

https://dspace.org/

Query Specific Pruning Strategies in Bayesian Networks for Inference Efficiency

Chirkoot, Mikhail

2025-02
University of the Witwatersrand, Johannesburg
https://hdl.handle.net/10539/50203
Chirkoot, Mikhail. (2025). Query Specific Pruning Strategies in Bayesian Networks for Inference Efficiency. [Master's dissertation, University of the Witwatersrand, Johannesburg]. WIReDSpace. https://hdl.handle.net/10539/50203

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



UNIVERSITY OF THE WITWATERSRAND

SCHOOL OF COMPUTER SCIENCE AND APPLIED MATHEMATICS

Query Specific Pruning Strategies in Bayesian Networks for Inference Efficiency

Masters Dissertation

Mikhail Chirkoot 2339279

Supervised by Prof. Ritesh Ajoodha & Dr. Pravesh Ranchod

A dissertation submitted to the School of Computer Science and Applied Mathematics
in partial fulfillment of the requirements for the degree of Masters in Computer Science

February 4, 2025

ExplainableAI Lab
— MODELLING. DECISION MAKING. CAUSALITY —

Declaration

I, Mikhail Chirkoot, hereby declare the contents of this dissertation to be my own work. This dissertation is submitted for the degree of Master of Science in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.



6/26/2025

Contents

1 Introduction **2**

2 Background **5**
2.1 Bayesian Networks . . . 5
2.2 Inference in Bayesian Networks . . . 5
2.3 Importance of Efficient Inference . . . 6
2.4 Conclusion . . . 6

3 Related Work **7**
3.1 Exact Inference in Bayesian Networks . . . 8
3.1.1 The Computational Complexity of Exact Inference . . . 8
3.1.2 Why Exact Inference Remains Relevant . . . 8
3.1.3 Efficient Elimination Orders . . . 9
3.1.4 Utilizing Sparsity in Bayesian Networks for Scalable Inference . . . 12
3.1.5 Hybrid Approaches: Combining Exact and Approximate Methods . . . 14
3.2 Pruning Strategies in Bayesian Networks . . . 16
3.3 Conclusion . . . 18

4 Methodology **20**
4.1 Problem Statement . . . 20
4.2 Experimental Setup . . . 21
4.3 Generation of ground truth networks ($\mathcal{M}^*$) . . . 22
4.3.1 Ground Truth Network Characteristics . . . 23
4.4 Pruning Techniques . . . 25
4.4.1 Basic Pruning . . . 25
4.4.2 Refined Pruning . . . 27
4.4.3 V-Structure Pruning . . . 28
4.4.4 Separator-Based Pruning . . . 30
4.4.5 Cluster-Based Pruning . . . 31
4.5 Comparison of Pruning Methods . . . 32
4.6 Evaluation Methodology . . . 33
4.6.1 Experimental Setup and Metrics . . . 33
4.6.2 Experimental Procedure . . . 34
4.6.3 Visualizing Results . . . 35

5 Results **36**
5.1 RQ1: How does query-specific pruning impact computational efficiency and accuracy in high-dimensional Bayesian networks? . . . 36

5.1.1 Comparison of Pruning Methods . . . 40
5.2 Analysis of Research Question 2: The Impact of Evidence Amounts on Pruning Effectiveness . . . 41
5.2.1 Evaluation of Pruning Methods at Different Evidence Levels . . . 41
5.2.2 Observations and Trends . . . 45
5.3 Analysis of Research Question 3: Pruning Effectiveness on Real World Data . . . 47

6 Discussion **55**
6.1 Detailed Evaluation of Pruning Methods . . . 55
6.1.1 Effectiveness of Basic and V-Structure Pruning . . . 55
6.1.2 Limitations of Separator and Cluster Pruning . . . 55
6.1.3 Challenges with Refined Pruning . . . 56
6.1.4 Impact of Evidence Levels . . . 56
6.1.5 Real-World Applications and Dataset-Specific Insights . . . 57
6.2 Thorough Evaluation of Research Problems . . . 57
6.3 Significance of the Research . . . 57
6.4 Contributions . . . 57
6.5 Limitations of the Study . . . 58
6.6 Impact of Technical Contributions . . . 59
6.7 Comparison to Research Questions . . . 59

7 Conclusion **60**

References **64**

Abstract

Bayesian networks (BNs) serve as a powerful tool for encoding probabilistic relationships among variables and facilitating inferential tasks. However, the complexity and size of these networks can pose significant challenges, particularly in scenarios demanding real-time decision-making. This research introduces query-specific pruning strategies designed to improve the efficiency of inference processes in Bayesian networks. By selectively pruning nodes and edges that are irrelevant to a given query, these strategies considerably reduce the computational overhead without compromising the accuracy of the inference results.

Our approach uniquely used several pruning techniques, each tailored to different aspects of the network's structure and the specifics of the query and evidence presented. These include basic pruning, refined pruning, V-structure pruning, and community-based pruning, all of which have been systematically evaluated to assess their impact on inference cost and accuracy. The effectiveness of these methods is demonstrated through a series of experiments conducted on synthetic networks as well as on real-world data.

The results reveal that query-specific pruning significantly outperforms traditional inference methods namely variable elimination, especially in large-scale networks, by efficiently managing the scope of the probabilistic inference task. This not only accelerates the inference process but also retains high fidelity in the results, thus supporting robust decision-making in critical applications such as healthcare. Through this research, we aim to provide a methodological framework that can be adapted to various domains, promoting more efficient use of Bayesian networks in practical settings.

Chapter 1

Introduction

Imagine a scenario in which robots or intelligent software systems continuously observe and learn from their environments, much like humans do through their senses. These systems interact with dynamic surroundings, accumulating vast amounts of data over time. To make sense of this ever-growing dataset, advanced models capable of reasoning under uncertainty are required. Bayesian networks (BNs), with their ability to encode complex probabilistic relationships, provide a powerful framework for decision-making in such contexts. However, the computational challenges associated with maintaining and updating these networks as they grow in size and complexity remain a significant barrier to their widespread deployment in real-world scenarios.

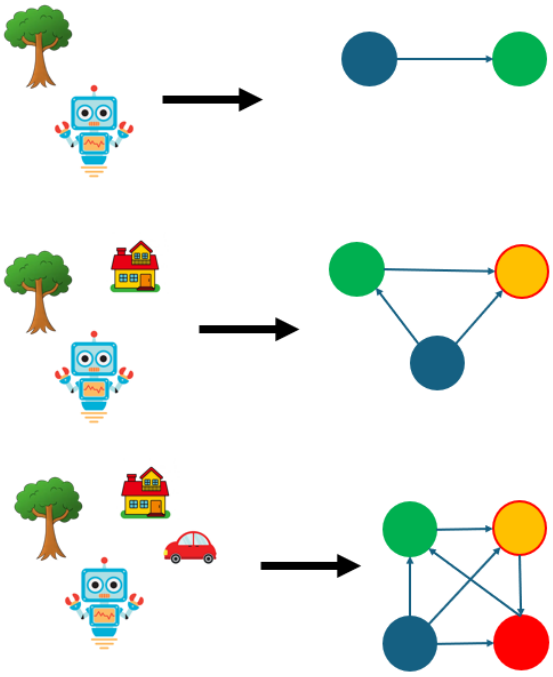


Figure 1.1: Progressive Complexity in Autonomous Learning: From Initial Observations to Advanced Bayesian Networks.

Bayesian networks are probabilistic graphical models that represent variables and their conditional dependencies through a directed acyclic graph (DAG). Each node in the DAG corresponds to a variable,

while the edges encode the probabilistic dependencies between these variables. This structure not only provides a compact representation of joint probability distributions but also enables interpretable and explainable reasoning, making Bayesian networks particularly suitable for high-stakes decision-making.

In domains such as healthcare, autonomous systems, and financial forecasting, decision-making often involves navigating complex and uncertain environments. In healthcare diagnostics, Bayesian networks serve as powerful tools to model the relationships between symptoms, diseases, and test results, allowing clinicians to leverage probabilistic reasoning in diagnosing conditions and prognosticating outcomes [Polotskaya *et al.* 2024; Kyrimi *et al.* 2020]. Polotskaya *et al.* [2024] provide a comprehensive review of Bayesian network applications in disease diagnosis and prognosis, while Kyrimi *et al.* [2020] highlight the current challenges in their adoption across clinical settings.

In autonomous systems, robots utilize Bayesian and dynamic Bayesian networks to interpret noisy sensor data, predict the outcomes of actions, and make informed decisions in uncertain and dynamic environments [Murphy 2002; Thompson *et al.* 2023]. For instance, Murphy [2002] discusses the foundational role of dynamic Bayesian networks in robotic inference, and Thompson *et al.* [2023] present recent advancements in Bayesian learning for runtime decision verification in autonomous systems.

In financial forecasting, Bayesian network models support risk assessment, market trend prediction, and evaluation of events such as credit defaults or systemic crises [Chong and Klüppelberg 2017; Bayev 2023]. Chong and Klüppelberg [2017] apply Bayesian networks to model financial contagion, and Bayev [2023] highlights their role in managing credit and market risk. These applications collectively demonstrate the versatility and effectiveness of Bayesian networks in addressing the challenges posed by uncertainty across diverse domains.

Unlike black-box machine learning models, which often lack interpretability, Bayesian networks provide transparent and explainable insights into the reasoning process. This transparency fosters trust and reliability, particularly in applications where decisions have significant consequences, such as medical diagnostics or autonomous vehicle control.

While Bayesian networks offer numerous advantages, their application in large-scale and high-dimensional settings is hindered by significant computational challenges. Inference, the process of calculating the posterior probability of one or more variables given observed evidence, is central to the functionality of Bayesian networks. Exact inference involves marginalizing over the joint probability distribution, which requires summing or integrating over all non-query variables. This task grows exponentially in complexity with the number of variables and their dependencies, making exact inference an $\mathcal{NP}$-hard problem [Darwiche 2009; Koller and Friedman 2009].

This computational burden becomes particularly challenging in dynamic environments where networks must adapt to new data and queries in real time. In healthcare, real-time diagnostic systems are required to seamlessly integrate new test results or symptoms, ensuring that critical decisions are not delayed. Similarly, in autonomous robotics, Bayesian networks must rapidly update their predictions as the robot navigates through its environment and encounters new information. These scenarios highlight the necessity for efficient and adaptive inference methods capable of maintaining accuracy under dynamic conditions.

As the size and density of the network increase, inference becomes computationally infeasible without significant optimization. Traditional approaches often involve approximate methods, which trade accuracy for computational efficiency. However, these methods are unsuitable for applications requiring precise and reliable results, such as diagnosing rare diseases or making life-critical decisions in autonomous systems.

To address these challenges, this research focuses on query-specific pruning as a means to reduce

computational overhead while preserving the accuracy and reliability of Bayesian network inference. Query-specific pruning selectively simplifies the network by removing nodes and edges that are irrelevant to the query and observed evidence. By focusing computational resources on the parts of the network directly relevant to the task at hand, this approach achieves significant reductions in inference cost without compromising the quality of the results.

This pruning strategy utilizes the inherent conditional independencies encoded in Bayesian networks, ensuring that the simplified network remains probabilistically equivalent to the original network for the given query. For instance, in a medical diagnostic application, pruning might involve removing variables related to unrelated conditions, focusing the computation on the subset of the network that directly impacts the query.

Beyond reducing computational cost, query-specific pruning aligns with the interpretability and explainability goals of Bayesian networks. By isolating the relevant portions of the network, the pruning process provides decision-makers with a clearer understanding of which variables and relationships are influencing the inference [Derks and de Waal 2021; Shih *et al.* 2018]. This clarity is invaluable in domains where understanding the reasoning behind a decision is as important as the decision itself [MLgeek 2016]. Moreover, query-specific pruning is particularly effective when the evidence provided is sparse (e.g., [**?**Gratius *et al.* 2025]). Unlike traditional methods that attempt to process the entire network regardless of the query, this approach dynamically adapts the network structure to the specific task, ensuring efficient and accurate inference even in high-dimensional settings [Yuan *et al.* 2014].

This research seeks to advance Bayesian network inference by systematically exploring and evaluating query-specific pruning strategies. The primary objectives include the development and implementation of advanced pruning techniques, such as V-structure pruning, graph separator-based pruning, and cluster-based pruning, all tailored to specific queries and evidence [Correia and others 2020; Kojima *et al.* 2010]. Additionally, the study evaluates the impact of these pruning strategies on critical metrics, including inference accuracy, computational cost, and scalability, across a diverse range of synthetic and real-world datasets [**?**]. Finally, the research aims to demonstrate the practical utility of query-specific pruning in real-world applications, such as healthcare diagnostics and autonomous systems, highlighting its potential to address computational inefficiencies in these domains [**?**].

This research addresses a critical gap in the literature by providing a comprehensive framework for pruning Bayesian networks in a query-specific manner. By focusing on the dynamic interaction between query, evidence, and network structure, this work offers a novel approach to enhancing the efficiency and scalability of exact inference in Bayesian networks.

