence optimization. *Neural Networks*, 152:169–180, 2022.

[Koedinger *et al.* 2015] Kenneth R. Koedinger, Julie L. Booth, and David Klahr. Instructional complexity and the science to constrain it. *Science*, 342:935–937, 2015.

[Koller and Friedman 2009a] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Koller and Friedman 2009b] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[Kosun 2021] D. Kosun. Analyzing vehicular speed characteristics using kl divergence and entropy-based methods. *Transportation Research Part C*, 124:102908, 2021.

[Kuldas *et al.* 2014] Seffetullah Kuldas, Hairul Nizam Ismail, and Shahabuddin Hashim. Cognitive load and learning preferences: Predicting academic achievements. *Educational Psychology*, 34(5):659–674, 2014.

[Kullback and Leibler 1951] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[Lee and Brunskill 2012] Jung In Lee and Emma Brunskill. The impact on individualizing student models on necessary practice opportunities. *International Educational Data Mining Society*, 2012.

[Li and Guo 2018] Hongru Li and Huiping Guo. A hybrid structure learning algorithm for bayesian network using experts' knowledge. *Entropy*, 20(8):620, 2018.

[Li *et al.* 2021] Qian Li, Hyeree Cho, Jimena Cosso, and Yukiko Maeda. Relations between students' mathematics anxiety and motivation to learn mathematics: A meta-analysis. *Educational Psychology Review*, pages 1–33, 2021.

[Liang *et al.* 2018] J. Liang, T. Zhang, and L. J. Li. Mentornet: Learning data-driven curriculum for very deep neural networks. In *Proceedings of the International Conference on Machine Learning*, 2018.

[Liu 2024] J. Liu. Semi-supervised anomaly detection using kl divergence. *Journal of Machine Learning Research*, 25:1–23, 2024.

[Machaba and Du Plooy 2019] France Machaba and Maryna Du Plooy. Mathematics and mathematical literacy on the career podium–sharing gold? *African Journal of Research in Mathematics, Science and Technology Education*, 23(3):363–375, 2019.

[Mekić *et al.* 2024] E. Mekić, K. Fetahović, and B. Popović. Data-driven educational decision-making model for curriculum optimization. *Ethics in Online AI-Based Education*, 2024.

[Millán and de-la Cruz 2010] Eva Millán and José L. Pérez de-la Cruz. Bayesian networks for student model engineering. *Computers Education*, 55(4):1663–1683, 2010.

[Mislevy 2003] Robert J. Mislevy. A brief introduction to evidence-centered design. *Journal of Technology, Learning, and Assessment*, 1(1):1–35, 2003.

[Mokhtari *et al.* 2022] Ramin Mokhtari, Nima Alizadeh, and Hamed Darabi. Weighted structural hamming distance for causal inference models. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5678–5691, 2022.

[Murphy 2012] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

[Nagarajan *et al.* 2013] Radhakrishnan Nagarajan, Marco Scutari, and Sandra Lebre. Bayesian networks in r: With applications in systems biology. *Springer*, 2013.

[Nazaritiji 2024a] F. Nazaritiji. Genomic clustering and comparative analysis of sars-cov-2 variants and subvariants in year 2022. 2024.

[Nazaritiji 2024b] Sarah Nazaritiji. Sars-cov-2 variant clustering using structural hamming distance. *Genome Biology*, 25:201, 2024.

[of Basic Education and Training 2023] Department of Basic Education and Training. *National Senior Certificate Examination Reports: 2021–2023*, 2023.

[Pearl and Mackenzie 2018] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018.

[Pedro *et al.* 2013] M. Pedro, R. Baker, N. Bowers, and N. Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Educational Data Mining Conference*, 2013.

[Piech *et al.* 2015a] C. Piech, J. Bassen, and J. Huang. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 2015.

[Piech *et al.* 2015b] Chris Piech, Jonathan Spencer, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. *Deep Knowledge Tracing*, 2015.

[Rafferty *et al.* 2016] Anna N. Rafferty, Emma Brunskill, and Thomas L. Griffiths. Inferring learners' knowledge from their actions. *Cognitive Science*, 40(3):812–862, 2016.

[Ramsey *et al.* 2012] Joseph Ramsey, Kun Zhang, and Peter Spirtes. Adjacency-faithfulness and conservative causal inference. *UAI*, 2012.

[Razzaza *et al.* 2007] L. Razzaza, N. T. Heffernan, and M. Feng. Developing fine-grained transfer models in the assistment system. *Instruction, Cognition Learning*, 2007.

[Ritter *et al.* 2022] Stefan Ritter, John R. Anderson, and Kenneth R. Koedinger. Challenges in modeling student engagement using synthetic datasets. *International Journal of Artificial Intelligence in Education*, 32:210–230, 2022.

[Scanagatta *et al.* 2019] Mauro Scanagatta, Antonio Salmerón, and Fabio Stella. A survey on bayesian network structure learning from data. *Progress in Artificial Intelligence*, 8(4):425–439, 2019.