The remainder of this dissertation is organized as follows. Chapter 2 provides the background, introducing foundational concepts and the theoretical framework for Bayesian networks and inference. Chapter 3 reviews related work, highlighting existing methodologies and identifying their limitations. Chapter 4 outlines the proposed methodology, detailing the technical framework for implementing and evaluating query-specific pruning strategies. Chapter 5 presents the experimental results, offering an in-depth analysis of the trade-offs between computational efficiency and accuracy. Chapter 6 discusses the findings in detail, examining why certain pruning methods performed as they did and exploring the implications of these results. Finally, Chapter 7 concludes with a summary of the research contributions and proposes potential directions for future work.

Chapter 2

Background

This chapter provides a foundational understanding of Bayesian networks (BNs) and the methodologies used in their inference processes. It introduces the key concepts and notations that will be consistently used throughout this dissertation, ensuring a structured and cohesive framework for subsequent chapters.

2.1 Bayesian Networks

A BN is a probabilistic graphical model that represents a set of variables and their conditional dependencies through a directed acyclic graph (DAG). Each node in the graph symbolizes a variable, and each directed edge represents a conditional dependency between the variables. Bayesian Networks are particularly useful for modeling complex stochastic processes and have been widely applied in fields ranging from medicine to artificial intelligence [Pearl 1988]. By providing a graphical representation of dependencies, Bayesian Networks facilitate reasoning under uncertainty and offer a robust mechanism for probabilistic inference.

Bayesian Networks consist of nodes that represent random variables, which can be either discrete or continuous, and edges that depict conditional dependencies. These dependencies are quantified through Conditional Probability Distributions (CPDs) associated with each node. The joint probability distribution of the network is derived from the CPDs, capturing the probabilistic relationships among all variables in the system. This capability makes Bayesian Networks an effective tool for both predictive and diagnostic reasoning.

The Bayesian Network illustrated in Figure 2.1 represents the relationships between a student's academic difficulty, intelligence, grade, SAT scores, and recommendation letters. Each node in the network corresponds to a random variable, and the directed edges indicate how one variable influences another. For example, the difficulty of a course and a student's intelligence directly affect their grade, which in turn impacts their SAT scores and the quality of their recommendation letters.

2.2 Inference in Bayesian Networks

Inference in Bayesian Networks involves calculating the posterior probabilities of certain variables given evidence. This process allows the network to make predictions, diagnose causes, and compute probabilities for specific scenarios. Several algorithms are employed to perform inference, including Variable Elimination, Belief Propagation, and Monte Carlo methods [Jensen and Nielsen 2007]. These algorithms utilize the structure of the Bayesian Network to efficiently compute the required probabilities while managing computational complexity.

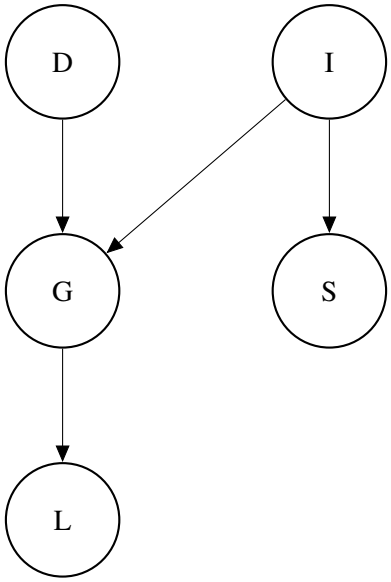


Figure 2.1: Bayesian Network depicting the relationship between Difficulty (D), Intelligence (I), Grade (G), SAT (S), and Letter (L).

Variable Elimination is a prominent method used for exact inference. It systematically eliminates variables from the network by marginalizing over their possible values, thereby simplifying the computation of marginal distributions. The order in which variables are eliminated plays a crucial role in determining the computational cost of the algorithm. An optimal elimination order minimizes the size of intermediate factors, reducing the overall computational burden [Neapolitan 2004].

Probabilistic inferences in Bayesian Networks often include marginalization and conditional queries. Marginalization determines the probability distribution of a subset of variables, integrating over others. Conditional queries compute the probability of a hypothesis given observed evidence, making Bayesian Networks highly applicable for decision-making under uncertainty.

2.3 Importance of Efficient Inference

The growing complexity of systems modeled by Bayesian Networks has intensified the need for efficient inference techniques. As networks increase in size and density, the computational cost associated with exact inference escalates exponentially, often making it impractical for large-scale or real-time applications [Korb and Nicholson 2010a]. These challenges necessitate the development of advanced methodologies to optimize computational efficiency without compromising accuracy.

2.4 Conclusion

The background provided here lays the groundwork for understanding the complexities associated with Bayesian Networks and the importance of efficient inference. The subsequent chapters will delve deeper into specific pruning techniques and their practical applications in various domains.

Chapter 3

Related Work

This section reviews the existing literature on Bayesian networks, with a focus on exact inference methods and graph pruning techniques. The discussion highlights key studies that have addressed the computational challenges inherent to Bayesian network inference, particularly in high-dimensional settings. By identifying the limitations of existing methods and outlining current gaps, this review provides the necessary context and rationale for our research on query-specific pruning strategies.

Bayesian networks are indispensable in fields such as artificial intelligence, healthcare, and machine learning, where they are used for tasks such as prediction, anomaly detection, and decision-making under uncertainty. However, inference in these networks remains a computationally challenging problem due to the $\mathcal{NP}$-hard nature of exact inference algorithms. As the size and connectivity of the network increase, the computational complexity of inference grows exponentially, making exact inference infeasible for large-scale or real-time applications [Koller and Friedman 2009; Darwiche 2009].

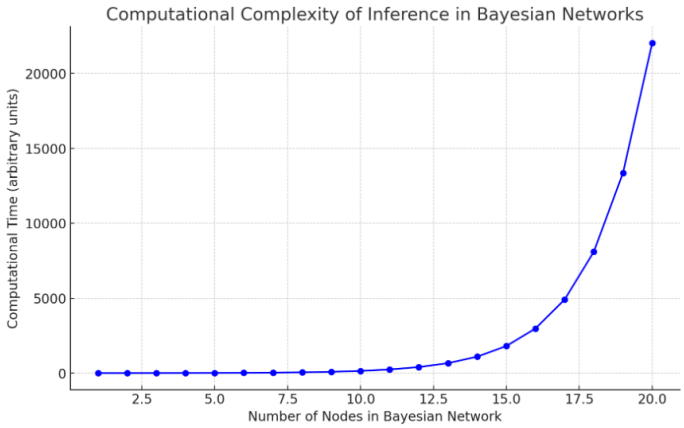


Figure 3.1: Inference in Bayesian networks is an $\mathcal{NP}$-hard problem. As the number of nodes increases, the time required for exact inference grows exponentially.

Research indicates that reducing the size and complexity of Bayesian networks significantly enhances the efficiency of inference [Chavira and Darwiche 2008; Korb and Nicholson 2010b]. Smaller networks have fewer parameters and simpler structures, which reduces the number of calculations needed to compute marginal probabilities or posterior distributions. Additionally, smaller networks often lead to fewer dependencies among variables, further streamlining the inference process and making real-time applications more feasible.

This section delves into two primary themes in the literature that address the challenges associated with Bayesian network inference. The first theme focuses on exact inference in Bayesian networks, providing a detailed review of the principal methods used to perform exact inference, their computational bottlenecks, and the limitations they encounter in high-dimensional settings. The second theme examines graph pruning and partitioning techniques, which aim to simplify Bayesian networks by removing or segmenting irrelevant components. This discussion highlights the importance of these techniques, explores existing methodologies, and identifies unresolved issues that warrant further investigation.

The subsequent discussion will delve into these themes in detail, examining foundational theories and recent advancements in each domain to establish the basis for the query-specific pruning strategies.

3.1 Exact Inference in Bayesian Networks

Exact inference refers to the precise computation of posterior probabilities for a query variable, given the evidence observed. Unlike approximate methods, which trade accuracy for computational feasibility, exact inference guarantees that the computed probabilities align perfectly with the joint distribution encoded by the network. This precision is critical for applications where even small deviations in probabilities can lead to substantial risks or errors [Koller and Friedman 2009].

3.1.1 The Computational Complexity of Exact Inference

Despite its precision, exact inference is inherently computationally intensive [Conrad *et al.* 2016]. It involves marginalizing over the joint distribution by summing or integrating across non-query variables—a process that becomes increasingly complex as the network's size and connectivity grow [Darwiche 2009]. The computational burden is exacerbated by dense dependencies and high-dimensional spaces, where the number of operations required grows exponentially with the number of variables and their dependencies [Dagum and Luby 1993]. These challenges underline the importance of developing innovative approaches to manage computational complexity without sacrificing the accuracy that exact inference guarantees.

To illustrate this, consider a Bayesian network with n nodes, where each node can have up to k parents. The size of the largest intermediate factor during inference can reach 2^k, and the number of computations required grows proportionally. This is particularly problematic in densely connected networks or networks with high tree-width, where large cliques are formed during variable elimination or the construction of a junction tree [Korb and Nicholson 2010b].

3.1.2 Why Exact Inference Remains Relevant

Despite its computational challenges, exact inference remains a critical area of research due to its ability to provide precise and reliable results. Unlike approximate methods, such as sampling or variational inference, exact inference guarantees correctness by fully respecting the probabilistic relationships encoded in the network [Korb and Nicholson 2010b].

This is particularly critical in applications where errors in inference could have severe and far-reaching consequences. In healthcare diagnostics, for instance, inaccuracies in probabilistic reasoning could lead to misdiagnoses, resulting in incorrect treatments that jeopardize patient safety. Similarly, in autonomous systems, unreliable inference can cause robots navigating uncertain environments to make unsafe decisions, potentially endangering both the robots and their surroundings. In financial systems, errors in risk prediction could lead to significant financial losses or even systemic risks, highlighting the need for precise and dependable inference methodologies in these high-stakes domains.

For example, in healthcare, an exact inference model could combine symptoms, test results, and prior probabilities to calculate the posterior likelihood of a rare disease. Approximations in this scenario could result in overlooking critical, low-probability conditions that exact inference would capture. Given its importance, researchers have explored numerous strategies to optimize exact inference in Bayesian networks.

These strategies include selecting efficient elimination orders, which aim to minimize the sizes of intermediate factors during variable elimination, thereby improving computational efficiency [Dechter 1999; Sümer *et al.* 2011; Peyrard *et al.* 2019]. Another critical approach is the exploitation of network structure, particularly by utilizing sparsity to significantly reduce the computational demands associated with inference tasks [Yu *et al.* 2020; Bianconi 2022; Karargyris *et al.* 2020]. Additionally, hybrid approaches have gained prominence by combining exact inference with approximate methods in targeted parts of the network. This balance between accuracy and efficiency ensures that the computational burden is reduced without compromising the reliability of probabilistic reasoning in complex systems [Neil *et al.* 2007; Shenoy 2012; Lerner and Parr 2013].

3.1.3 Efficient Elimination Orders

With regards to exact inference in Bayesian networks, the sequence of variable elimination is critical in determining computational efficiency [Ren *et al.* 2022]. An optimal elimination order minimizes the size of intermediate factors generated during inference, directly impacting both the time and memory requirements. Several studies have explored strategies to optimize variable elimination, utilizing heuristics, dynamic adjustments, and graph-based approaches to address these computational challenges [Dechter 1999; Sümer *et al.* 2011; Peyrard *et al.* 2019].

The Min-Fill heuristic, introduced within the bucket elimination framework by Dechter [1999], is a foundational method for determining efficient elimination orders. This heuristic aims to minimize the formation of large cliques during variable elimination by selecting the variable whose elimination introduces the fewest additional edges into the graph. In doing so, the Min-Fill heuristic reduces the size of intermediate factors, effectively lowering the computational burden of inference. This approach is particularly effective in networks with low to moderate connectivity, where minimizing the creation of unnecessary dependencies significantly enhances efficiency.

For example, consider the Bayesian network in Figure 3.2, which illustrates a graph with nodes A, B, C, D, E, F, G, and H. Eliminating node A, which is connected only to B and C, does not introduce any additional edges, thereby preserving the sparsity of the graph. This demonstrates the heuristic's ability to reduce computational complexity by keeping intermediate factors as small as possible.

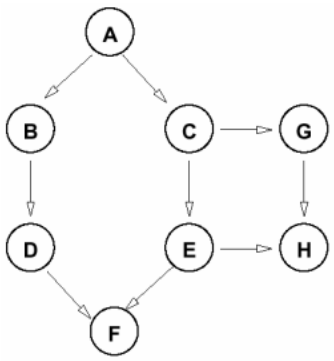


Figure 3.2: Bayesian Network with Nodes A, B, C, D, E, F, G, H.

The Min-Fill heuristic forms the basis for numerous subsequent developments in variable elimination, including its integration into adaptive and dynamic methods, as explored in later works.

Building upon the static nature of heuristics like Min-Fill, Sümer *et al.* [2011] proposed an adaptive exact inference framework that dynamically adjusts the elimination order based on the evolving structure of the network. Unlike static approaches, which determine the elimination sequence upfront, this method recalculates the order after each variable elimination step. By continuously considering the current state of the graph, adaptive inference ensures that the computational complexity remains optimized throughout the process.