[Scutari and Denis 2021] Marco Scutari and Jean-Baptiste Denis. Bayesian networks: With examples in r. *CRC Press*, 2021.

[Scutari *et al.* 2019] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115:235–253, 2019.

[Scutari 2010] Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(3):1–22, 2010.

[Sebatana 2023] J. Sebatana. *Expert warns of long-term risks as schools drop Pure Mathematics*. `https://news.nwu.ac.za/`

expert-warns-long-term-risks-schools-dropping-pure-mathematics, 2023. Accessed: 2025-03-03.

[Selent *et al.* 2016] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third ACM Conference on Learning*, 2016.

[Shen 2024] W. Shen. Utilizing machine learning to optimize the data-driven talent cultivation of higher education in the digital age. *Conference on Computational Finance and Business*, 2024.

[Shute and Rahimi 2017] Valerie J. Shute and Sana Rahimi. Review of computer-based assessment for learning. *Educational Psychologist*, 52(4):233–256, 2017.

[Si 2021] L. Si. Using assistments to help teachers recover covid-19-related learning loss. *DigitalCommons*, 2021.

[Song *et al.* 2024] C. Song, K.S. Shin, and K. Binh. Implementing the dynamic feedback-driven learning optimization framework: A machine learning approach to personalize educational pathways. *Applied Sciences*, 2024.

[Spirtes *et al.* 2000] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2000.

[Stamper and Barnes 2010] John Stamper and Tiffany Barnes. Synthetic student data for analyzing learning progressions and prerequisite relationships. In *International Conference on Educational Data Mining*, pages 99–108, 2010.

[Trabelsi *et al.* 2013a] Amine Trabelsi, Mohamed Ben Ayed, and Ridha Bouaziz. Temporal structural hamming distance for dynamic bayesian networks. *Journal of Machine Learning Research*, 14:1527–1554, 2013.

[Trabelsi *et al.* 2013b] Ghada Trabelsi, Philippe Leray, Mounir Ben Ayed, and Adel M Alimi. Benchmarking dynamic bayesian network structure learning algorithms. In *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pages 1–6. IEEE, 2013.

[Tsamardinos *et al.* 2006a] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[Tsamardinos *et al.* 2006b] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

[Van de Sande 2013] Carla Van de Sande. The role of student misconceptions in learning. *Educational Psychology Review*, 25(3):377–404, 2013.

[Van der Linden and Glas 2010] Wim J. Van der Linden and Cees A. W. Glas. Elements of adaptive testing. *Springer*, 2010.

[Wang and Sun 2021] Y. Wang and H. Sun. Efficient bayesian network structure learning via heuristic search. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Wang *et al.* 2023] Huanhuan Wang, Ahmed Tlili, Ronghuai Huang, Zhenyu Cai, Min Li, Zui Cheng, Dong Yang, Mengti Li, Xixian Zhu, and Cheng Fei. Examining the applications of intelligent tutoring systems in real educational contexts: A systematic literature review from the social experiment perspective. *Education and information technologies*, 28(7):9113–9148, 2023.

[Yang and Wei 2024] F. Yang and Y. Wei. Data-driven digital literacy assessment and optimization of college physical education teachers: An application study of machine learning algorithms. *Journal of Electrical Systems*, 2024.

[Yaramala 2023] A. Yaramala. Analysis of student behavior and score prediction in assistments online learning. *DigitalCommons*, 2023.

[Yu and Wang 2022] Wei Yu and Xin Wang. Regularized hybrid bayesian network learning. *Journal of Machine Learning Research*, 23:1–26, 2022.

[Yudelson *et al.* 2013] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer Berlin Heidelberg, 2013.

[Zhang *et al.* 2017] Jing Zhang, Xiaoxiao Shi, Xing Xie, and Qiang Xu. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 765–774. ACM, 2017.

[Zhao *et al.* 2023] Tian Zhao, Wei Liu, and Fang Zhang. Integrating structural hamming distance with deep reinforcement learning for graph-based decision-making. *Neural Networks*, 161:42–58, 2023.

[Zhu and Zhao 2020] L. Zhu and W. Zhao. Reinforcement learning for bayesian network optimization. *Neurocomputing*, 400:189–200, 2020.

[Zhu *et al.* 2023] Ming Zhu, Yifan Chen, and Lijun Wang. Secure multi-party computation in computational biology: Advances and challenges. *Bioinformatics*, 39(12):987–1002, 2023.

[Zhu *et al.* 2024] Y. Zhu, P. V. Benos, and M. Chikina. A hybrid constrained continuous optimization approach for optimal causal discovery from biological data. *Bioinformatics*, 2024.

[Zia *et al.* 2023] X. Zia, N. Dong, J. Wu, and C. Ma. Multi-variate knowledge tracking based on graph neural network in assistments. *IEEE Transactions on Learning*, 2023.