Figure 3.3 highlights the impact of elimination ordering on computational efficiency. In the first panel (a), a chain graph with sequential dependencies demonstrates high computational cost due to its unoptimized structure. Panel (b) shows an unbalanced elimination tree, where hierarchical grouping remains inefficient. Finally, panel (c) illustrates a balanced elimination tree that optimizes computational efficiency by minimizing tree height and intermediate factor sizes. This adaptive approach is particularly beneficial in networks with dynamic or evolving structures, where static elimination orders may become suboptimal.

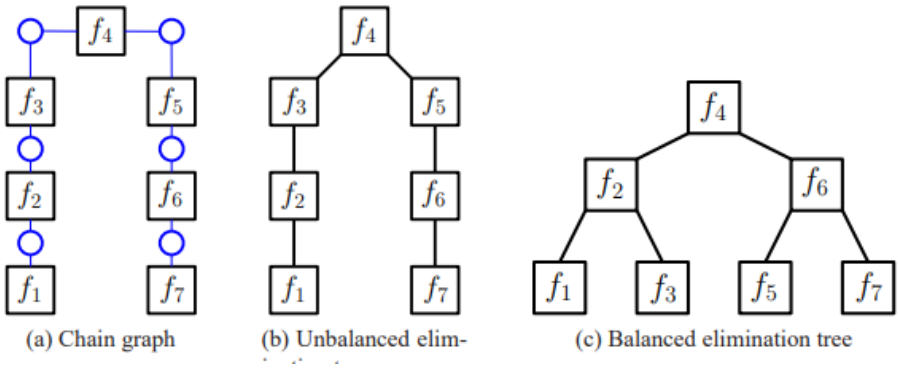


Figure 3.3: Comparison of elimination structures: (a) A chain graph showing sequential dependencies with high computational cost, (b) an unbalanced elimination tree illustrating inefficient hierarchical grouping, and (c) a balanced elimination tree optimizing computational efficiency by minimizing tree height and intermediate factor sizes. Sümer *et al.* [2011]

The dynamic nature of this method draws parallels with tree decomposition techniques, discussed below, which also aim to reduce computational complexity by restructuring the graph.

Treewidth plays a central role in the computational complexity of exact inference. As highlighted by Peyrard *et al.* [2019], the treewidth of a Bayesian network dictates the size of intermediate factors during variable elimination. Their work demonstrated how tree decomposition techniques, combined with heuristics like Min-Fill, can reduce computational overhead by organizing the graph into clusters with bounded treewidth.

Figure 3.4 illustrates this concept with a graphical model (left) and its corresponding tree decomposition (right). The decomposition groups variables into clusters (C_1 to C_5) such that each cluster maintains the dependencies of the original graph while limiting the computational complexity of inference. This restructuring is particularly valuable in high-dimensional networks, where unbounded treewidth can lead to prohibitive computational costs.

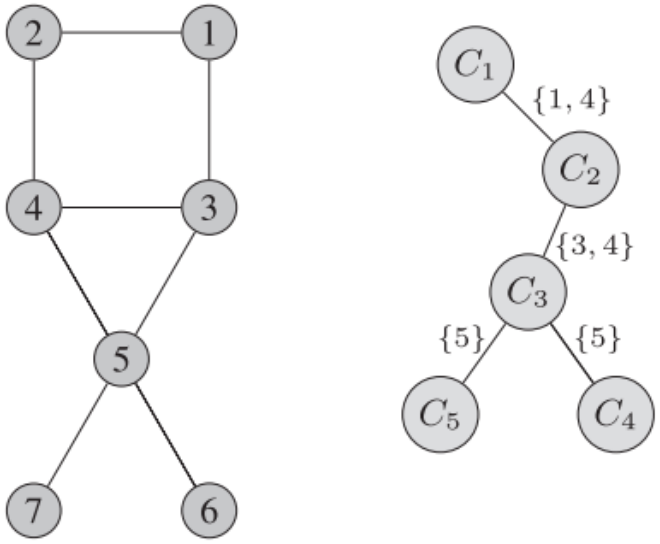


Figure 3.4: Left: graphical representation of a graphical model. Right: tree decomposition over clusters Peyrard *et al.* [2019]

The tree decomposition approach shares similarities with adaptive inference in its focus on restructuring the graph for computational efficiency. However, it also emphasizes the importance of treewidth as a formal metric for evaluating and optimizing network complexity.

Lifted Probabilistic Inference: Complementing the focus on treewidth and dynamic ordering, Kazemi and Poole [2014] extended the principles of variable elimination to lifted probabilistic models. By combining the Min-Fill heuristic with symmetry detection, their method identifies structural redundancies in the network and exploits these redundancies to optimize elimination orders. This approach has proven particularly effective in relational models, where repetitive patterns introduce opportunities for computational savings.

Lifted inference builds on the principles of sparsity and conditional independence, similar to tree decomposition and adaptive methods. However, its emphasis on exploiting structural symmetry makes it uniquely suited to networks with repetitive or hierarchical patterns.

Why These Approaches Were Taken

The chosen approaches are motivated by specific objectives aimed at enhancing the efficiency of inference in Bayesian Networks. One primary consideration is the need to limit treewidth. The size of intermediate factors generated during inference is directly proportional to the treewidth of the graph. Consequently, methods like Min-Fill explicitly aim to minimize treewidth during variable elimination [Dechter 1999]. This strategy ensures that the computational complexity associated with inference remains manageable, particularly in scenarios involving large networks.

Another critical factor is the exploitation of graph sparsity. Many real-world Bayesian networks are inherently sparse, characterized by limited connectivity between variables. Optimization techniques that capitalize on this sparsity can significantly reduce computational overhead while preserving accuracy [Peyrard *et al.* 2019]. By focusing on sparse structures, these methods align with the natural properties of many networks, allowing for efficient inference without introducing unnecessary complexity.

Additionally, dynamic adaptation is a key feature of certain methods. Dynamic approaches ensure that the elimination order remains optimal as evidence is introduced, which often alters the graph's structure. By adapting dynamically, these methods maintain efficiency and avoid the pitfalls of static elimination orders, particularly in scenarios where the graph undergoes frequent changes [Sümer *et al.* 2011]. This adaptability is essential for ensuring robust performance across a wide range of applications.

Drawbacks with These Approaches

Despite their advantages, these approaches come with notable limitations that affect their applicability in certain contexts. One significant drawback is their suboptimal performance in dense networks. While the Min-Fill heuristic is effective in sparse networks, its efficiency diminishes in dense networks where large cliques are unavoidable. This limitation stems from the inherent difficulty of minimizing treewidth in densely connected graphs, which often results in computationally expensive intermediate factors [Dechter 1999].

Dynamic methods, while offering adaptability, introduce substantial computational overhead. The continuous updates required to maintain an optimal elimination order can be resource-intensive, particularly in large networks. This overhead limits the practicality of adaptive inference methods in scenarios involving extensive networks with frequent updates [Sümer *et al.* 2011].

Another challenge lies in the complexity of tree decomposition. Decomposing a graph to minimize treewidth is computationally intensive, especially as the size of the network increases. This complexity constrains the scalability of these methods, making them less suitable for extremely large networks [Peyrard *et al.* 2019].

Finally, lifted inference methods, which rely on exploiting symmetries within the network, have limited applicability. In scenarios where such symmetries are absent, these methods lose their efficiency and fail to offer any significant computational advantages [Kazemi and Poole 2014]. This reliance on specific structural properties of the network restricts the generalizability of lifted methods, limiting their use in more diverse or irregular networks.

Efficient elimination order strategies are indispensable for managing the computational demands of exact inference. However, their limitations, especially in high-treewidth or dense networks, emphasize the need for alternative methods. This research complements existing strategies by introducing query-specific pruning techniques, which reduce the network's effective size and complexity, thereby alleviating the challenges associated with elimination order optimization.

3.1.4 Utilizing Sparsity in Bayesian Networks for Scalable Inference

Sparsity in Bayesian networks, characterized by the presence of conditional independencies and a limited number of direct dependencies, has emerged as a cornerstone for reducing computational complexity in exact inference tasks. This property allows researchers to focus on essential connections within the network, enabling faster, more scalable inference while preserving model accuracy [Yu *et al.* 2020; Bianconi 2022; Karargyris *et al.* 2020]. This section explores key advancements and applications of sparsity in Bayesian networks, emphasizing its role in addressing computational challenges across diverse domains.

One of the foundational approaches to utilizing sparsity involves automatic sparsity determination during network construction. Yu *et al.* [2020] introduced a fast Bayesian inference framework that incorporates sparsity regularization into the learning process. By penalizing unnecessary dependencies through Bayesian regularization techniques, this method iteratively refines the network structure to focus on essential connections.

The resulting models not only minimize computational overhead but also retain their ability to capture complex probabilistic relationships. For instance, applications in gene expression analysis and brain connectivity studies—where natural sparsity is prevalent—demonstrate the method's efficiency and interpretability [Friedman *et al.* 2000; Xing *et al.* 2017; Bielza and Larrañaga 2014]. This approach underscores how enforcing sparsity during model learning can balance computational efficiency and representational fidelity, providing practical solutions for high-dimensional data analysis.

Building on the idea of sparsity, Bianconi [2022] proposed a grand canonical ensemble framework to model sparse networks probabilistically. This framework employs hierarchical Bayesian methods to prioritize meaningful dependencies while discarding weak or redundant connections. By utilizing sparsity, the ensemble approach accurately captures the structural constraints of real-world networks.

A schematic representation of the hierarchical process is shown in Figure 3.5, illustrating how sparse networks are generated. Starting with a root node for the total number of nodes N, the framework samples the degree sequence $k = \{k_1, k_2, \ldots, k_N\}$ from a conditional distribution $P(k|N)$, ensuring real-world sparsity in the degree distribution. Finally, the adjacency matrix a is sampled from $P(G|k, N)$, encapsulating all possible networks consistent with the degree sequence and total node count.

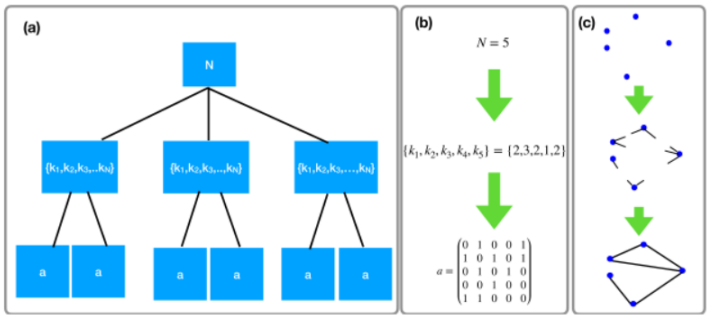


Figure 3.5: Schematic representation of the hierarchical grand canonical ensemble of sparse networks [Bianconi 2022].

This method's scalability and efficiency make it particularly effective for applications like social network analysis and infrastructure modeling. By utilizing sparse representations, the grand canonical ensemble ensures that network complexity is reduced without compromising the ability to model significant dependencies.

In the area of medical imaging, sparsity in Bayesian networks has proven instrumental for efficiently handling high-dimensional datasets. Karargyris *et al.* [2020] demonstrated a method that strategically utilizes sparsity to localize inference computations in critical regions, such as tumor boundaries in CT or MRI scans. This selective focus on essential regions reduces computational demands, enabling real-time analysis while maintaining high accuracy.

The ability to focus inference on sparse, clinically relevant areas of radiological images provides significant benefits in time-sensitive scenarios. This balance between computational efficiency and precision underscores the transformative potential of sparsity in medical decision-making, where delays in processing can have critical consequences.

The advancements discussed above highlight the transformative impact of sparsity on Bayesian network design and inference. Automatic sparsity determination, ensemble methods, and domain-specific applications demonstrate the versatility of this guiding principle. These approaches share a common objective: reducing the number of active dependencies in the network while preserving its ability to model complex systems.

By utilizing sparsity, researchers have not only enhanced the scalability and efficiency of Bayesian networks but also broadened their applicability across domains ranging from biological sciences to infrastructure modeling and healthcare. This paradigm shift has established sparsity as a robust framework for overcoming computational challenges in exact inference, laying the groundwork for future innovations.

Challenges in Sparsity Exploitation

While sparsity-based methods offer substantial computational advantages in Bayesian Networks, they are not without challenges. One significant issue is the preprocessing overhead associated with identifying and enforcing sparsity. The process of determining which connections are essential and which can be pruned often demands considerable computational resources during the model initialization phase. This initial investment in computational effort can offset some of the advantages gained during the inference process, particularly in large and complex networks [Yu *et al.* 2020].

Another critical limitation is the reduced applicability of sparsity-based approaches in dense networks. In scenarios where network dependencies are inherently dense, the benefits of exploiting sparsity diminish significantly. Sparse methods, which rely on the existence of limited dependencies, may even introduce unnecessary complexity when applied to dense networks. In such cases, the overhead of enforcing sparsity may outweigh its computational advantages, rendering these methods less effective [Bianconi 2022].

Additionally, dynamic adaptation poses a challenge for sparsity-based techniques. Many real-world networks are dynamic in nature, with edges and nodes evolving over time. Sparse methods often struggle to maintain their efficiency in such settings, as they require continuous updates to the network structure to reflect these changes. This need for frequent recalibration can lead to significant computational overhead, limiting the practicality of sparsity-based methods in applications where the network is subject to constant evolution [Karargyris *et al.* 2020].

In summary, while sparsity-based methods are powerful tools for optimizing computational efficiency in Bayesian Networks, they are best suited for specific contexts where the network structure aligns with their assumptions. Their challenges, particularly in preprocessing overhead, dense networks, and dynamic environments, highlight the need for complementary strategies to address these limitations.

This research builds on the principles of sparsity exploitation by introducing query-specific pruning techniques. Unlike static sparsity-exploitation methods, this approach dynamically adjusts the network structure based on the evidence and query, effectively increasing sparsity. This ensures that computational resources are focused on the most relevant parts of the network, reducing inference costs while maintaining high accuracy.

3.1.5 Hybrid Approaches: Combining Exact and Approximate Methods

Hybrid inference approaches in Bayesian networks represent a critical innovation aimed at balancing computational efficiency and accuracy. By selectively applying exact inference to computationally manageable regions of the network and employing approximate methods in more complex areas, these approaches provide a scalable solution to the challenges posed by large or densely connected networks. This flexibility ensures that precision is maintained in critical regions, while computational costs are minimized in areas where exact inference is impractical [Neil *et al.* 2007; Shenoy 2012; Lerner and Parr 2013].

Neil *et al.* [2007] introduced a dynamic discretization framework as a hybrid solution for inference in Bayesian networks containing both discrete and continuous variables. This method dynamically

discretizes continuous variables during inference, creating an adaptable representation that allows for exact computations on the discretized form. The adaptive refinement of the discretization, based on the specific evidence and queries, ensures an effective balance between computational cost and accuracy. Dynamic discretization has proven particularly valuable in fields such as risk analysis and reliability modeling, where hybrid networks often represent systems with mixed-variable interactions.

Building on the idea of utilizing approximations for continuous variables, Shenoy [2012] proposed a method using mixtures of Gaussians to represent continuous distributions in hybrid Bayesian networks. By approximating continuous variables as a weighted combination of Gaussian components, this method facilitates the application of exact inference algorithms tailored for Gaussian mixtures. The approach is computationally efficient while preserving a high degree of accuracy, making it ideal for domains like sensor fusion and robotics, where mixed-variable dependencies are prevalent and timely decision-making is critical.

Expanding on the challenges and solutions in hybrid inference, Lerner and Parr [2013] conducted a detailed investigation into the theoretical limits and practical implementations of hybrid methods in Bayesian networks. They identified key challenges, such as the potential for error propagation when transitioning between exact and approximate methods. To mitigate these issues, the authors introduced partitioning algorithms that divide the network into regions optimized for either exact or approximate inference. This strategic partitioning minimizes error propagation and enhances computational efficiency. The methodology has been successfully applied in high-stakes applications such as medical diagnostics, where precise probabilistic reasoning is essential, and environmental modeling, which often involves high-dimensional hybrid networks [Cooper *et al.* 2012; Marcot and Penman 2019].

Together, these studies highlight the promise of hybrid inference methods in addressing the scalability challenges inherent in Bayesian networks while maintaining accuracy in critical areas. By dynamically adapting inference strategies to the network's structure and computational constraints, these methods offer a versatile solution for complex, real-world problems.

Challenges in Hybrid Approaches

Hybrid approaches, which combine exact and approximate inference methods, offer a flexible framework for managing large and complex Bayesian networks. However, these methods face several significant challenges. One critical issue is error propagation. When exact and approximate methods are integrated, errors from the approximate components can accumulate over multiple iterations of the inference process. This is particularly problematic in dense networks with high interdependencies, where the cumulative effect of these errors can compromise the overall accuracy of the results [Lerner and Parr 2013].

Another challenge lies in dynamic adaptation. Hybrid methods, such as dynamic discretization, often require iterative refinement to maintain accuracy during inference. For networks with high-dimensional continuous variables, this iterative process can become computationally expensive, potentially negating the efficiency gains offered by hybrid approaches. The need for constant adjustments to the discretization scheme places a significant computational burden on these methods, particularly when applied to large-scale or real-time systems [Neil *et al.* 2007].

The representation of models in hybrid approaches also presents a challenge. For example, approximating continuous probability distributions using techniques like mixtures of Gaussians can lead to oversimplifications. In regions of the network where precise modeling is crucial, such approximations may fail to capture important subtleties, resulting in a loss of fidelity. This limitation is particularly pronounced in applications requiring high accuracy, such as medical diagnosis or risk assessment, where small deviations in probability estimates can have significant consequences [Shenoy 2012].

While exact inference guarantees precision, its computational demands grow exponentially with the size and connectivity of the Bayesian network, as highlighted in the previous section. These challenges necessitate innovative approaches to manage the complexity of large-scale networks, especially in high-dimensional applications. One promising avenue involves restructuring the network through graph pruning and partitioning techniques.

3.2 Pruning Strategies in Bayesian Networks

Pruning Bayesian networks is a crucial approach to reduce the computational overhead of exact inference. Baker and Boult [2013] introduced a framework for constructing minimal computationally equivalent subgraphs tailored to specific queries and evidence. Their method systematically eliminates irrelevant nodes and edges by utilizing the conditional independence relationships encoded in the network's directed acyclic graph (DAG).

The primary goal of Baker and Boult [2013]'s framework is to decompose a Bayesian network into smaller, disjoint subgraphs that are computationally equivalent to the original network for a given query. This is achieved through a two-step methodology. First, d-separation pruning is applied, where nodes and edges that are conditionally independent of the query and evidence are identified and removed based on d-separation principles. Figure 3.6 illustrates these principles.

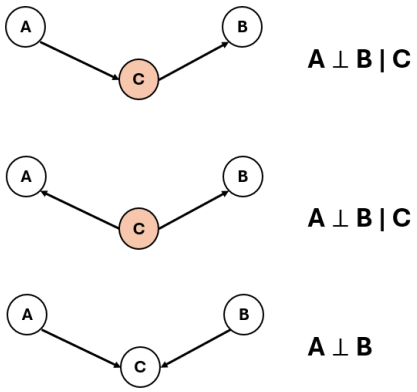


Figure 3.6: d-separation principles in a Bayesian Network [Pearl 1988]

This reduces the network to only those components directly relevant to the query and evidence. Second, recursive pruning of barren nodes is conducted to eliminate nodes that do not contribute to the query, such as barren leaf nodes with no relevant descendants. This step further simplifies the network while ensuring that the pruned structure retains computational equivalence to the full network. Together, these steps produce minimal subgraphs tailored to the query, enabling efficient computation while maintaining accuracy.

Figure 3.7 illustrates the pruning process, starting with the full Bayesian network, applying d-separation pruning to remove irrelevant components, and finally recursively pruning barren nodes. The result is a set of minimal subgraphs tailored to the query and evidence.

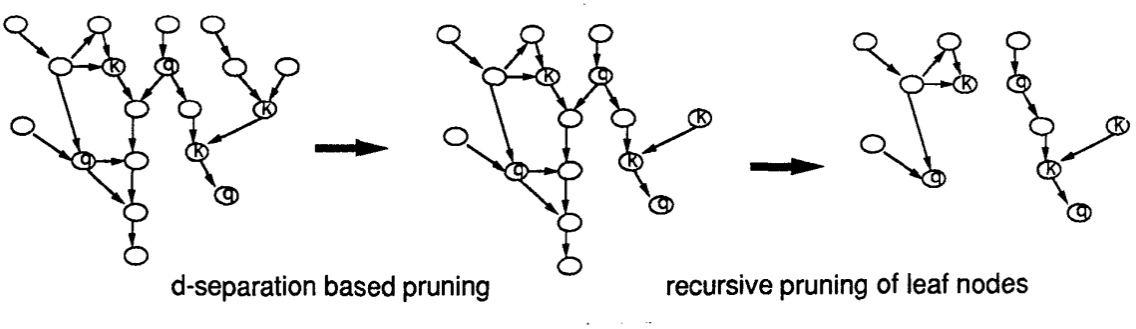


Figure 3.7: Illustration of Baker and Boult [2013]'s pruning methodology. The process begins with d-separation pruning to remove conditionally independent components, followed by recursive pruning of barren nodes to produce minimal subgraphs. [Baker and Boult 2013]

Despite its effectiveness, Baker and Boult [2013]'s framework has several limitations that restrict its broader applicability. The method is particularly suited to sparse networks, where the presence of numerous conditional independencies allows for significant pruning. However, in densely connected networks, the number of nodes and edges deemed relevant remains high, which limits the potential computational gains. Additionally, the framework is designed for static queries and evidence, making it less adaptable to dynamic scenarios where the query or evidence frequently changes, as in real-time or evolving systems. Furthermore, the process of identifying irrelevant nodes and edges requires substantial preprocessing. While this preprocessing effort can lead to computational savings during inference, it may offset those benefits in applications that demand immediate results, thus posing a challenge for time-sensitive tasks. These limitations underscore the need for more dynamic and adaptable pruning methods to address the challenges posed by complex, high-dimensional Bayesian networks.

Building on the idea of simplifying Bayesian networks, Xiangyuan *et al.* [2021] proposed a pruning strategy specifically designed to enhance the efficiency of structure learning in Bayesian networks. Their method introduces causal constraints that prune irrelevant parent sets during the network learning process, significantly reducing the search space. By focusing on causally plausible parent-child relationships, this approach ensures that only the most relevant dependencies are retained, resulting in a substantial reduction in computational costs.

Unlike Baker and Boult [2013]'s method, which focuses on inference, this work targets the structure learning phase of Bayesian network modeling. It demonstrates how causal constraints can be used to guide the pruning process, leading to up to a 60% reduction in runtime while maintaining high accuracy in structure learning. However, this approach is limited to structure learning tasks and does not directly address inference challenges, particularly in dynamic or high-dimensional environments.

Zhang and Poole [1994] extended the concept of pruning by introducing techniques to construct minimal subnetworks for decision-making tasks. This work focused on efficiently pruning Bayesian networks to retain only the variables and dependencies directly relevant to specific decision criteria. Unlike the work of Baker and Boult [2013], which generates subgraphs computationally equivalent to the original network for inference, Zhang and Poole [1994]'s approach emphasizes decision relevance, which may lead to suboptimal accuracy in probabilistic queries. Their pruning method uses conditional independence tests to identify and remove irrelevant components, similar to Baker and Boult [2013]'s d-separation principles. However, Zhang and Poole [1994]'s focus on decision-specific pruning limits its applicability to general probabilistic inference tasks, as the pruned network may not accurately reflect the posterior distributions for arbitrary queries. Despite this limitation, their work highlights the potential

for pruning strategies to optimize decision-making in resource-constrained environments.

These foundational works establish a robust basis for the development of efficient pruning strategies in Bayesian networks. However, existing methods face several notable limitations that restrict their applicability and effectiveness in more dynamic and complex scenarios.

One significant limitation is the lack of dynamic adaptability. The methods proposed by Baker and Boult [2013] and Zhang and Poole [1994] primarily focus on static queries and evidence, assuming that the structure of the network and the available evidence remain unchanged throughout the inference process. This assumption limits their suitability for dynamic scenarios where queries or evidence may evolve over time. In such cases, the lack of adaptability results in suboptimal performance, as the pruning strategies cannot respond effectively to changing conditions or requirements.

Another challenge is the narrow applicability of these approaches. While Xiangyuan *et al.* [2021] provides valuable insights into structure learning and Zhang and Poole [1994] focuses on decision-making, neither method fully addresses the broader impact of pruning on dynamic, query-specific inference tasks. This narrow focus leaves a gap in their utility for real-world applications, where the ability to adapt pruning strategies to specific queries and dynamic conditions is crucial for ensuring efficiency and accuracy.

A further limitation lies in the restricted metrics used for evaluation. Existing methods primarily emphasize computational savings, often neglecting comprehensive assessments of inference accuracy, the percentage of nodes pruned, or the influence of evidence density on the pruning process. Without these additional metrics, it becomes challenging to fully understand the trade-offs involved or to evaluate the effectiveness of pruning strategies across diverse scenarios.

In summary, while foundational pruning methods have significantly contributed to the field of Bayesian networks, their limitations—such as a lack of dynamic adaptability, narrow applicability, and limited evaluation metrics—highlight the need for more versatile and comprehensive approaches. Addressing these shortcomings is essential to develop pruning strategies that are not only efficient but also robust and adaptable to a wide range of real-world applications.

By drawing inspiration from these works and addressing their limitations, our research introduces advanced query-specific pruning techniques designed for dynamic, real-time scenarios. By evaluating multiple metrics, including computational cost, inference accuracy, and node pruning percentages, our approach aims to extend the applicability of pruning strategies to a broader range of tasks and domains.

3.3 Conclusion

The body of literature on exact inference and graph pruning in Bayesian networks underscores the significant progress made in addressing the computational challenges associated with these models. Exact inference, methodologies such as variable elimination, efficient elimination orders, and hybrid inference techniques have been instrumental in optimizing computational efficiency while retaining accuracy. These approaches have laid a strong foundation, but challenges remain, particularly in high-dimensional and densely connected networks where computational costs grow exponentially.

Graph pruning and partitioning strategies have emerged as powerful techniques to simplify Bayesian networks for query-specific computations. Work such as Baker and Boult [2013]'s minimal subgraph framework demonstrates how utilizing conditional independence and recursive pruning can effectively reduce computational overhead. However, the static nature of their approach, along with its reliance on network sparsity, limits its applicability in dynamic and densely connected networks. Similarly, other pruning and sparsity-driven techniques have achieved notable reductions in computational complexity,

but many fail to adapt dynamically to evolving queries or varying evidence densities, leaving room for improvement in real-time applications.

This review highlights the inadequacies in existing methods, particularly their inability to address queries in high dimensional/densely connected domains, varying levels of evidence, and real-time computational demands effectively. Furthermore, many approaches do not sufficiently evaluate trade-offs between inference cost and accuracy, leaving practitioners without clear guidelines for optimizing Bayesian network performance under different conditions.

The limitations in the current literature, such as static frameworks and inadequate evaluation metrics, point to the need for more flexible and comprehensive approaches. By addressing these gaps, our research aims to advance the state of the art in Bayesian network inference, offering practical solutions that are robust to varying scenarios and capable of meeting the demands of real-world applications.

Chapter 4

Methodology

This section outlines the research design and methods that were employed to investigate the study of enhancing exact inference in Bayesian networks. The approach involves a combination of experimental analysis, algorithm development, and empirical evaluation. We first start off by defining the problem statement in subsection 4.1. The subsequent subsections provide the details of the experimental setup and methodology that were employed to solve the problem statement.

4.1 Problem Statement

Exact inference in probabilistic graphical models (PGMs) is key for applications in machine learning and artificial intelligence, involving tasks such as computing conditional probabilities $P(X \mid E = e)$, marginal probabilities $P(X)$, and expectations $E[X \mid E = e]$. Here, X represents the set of variables in the model, E represents the evidence variables, and e is the observed value of E. These tasks are necessary for prediction, decision-making, and anomaly detection in uncertain environments. The primary difficulty in exact inference stems from the exponential growth in computational complexity as a function of the number of variables n in the joint distribution $P(X_1, \ldots, X_n)$.

This complexity can be expressed as $O(2^n)$ [Figure 3.1], making direct computation impractical for models with even a moderate number of variables, thereby classifying this problem as $\mathcal{NP}$-hard [Koller and Friedman 2009]. For example if we can calculate a query in a network of 2 nodes to be 10 milliseconds then to calculate a query in a network with 30 nodes will take around 1 month to process. This exponential scaling poses a significant barrier in larger networks, leading to a computational requirement explosion that cannot be efficiently managed even with substantial computational resources. The main challenge in exact inference, therefore, is to develop methods that can significantly reduce the computational load without compromising the accuracy of the results.

Our proposed solution directly tackles the computational and scalability challenges inherent in traditional Bayesian Network (BN) inference methods, particularly in scenarios where increasing numbers of variables amplify computational costs. We present a two-fold strategy: first, we apply query-specific pruning techniques to dynamically reduce the size and complexity of the network by retaining only the relevant subgraph. These pruning methods include the following techniques : basic, dynamic, v-structure and cluster pruning; second, we do variable elimination on the subgraph and compute the inference cost and accuracy when compared to the full unpruned network. This approach, as illustrated in Figure 4.1, demonstrates how a complex Bayesian Network can be systematically pruned into a simpler, query-relevant subgraph, enabling efficient inference while maintaining accuracy.

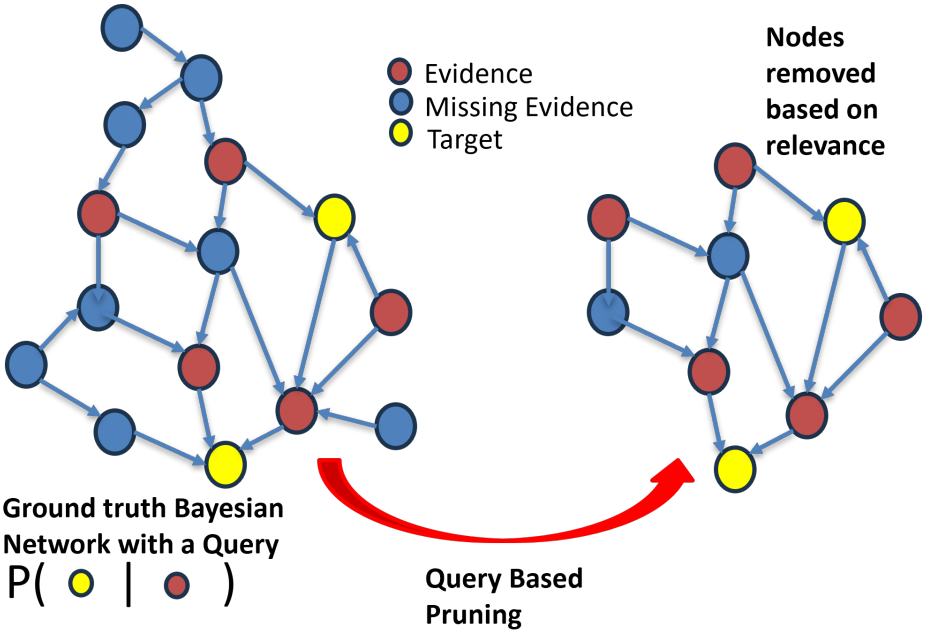


Figure 4.1: Visualization of the methodology, illustrating the transition from an Initial Complex Network to a Pruned network based on a specific query

4.2 Experimental Setup

This section outlines the experimental setup used to evaluate the proposed query-specific pruning strategies for enhancing exact inference in Bayesian networks. We pose three research questions and the experimental methodologies designed to address them.

Research Question 1: How does query-specific pruning impact computational efficiency and accuracy in high-dimensional Bayesian networks?

To evaluate this question, experiments were conducted on various ground truth Bayesian networks ($\mathcal{M}^*$), ranging from sparse networks (less than 10 nodes and 20% edge density) to highly connected networks (greater than 100 nodes and more than 50% edge density). The networks were pruned based on randomly generated probabilistic queries, where partial nodes and evidence are specified. Pruning techniques included basic pruning, dynamic pruning, V-structure pruning, graph separators, and cluster/clique-based pruning (explained in Section 4.4).

Inference was performed on both the pruned networks ($\mathcal{P}$) and the original ground truth networks ($\mathcal{M}^*$) using the variable elimination algorithm. The computational cost (cost to perform inference), the accuracy (comparable to the unpruned network), and the number of pruned nodes were recorded. Ten trials were conducted for each configuration, and the results were plotted with averages and standard deviations to assess the reliability of the results. The aim is to demonstrate that query-specific pruning significantly reduces inference costs without compromising accuracy.

Research Question 2: How does varying the density of evidence affect the performance of query-specific pruning strategies in Bayesian networks?

This question evaluates the impact of evidence density on pruning effectiveness. For each ground truth network ($\mathcal{M}^*$), the number of evidence variables ($\mathcal{E}$) was systematically varied, ranging from sparse

evidence (10% of nodes as evidence) to dense evidence (80% of nodes as evidence). For each query, the network was pruned using the methods described earlier, and the inference was performed on both the pruned networks ($\mathcal{P}$) and the original networks ($\mathcal{M}^*$).

The performance was evaluated by measuring the computational cost, accuracy, and the number of nodes pruned under different evidence densities. Ten trials were conducted for each configuration, and the results were analyzed to determine how the amount of evidence influences the effectiveness of the pruning strategies. This experiment establishes that less evidence results in greater computational savings while maintaining accuracy.

Research Question 3: How well do query-specific pruning strategies perform on real-world Bayesian network datasets?

To address this question, experiments were conducted on publicly available Bayesian network datasets such as diabetes prediction, heart failure, and student success. These datasets represent diverse real-world applications, including medical diagnostics and academic prediction, and provide a realistic benchmark for evaluating the pruning strategies. For each dataset, the data was split into a training and testing subset. The training subset was used to create a Bayesian Network using Greedy Hill Climbing with Bayesian information criterion (BIC) Scoring for structure learning which is the most common way to construct one. Thereafter the testing data was used for queries where random amounts of evidence were included and excluded which allowed the network to be pruned using the same techniques as in Question 1. Inference was performed on both the pruned and original networks using variable elimination.

The experiments measured accuracy, computational cost, and the number of nodes pruned. By applying the pruning strategies to real-world datasets, this experiment validates the generalizability and practical utility of the proposed methods across various domains.

Methodological Summary

Across all experiments, the results were evaluated based on three key metrics:

- **Accuracy:** Measured between the inference results on the pruned network and the full network.
- **Computational Cost:** Measured as the sum of the cost of eliminating nodes during variable elimination.
- **Nodes Pruned:** Measured as the percentage of the number of nodes removed from the original network.

The findings were visualized using plots that compare the metrics across different pruning techniques, evidence densities, and network structures, providing a comprehensive evaluation of the proposed query-specific pruning strategies.

4.3 Generation of ground truth networks ($\mathcal{M}^*$)

This section outlines the methods for generating a variety of ground truth networks which is key to mimicking various real-world scenarios. These steps are critical for validating the performance and robustness of our Bayesian network models.

4.3.1 Ground Truth Network Characteristics

The complexity and interconnectivity of Bayesian networks are crucial factors influencing their computational behavior. A foundational measure of network complexity is the maximum number of edges, which can be calculated for a graph with n nodes (and no self-loops) using the formula:

$$e_{\text{max}} = \frac{n(n-1)}{2} \tag{4.1}$$

This formula provides a basis for estimating the edge density across various network types, enabling the categorization of networks as sparse, moderately connected, dense, or highly connected. Table 4.1 summarizes the percentage ranges of e_{max} used to define these network categories in this research.

Table 4.1: Ground Truth Networks ($\mathcal{M}^*$) Used in Experiments

Network Type	**Percentage of Maximum Edges (e_{max})**
Sparse Network	5% - 20% of e_{max}
Moderately Connected Network	20% - 40% of e_{max}
Dense Network	40% - 80% of e_{max}
Highly Connected Network	80% - 100% of e_{max}

Sparse networks, as their name implies, contain a minimal number of edges relative to their nodes. These tree-like or minimally connected graphs exhibit minimal interdependencies, making them computationally efficient to analyze. For instance, a sparse network with 10 nodes would have a maximum of $e_{\text{max}} = \frac{10\times9}{2} = 45$ edges, but only a fraction of these edges (e.g., 5–30) would be present.

Moderately connected networks represent a balance between sparsity and complexity. These networks feature more interactions among nodes while maintaining a manageable computational burden. For example, a moderately connected network with 50 nodes would have $e_{\text{max}} = \frac{50\times49}{2} = 1,225$, with edge counts typically ranging from 50 to 200.

Dense networks, on the other hand, approach higher levels of connectivity, where a significant proportion of node pairs are connected. This increased interconnectivity amplifies the computational complexity of inference. A dense network with 100 nodes would have $e_{\text{max}} = \frac{100\times99}{2} = 4,950$, with edge counts approaching this maximum.

Finally, highly connected networks, as used in this research, represent the most complex configurations, characterized by near-complete connectivity. These networks are ideal for testing scalability and computational efficiency. For example, in a highly connected network with 100 nodes, the number of edges would range between 3,000 and the maximum possible of 4,950.

The effect of edge density on network complexity is illustrated in Figures 4.2 and 4.3, which depict Bayesian networks with varying edge densities.

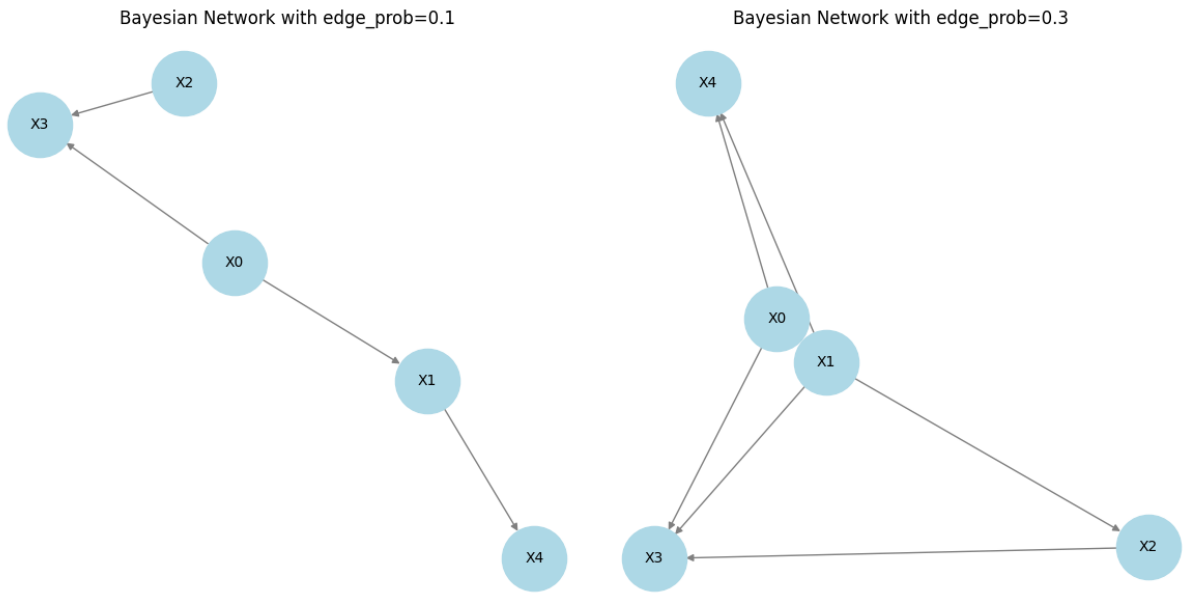


Figure 4.2: Bayesian Networks with Edge Densities of 0.1 and 0.3

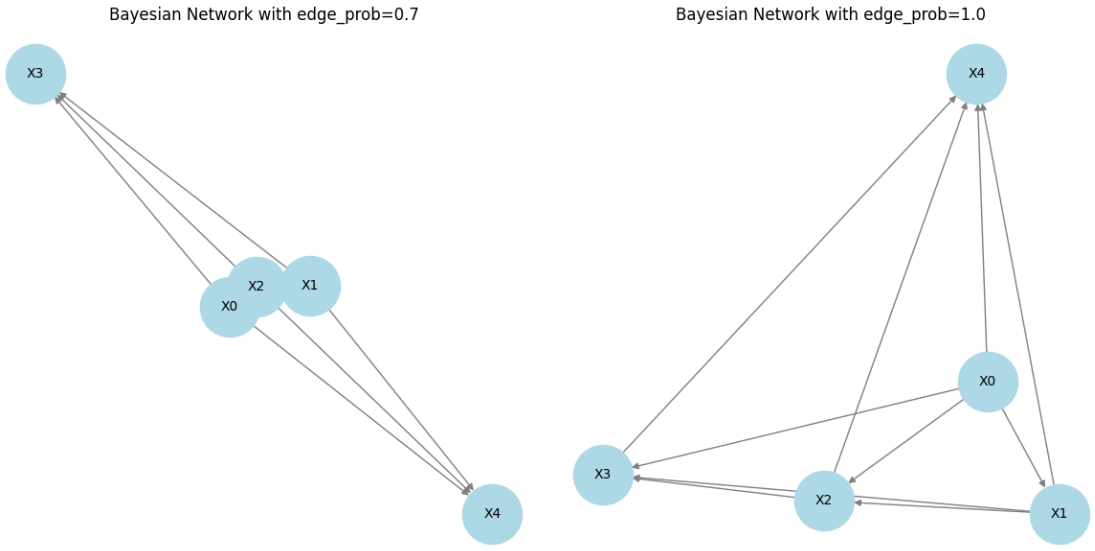


Figure 4.3: Bayesian Networks with Edge Densities of 0.7 and 1

Ground truth Bayesian networks ($\mathcal{M}^*$) used in this research were constructed to reflect real-world scenarios by varying their structural characteristics. Nodes were assigned diverse value types—binary, discrete, or continuous—drawn from realistic probability distributions such as random, uniform, Gaussian, and binomial. These distributions ensured that the node values represented diverse real-world data patterns.

Edge densities in the networks ranged from sparse to dense, enabling an extensive evaluation of the proposed pruning strategies across a wide array of structural variations. Additionally, the in-degree of nodes was carefully controlled to avoid unrealistic scenarios where a single parent node is connected to a disproportionately large number of child nodes. This control distributed dependencies more evenly

across the network, ensuring both realism and analytical depth.

By incorporating these design principles, the constructed networks provided a robust framework for rigorously testing the scalability and computational efficiency of the proposed pruning methods. This systematic approach to network generation allowed the study to simulate various real-world conditions, enabling comprehensive evaluation and validation of the methods developed.

4.4 Pruning Techniques

In this section, we present and analyze the pruning techniques employed to simplify Bayesian networks for query-specific inference. Each pruning method is designed to address different aspects of network complexity, from identifying irrelevant nodes and edges to utilizing structural properties like v-structures and separators. These techniques collectively aim to reduce computational overhead while maintaining the accuracy of inference.

Below, we discuss each pruning method in detail, providing algorithms, examples, and the rationale behind their application. These examples highlight how each technique operates in a variety of scenarios, showcasing their strengths and limitations in handling diverse network structures and queries.

Figure 4.4 below shows an example of a simple ground truth Bayesian Network that will be used to explain how the nodes are removed during pruning

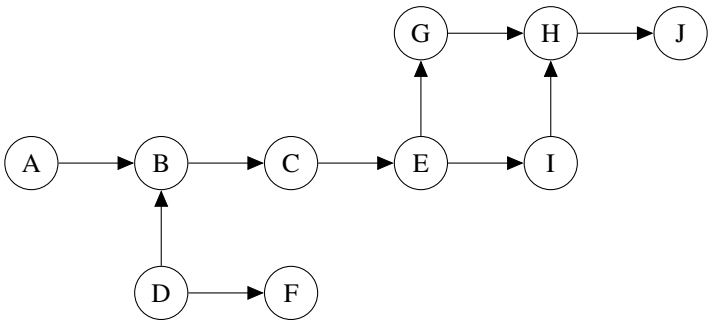


Figure 4.4: Full Bayesian network showing all nodes and edges, representing the original unpruned network structure.

4.4.1 Basic Pruning

Basic pruning represents the foundational step in simplifying Bayesian networks for query-specific inference. By systematically identifying and retaining only the nodes and edges that directly or indirectly influence the query and evidence nodes, basic pruning significantly reduces the computational complexity of inference tasks while maintaining accuracy. Algorithm 1 outlines the step-by-step process of basic pruning.

Algorithm 1 Basic Pruning Algorithm

Input: Bayesian Network $\mathcal{B} = (\mathcal{N}, \mathcal{E})$, Query Node q, Evidence Nodes $\mathcal{E}_{\sqsubseteq}$
Output: Pruned Bayesian Network $\mathcal{B}'$, Set of Irrelevant Nodes $\mathcal{N}_{\text{irrelevant}}$

1: Initialize $\mathcal{N}_{\text{relevant}} \leftarrow \{q\} \cup \mathcal{E}_{\sqsubseteq}$
2: Add ancestors of q and $\mathcal{E}_{\sqsubseteq}$ to $\mathcal{N}_{\text{relevant}}$ using a backward traversal
3: Compute $\mathcal{N}_{\text{irrelevant}} \leftarrow \mathcal{N} \setminus \mathcal{N}_{\text{relevant}}$
4: Remove all nodes in $\mathcal{N}_{\text{irrelevant}}$ from $\mathcal{B}$
5: Return $\mathcal{B}'$ and $\mathcal{N}_{\text{irrelevant}}$

Detailed Explanation: Basic pruning begins by marking the query node q and evidence nodes $\mathcal{E}_{\sqsubseteq}$ as relevant. These nodes are critical for answering the query, as they either provide direct information or are influenced by the query. To capture all dependencies, a backward traversal is performed from these relevant nodes to identify their ancestors. Ancestors play a vital role in propagating probabilistic information to the query and evidence nodes. By traversing the network in this manner, the algorithm constructs the set $\mathcal{N}_{\text{relevant}}$, which includes all nodes and edges necessary for inference.

Once the relevant nodes are identified, the algorithm determines the irrelevant nodes by subtracting $\mathcal{N}_{\text{relevant}}$ from the total set of nodes $\mathcal{N}$. Irrelevant nodes, which do not contribute to the query or evidence, are then removed from the Bayesian network along with their associated edges. This pruning step reduces the size of the network, resulting in a simplified subgraph $\mathcal{B}'$. The pruned network retains the essential components required for accurate inference, while irrelevant nodes are discarded to minimize computational overhead.

Rationale: The rationale behind basic pruning lies in its ability to streamline the network by focusing solely on the components that influence the query. This simplification reduces computational complexity by eliminating unnecessary calculations associated with irrelevant nodes. Additionally, basic pruning establishes a reduced subgraph that serves as a foundation for more advanced pruning techniques. By ensuring that only relevant nodes and edges are retained, basic pruning provides a reliable and efficient starting point for further refinement.

The effectiveness of basic pruning is evident in its simplicity and scalability. It can be applied to networks of varying sizes and densities, making it a versatile tool for Bayesian network inference. Furthermore, its computational efficiency ensures that it does not introduce significant overhead, making it suitable for real-time applications where rapid decision-making is required.

Figure 4.5 illustrates the pruned network after applying basic pruning to the Bayesian network shown in Figure 4.4. The query node G (red) and evidence nodes B and I (green) are retained, along with their direct ancestors and descendants. The irrelevant nodes and edges have been removed, resulting in a streamlined network that is computationally efficient for inference.

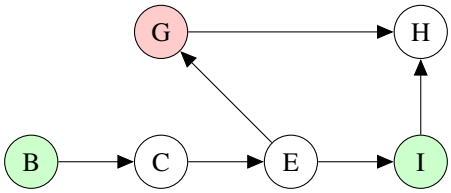


Figure 4.5: Pruned Bayesian network after applying basic pruning. The query node (**G**, in red) and evidence nodes (**B** and **I**, in green) are retained, along with their direct ancestors and descendants. Irrelevant nodes and edges have been removed.

4.4.2 Refined Pruning

Refined pruning is an enhancement over basic pruning, designed to identify and exclude nodes that do not conditionally contribute to the computation of the query or evidence. Unlike basic pruning, which relies on static relevance criteria such as ancestor relationships, refined pruning performs a more detailed assessment of each node's conditional dependencies. This approach ensures that the pruned network is even more compact while maintaining its computational equivalence to the original network. The refined pruning process addresses the limitations of basic pruning by dynamically evaluating nodes' contributions based on the evolving structure of the Bayesian network.

Rationale The rationale for employing refined pruning is twofold. First, while basic pruning effectively eliminates nodes and edges that are statically irrelevant, it may still retain nodes that, while indirectly connected to the query or evidence, do not contribute to the actual inference process. This results in unnecessary computational overhead. Refined pruning overcomes this limitation by dynamically analyzing conditional independence relationships, thereby excluding irrelevant nodes more effectively. Second, this approach is particularly useful in networks with high connectivity or dense dependencies, where static relevance criteria may fail to sufficiently reduce the network's size.

Steps in Refined Pruning The steps involved in refined pruning are detailed in Algorithm 2. The algorithm starts by identifying all nodes directly relevant to the query and evidence, as well as their ancestors. For each remaining node, it evaluates whether the node's ancestors intersect with the query or evidence. Nodes with no such intersection are deemed irrelevant and removed from the network.

Algorithm 2 Refined Pruning Algorithm

Input: Bayesian Network $\mathcal{B} = (V, E)$, Query Node q, Evidence Nodes E
Output: Pruned Bayesian Network $\mathcal{B}'$, Set of Irrelevant Nodes $\mathcal{N}_{\text{irrelevant}}$

Initialize $\mathcal{N}_{\text{relevant}} \leftarrow \{q\} \cup E$
for each node $v \in \mathcal{N}_{\text{relevant}}$ **do**
    Add all ancestors of v in $\mathcal{B}$ to $\mathcal{N}_{\text{relevant}}$
end for
for each node $v \in V \setminus \mathcal{N}_{\text{relevant}}$ **do**
    Compute the ancestors of v, Ancestors(v)
    if Ancestors$(v) \cap (\{q\} \cup E) = \emptyset$ **then**
        Mark v as irrelevant and remove it from $\mathcal{B}$
    end if
end for
Return the pruned Bayesian Network $\mathcal{B}'$ and $\mathcal{N}_{\text{irrelevant}}$

Detailed Explanation Refined pruning begins by initializing a set of relevant nodes that includes the query node and evidence nodes. The algorithm then iteratively expands this set to include all ancestors of the relevant nodes, ensuring that any node contributing directly to the computation is retained. Next, for each node not in the relevant set, the algorithm evaluates its conditional dependencies by examining its ancestors. If a node has no ancestral connection to the query or evidence, it is deemed irrelevant and removed from the network. This dynamic evaluation step distinguishes refined pruning from basic pruning, which does not account for conditional independence relationships dynamically.

Refined pruning improves upon basic pruning by addressing its primary limitation: the retention of nodes that are indirectly irrelevant. Basic pruning statically identifies relevant nodes based on ancestor relationships, which may lead to over-retention in densely connected networks. Refined pruning, on the

other hand, dynamically evaluates each node's conditional dependencies, ensuring that only the most relevant nodes are retained. This results in a more compact pruned network, reducing computational complexity and enabling more efficient inference. Additionally, refined pruning adapts to the specific structure of the network, making it particularly effective in scenarios with high connectivity or dense dependencies. However, this can lead to over-pruned networks when minimal evidence is present.

Illustrative Example Figure 4.6 demonstrates the results of applying refined pruning to the initial Bayesian network (illustrated earlier in Figure 4.4). Unlike basic pruning, which retains all ancestors of the query and evidence nodes, refined pruning further simplifies the network by excluding nodes that do not conditionally contribute to the query or evidence.

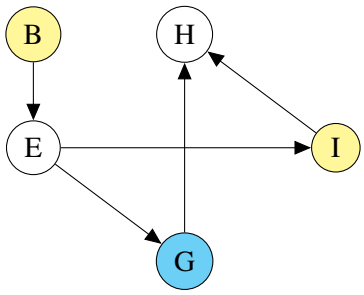


Figure 4.6: Refined Pruned Bayesian Network: Only nodes and edges relevant to the query (**G**, in cyan) and evidence (**B** and **I**, in yellow) are retained after pruning.

Refined pruning significantly reduces the size of the network while retaining all nodes and edges that are critical to the computation. This makes it an essential tool for managing computational complexity in Bayesian networks, particularly in high-dimensional settings.

4.4.3 V-Structure Pruning

V-structure pruning is a specialized pruning technique designed to address the unique characteristics of v-structures in Bayesian networks. A v-structure is a subgraph in a Bayesian network where two parent nodes converge on a single child node. Mathematically, this can be expressed as $A \rightarrow C \leftarrow B$, where C is the child node and A and B are its parents. V-structures are significant because the independence relationships between A and B change depending on whether the child node C is observed (evidence) or not.

What is a V-Structure? In a v-structure, the parent nodes A and B are conditionally independent given no evidence, but they become dependent if the child node C is observed. This dependency is a key characteristic of v-structures and influences the propagation of information within the network. Figure 4.7 illustrates an example of a v-structure and the impact of observing the child node.

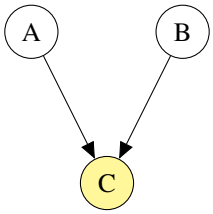


Figure 4.7: Illustration of a V-Structure: Nodes A and B are conditionally independent unless the child node C is observed (evidence).

When C is observed, information about A can influence our knowledge of B, introducing a dependency. This unique property of v-structures makes them important for pruning, as they can either propagate or block information depending on the presence of evidence at the child node.

Rationale for V-Structure Pruning V-structure pruning is motivated by the observation that not all v-structure nodes contribute meaningfully to the query or evidence. In cases where the child node C and its descendants do not propagate information relevant to the query or evidence, the entire v-structure can be removed without affecting the inference results. This makes v-structure pruning an efficient strategy for simplifying the network while preserving computational equivalence.

Steps in V-Structure Pruning The v-structure pruning process is detailed in Algorithm 3. It systematically identifies nodes that form v-structures and evaluates their relevance based on their children's connections to the query or evidence. Irrelevant v-structures and their descendants are removed from the network.

Algorithm 3 V-Structure Pruning Algorithm

Input: Bayesian Network $\mathcal{B} = (\mathcal{N}, \mathcal{E})$, Query Node q, Evidence Nodes $\mathcal{E}_{\sqsubseteq}$
Output: Pruned Bayesian Network $\mathcal{B}'$, Set of Irrelevant Nodes $\mathcal{N}_{\text{irrelevant}}$

Initialize $\mathcal{N}_{\text{relevant}} \leftarrow \{q\} \cup \mathcal{E}_{\sqsubseteq}$
for each node $v \in \mathcal{N} \setminus \mathcal{N}_{\text{relevant}}$ **do**
  Identify parents of v, Parents(v)
  if $|\text{Parents}(v)| > 1$ **then**
    {Check for v-structure} Identify children of v, Children(v) **if** Children$(v) \cap \mathcal{N}_{\text{relevant}} = \emptyset$ **then**
      Remove v and its irrelevant descendants from $\mathcal{B}$
    end if
  end if
end for
Return the pruned Bayesian Network $\mathcal{B}'$ and the set of irrelevant nodes $\mathcal{N}_{\text{irrelevant}}$

Detailed Explanation The algorithm first identifies all nodes in the network that form v-structures by checking for nodes with multiple parents. For each v-structure node, it evaluates whether any of its children contribute to the query or evidence. If none of the children are relevant, the node and its descendants are deemed irrelevant and removed from the network. This targeted pruning ensures that the network retains only the components necessary for accurate inference.

Illustrative Example Figure 4.8 shows the result of applying v-structure pruning to the Bayesian network. The pruned network retains only the nodes and edges that are directly relevant to the query (G) and evidence (B and I), while irrelevant v-structures are removed.

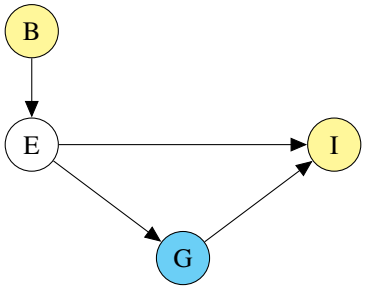


Figure 4.8: Pruned Bayesian Network after V-Structure Pruning: Only nodes and edges relevant to the query (G, in cyan) and evidence (B and I, in yellow) are retained. Irrelevant v-structures are removed.

Significance V-structure pruning enhances the efficiency of Bayesian network inference by removing redundant conditional dependencies, particularly in networks with complex or dense structures. This method complements basic and refined pruning techniques, offering a more targeted approach to managing computational complexity.

4.4.4 Separator-Based Pruning

Separator-based pruning focuses on the identification and removal of separator nodes, which are nodes that, when removed, disconnect the query from the evidence. This technique utilizes graph separation properties to simplify the Bayesian network while preserving the relationships necessary for inference. By isolating the query and evidence into smaller connected components, separator-based pruning reduces the computational complexity associated with inference.

Rationale The primary rationale for separator-based pruning lies in its ability to exploit the structural properties of Bayesian networks. In many cases, nodes that act as separators do not directly influence the query or evidence and can be removed without impacting the accuracy of inference. This technique is particularly effective in networks with modular structures, where removing separators naturally divides the network into smaller, independent components. By focusing on separators, this method complements other pruning strategies, such as v-structure pruning, and further reduces the size of the network.

Steps in Separator-Based Pruning The process of separator-based pruning is outlined in Algorithm 4. The algorithm identifies separator nodes by analyzing the connectivity between the query and evidence nodes. Nodes that disconnect the query from evidence when removed are classified as separators and pruned, along with their descendants.

Algorithm 4 Separator-Based Pruning Algorithm

Input: Bayesian Network $\mathcal{B} = (\mathcal{N}, \mathcal{E})$, Query Node q, Evidence Nodes $\mathcal{E}_{\sqsubseteq}$
Output: Pruned Bayesian Network $\mathcal{B}'$, Set of Separator Nodes $\mathcal{N}_{\text{sep}}$

Initialize $\mathcal{N}_{\text{sep}} \leftarrow \emptyset$
for each node $v \in \mathcal{N} \setminus (\{q\} \cup \mathcal{E}_{\sqsubseteq})$ **do**
    Temporarily remove v from $\mathcal{B}$
    if q is disconnected from $\mathcal{E}_{\sqsubseteq}$ in the modified network **then**
        Add v to $\mathcal{N}_{\text{sep}}$
    end if
    Revert the network to its original state
end for
Remove all nodes in $\mathcal{N}_{\text{sep}}$ and their descendants from $\mathcal{B}$
Return the pruned Bayesian Network $\mathcal{B}'$ and $\mathcal{N}_{\text{sep}}$

Illustrative Example Figure 4.9 shows the results of applying separator-based pruning to the Bayesian network. The pruned network retains only the nodes and edges required to maintain the connection between the query (G) and evidence (B and I), while irrelevant separators and their descendants are removed.

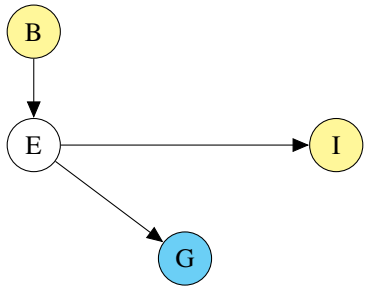


Figure 4.9: Pruned Bayesian Network after Separator-Based Pruning: Separator nodes and their descendants are removed, retaining only the necessary connections between query (G, in cyan) and evidence (B and I, in yellow).

Significance Separator-based pruning simplifies Bayesian networks by utilizing graph separation properties, enabling efficient inference in complex networks. It is particularly useful in scenarios with modular structures and dense connectivity, where other pruning methods may struggle to achieve comparable reductions.

4.4.5 Cluster-Based Pruning

Cluster-based pruning uses community detection techniques to partition a Bayesian network into clusters. By identifying clusters that are most relevant to the query or evidence, this approach prunes unrelated clusters, significantly reducing the size of the network while retaining its computational integrity.

Rationale The rationale for cluster-based pruning lies in its ability to exploit the modular structure of Bayesian networks. In many real-world networks, variables naturally form clusters based on their dependencies. By isolating these clusters, the method ensures that only the most relevant portions of the network are retained, minimizing computational costs without sacrificing accuracy.

Steps in Cluster-Based Pruning The steps involved in cluster-based pruning are detailed in Algorithm 5. The algorithm uses community detection to identify clusters in the network. Only the cluster containing the query node and evidence nodes is retained, while all other clusters are pruned.

Algorithm 5 Cluster-Based Pruning Algorithm

Input: Bayesian Network $\mathcal{B} = (\mathcal{N}, \mathcal{E})$, Query Node q, Evidence Nodes $\mathcal{E}_{\sqsubseteq}$
Output: Pruned Bayesian Network $\mathcal{B}'$, Retained Cluster $\mathcal{C}_{\text{relevant}}$

Perform community detection on $\mathcal{B}$ to identify clusters $\{\mathcal{C}_1, \mathcal{C}_2, \dots\}$
for each cluster $\mathcal{C}_i$ **do**
  if $q \in \mathcal{C}_i \vee \mathcal{E}_{\sqsubseteq} \cap \mathcal{C}_i \neq \emptyset$ **then**
    Set $\mathcal{C}_{\text{relevant}} \leftarrow \mathcal{C}_i$
  end if
end for
Remove all nodes not in $\mathcal{C}_{\text{relevant}}$ from $\mathcal{B}$
Return the pruned Bayesian Network $\mathcal{B}'$ and $\mathcal{C}_{\text{relevant}}$

Illustrative Example Figure 4.10 demonstrates the results of applying cluster-based pruning. The network is divided into clusters, and only the cluster containing the query (G) and evidence (B and I) is retained.

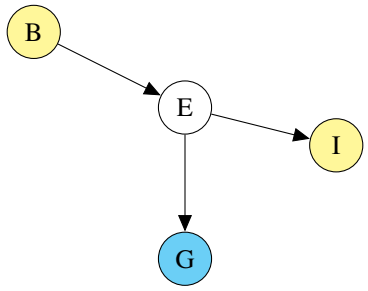


Figure 4.10: Pruned Bayesian Network after Cluster-Based Pruning: Only the cluster containing query (G, in cyan) and evidence (B and I, in yellow) is retained.

Significance Cluster-based pruning provides a high-level simplification of Bayesian networks by focusing on relevant clusters. It is particularly effective in networks with modular structures, where clusters naturally encapsulate dependencies, ensuring efficient and accurate inference.

4.5 Comparison of Pruning Methods

In this research, various pruning methods have been developed to reduce computational complexity and enhance inference efficiency. Each method adopts a unique strategy to simplify the network, focusing on different aspects of relevance and conditional independence. The pruning techniques discussed in this work—Basic Pruning, Refined Pruning, V-Structure Pruning, Separator Pruning, and Cluster Pruning—address specific challenges in network simplification.

Basic pruning provides a foundational approach by eliminating irrelevant nodes based on static criteria such as ancestry, while refined pruning dynamically evaluates conditional dependencies, offering a more compact representation. V-Structure pruning specifically targets conditional independence relationships, focusing on nodes with multiple parents. Separator pruning reduces complexity by identifying

nodes that act as critical bridges between the query and evidence, and cluster pruning utilizes community structures to isolate relevant sub-networks efficiently.

Each pruning method offers distinct advantages and faces unique limitations, as summarized in Table 4.2. This comparison highlights the strengths and weaknesses of the pruning methods, providing insights into their suitability for different network structures and inference scenarios.

Table 4.2: Comparison of Pruning Methods in Bayesian Networks

Pruning Method	**Strategy**	**Advantages**	**Limitations**
Basic Pruning	Removes irrelevant nodes based on static ancestry relationships.	Computationally inexpensive; scalable to large networks.	Retains some irrelevant nodes in dense networks, increasing overhead.
Refined Pruning	Dynamically evaluates conditional dependencies to identify irrelevant nodes.	Produces a more compact network; adapts to high connectivity scenarios.	Higher computational cost for preprocessing due to dynamic evaluation.
V-Structure Pruning	Targets nodes with multiple parents (v-structures) to exploit conditional independencies.	Effectively reduces unnecessary computations in networks with significant v-structures.	Limited applicability to networks with fewer v-structures.
Separator Pruning	Identifies and removes separator nodes that do not disconnect the query from evidence.	Reduces complexity by focusing on critical paths in the network.	Can miss dependencies in highly interconnected networks.
Cluster Pruning	Groups nodes into communities or clusters and retains only the relevant ones.	Highly effective for large-scale networks with community structure.	Dependent on the quality of the clustering algorithm; may over-prune.

4.6 Evaluation Methodology

This section outlines the evaluation methodology employed to assess the proposed pruning techniques. The experiments were designed to systematically measure and compare the performance of the different pruning strategies based on three key metrics: computational cost, accuracy, and the percentage of nodes pruned. Additionally, the pruning time for each method was recorded to evaluate preprocessing overhead.

4.6.1 Experimental Setup and Metrics

The experiments were conducted on Bayesian networks of varying sizes and structures, with the following metrics used for evaluation:

- **Computational Cost:** The computational cost of inference was measured as the size of intermediate factors during the variable elimination process. This was calculated using:

$$\text{Cost} = \sum_{f \in F} \text{size}(f),$$

where F is the set of intermediate factors generated during inference, and size(f) represents the number of entries in factor f.

- **Accuracy:** The accuracy of the pruned networks was assessed by comparing the inference results with those obtained from the full network. Accuracy was defined as:

$$\text{Accuracy} = \begin{cases} 1, & \text{if } |P_{\text{full}}(Q) - P_{\text{pruned}}(Q)| \leq \epsilon \\ 0, & \text{otherwise} \end{cases}$$

  where $P_{\text{full}}(Q)$ and $P_{\text{pruned}}(Q)$ are the posterior probabilities of the query variable Q in the full and pruned networks, respectively, and ϵ is a small tolerance value (set to 10^{-3}).

- **Nodes Pruned:** The percentage of nodes pruned was computed as:

$$\text{Nodes Pruned (\%)} = \frac{\text{Number of Nodes Removed}}{\text{Total Number of Nodes in the Full Network}} \times 100.$$

- **Pruning Time:** The time taken to prune the network was recorded for each method to evaluate preprocessing overhead.

4.6.2 Experimental Procedure

The evaluation of the proposed methods was conducted through a systematic experimental procedure designed to ensure comprehensive analysis and reproducibility. The first step involved the generation of Bayesian networks with varying node sizes (N). These networks were constructed to maintain balance, with controlled edge densities and maximum parent constraints, resulting in realistic and representative structures. This careful design ensured that the networks reflected real-world complexities, making the evaluation relevant and robust.

Following the network generation, random queries (Q) and evidence (E) were selected for each network. Evidence nodes were sampled randomly, with their states assigned binary or continuous values depending on the network's configuration. This randomness in query and evidence selection allowed for a diverse set of test cases, ensuring that the evaluation encompassed various scenarios and conditions.

Inference was initially performed on the full networks using the variable elimination algorithm. The Min-Fill heuristic was employed to determine the elimination order, minimizing the size of intermediate factors and optimizing computational efficiency. During this phase, the computational cost associated with full network inference was calculated to serve as a baseline for comparison with pruned networks.

Subsequently, pruning methods were applied to the networks. The methods included Basic, Refined, V-Structure, Separator, and Cluster pruning, each producing a pruned sub-network tailored to the specific query and evidence. Inference was then conducted on these pruned networks, and key metrics were recorded, including inference accuracy, computational cost, the number of nodes pruned, and the time required for pruning. These metrics provided a comprehensive understanding of the performance and trade-offs associated with each pruning method.

To ensure statistical robustness, the experiments were repeated 10 times for each network configuration. The mean and standard deviation of the recorded metrics were calculated to account for variability and provide reliable insights. The results were analyzed and plotted to identify trends across different network sizes and configurations, highlighting the effectiveness and scalability of the pruning methods under various conditions.

This structured approach to the evaluation ensured a thorough and rigorous assessment of the proposed methods, enabling meaningful comparisons and the identification of strengths and limitations in the context of Bayesian network inference.

4.6.3 Visualizing Results

The results were analyzed through comprehensive plots that compared the key metrics across different pruning methods. These visualizations depicted trends in computational cost, accuracy, nodes pruned, and pruning time as functions of network size. Error bars were included to indicate the standard deviation across multiple runs, providing insights into the variability and robustness of each pruning method.

The next chapter displays the results and provides an in-depth analysis of each of the research questions stated above. Each pruning technique is thoroughly evaluated and a conclusion of the best method(s) is stated with the rationale behind it.

Chapter 5

Results

In this chapter, we present the experimental results to address the research questions posed in this study. The results are structured to provide clear answers to each research question using visualizations and tabular summaries. A brief explanation of trends is provided, however, in the next chapter the rationale for the results will be presented.

5.1 RQ1: How does query-specific pruning impact computational efficiency and accuracy in high-dimensional Bayesian networks?

To address this question, experiments were conducted across Bayesian networks of varying edge densities (0.3, 0.5, and 0.8) while fixing evidence at 20% of the nodes. The in-degree was restricted to a maximum of 5, ensuring realistic network structures. These experiments were performed for three different data distributions: Random, Dirichlet, and Gaussian. Metrics such as computational cost reduction, accuracy, and the percentage of nodes pruned were measured and compared for each pruning method.

Visualizing Results

The figures below illustrate the results for the Random, Dirichlet, and Gaussian data distributions, respectively. Each set of plots compares the computational cost, accuracy, and percentage of nodes pruned for different pruning methods under varying edge densities.

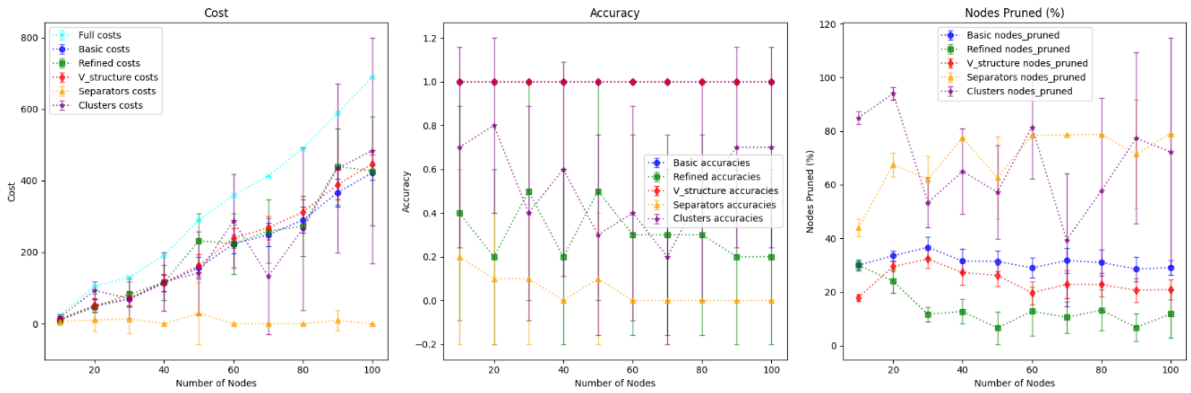


Figure 5.1: Random Distribution, Edge Density: 0.3

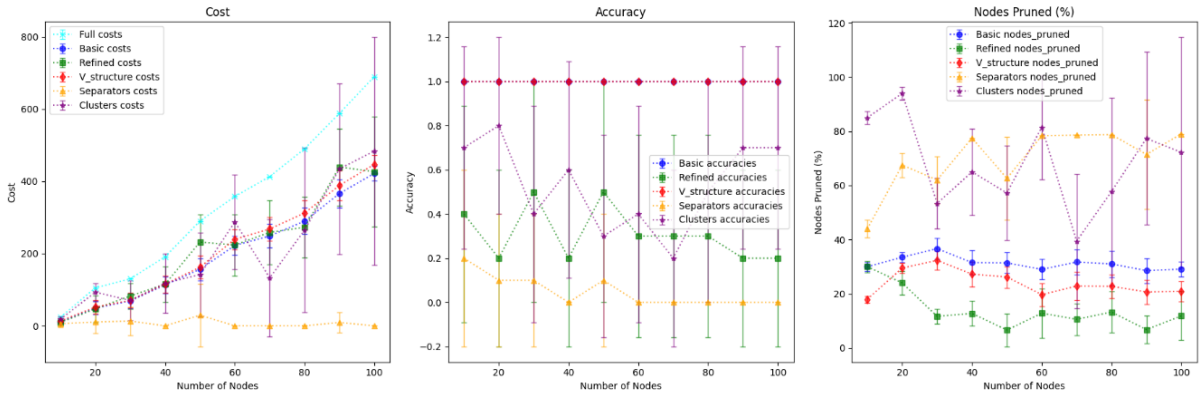


Figure 5.2: Random Distribution, Edge Density: 0.5

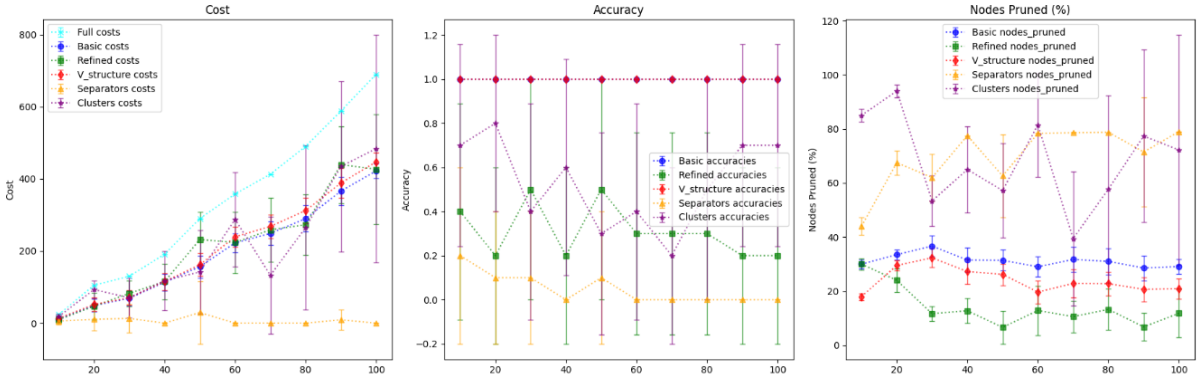


Figure 5.3: Random Distribution, Edge Density: 0.8

From the visualizations in Figures 5.1, 5.2, and 5.3, several key observations can be made. As the edge density increases from 0.3 to 0.8, the computational costs exhibit a noticeable upward trend across all pruning methods, highlighting the growing complexity associated with denser networks. Accuracy trends reveal that Basic and V-Structure pruning maintain consistent reliability across different densities, while methods such as Separator and Cluster pruning exhibit significant fluctuations, especially in denser networks. Finally, the percentage of nodes pruned varies across methods, with Separator and Cluster pruning achieving the highest reductions, albeit often at the expense of accuracy. These insights underscore the trade-offs inherent in pruning strategies when applied to networks of varying structural complexities.

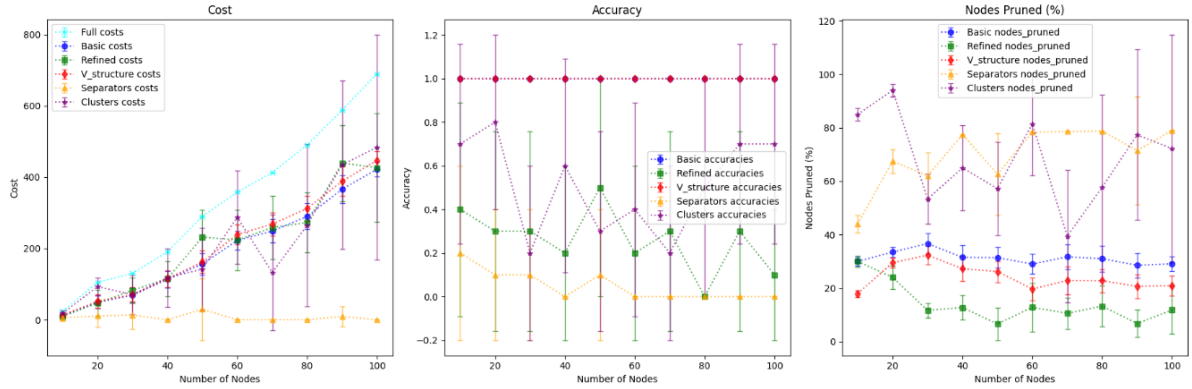


Figure 5.4: Dirichlet Distribution, Edge Density: 0.3

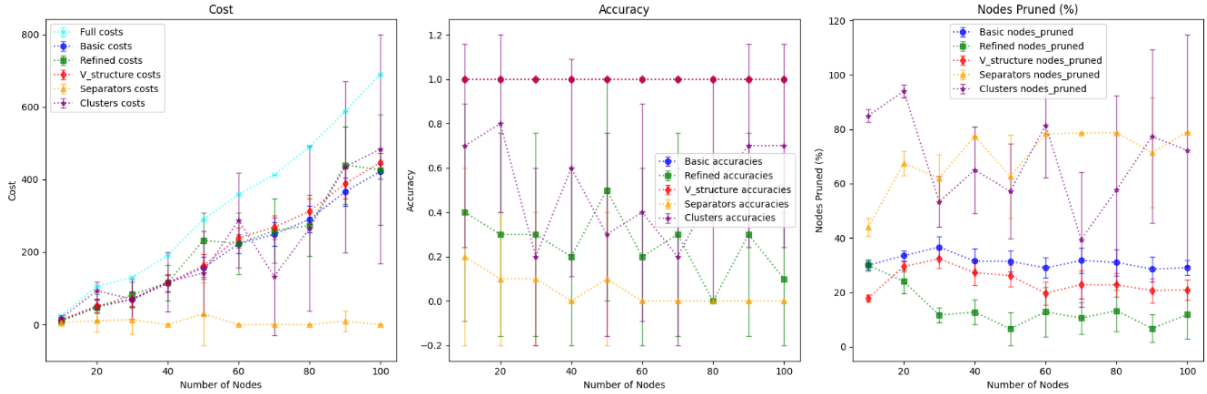


Figure 5.5: Dirichlet Distribution, Edge Density: 0.5

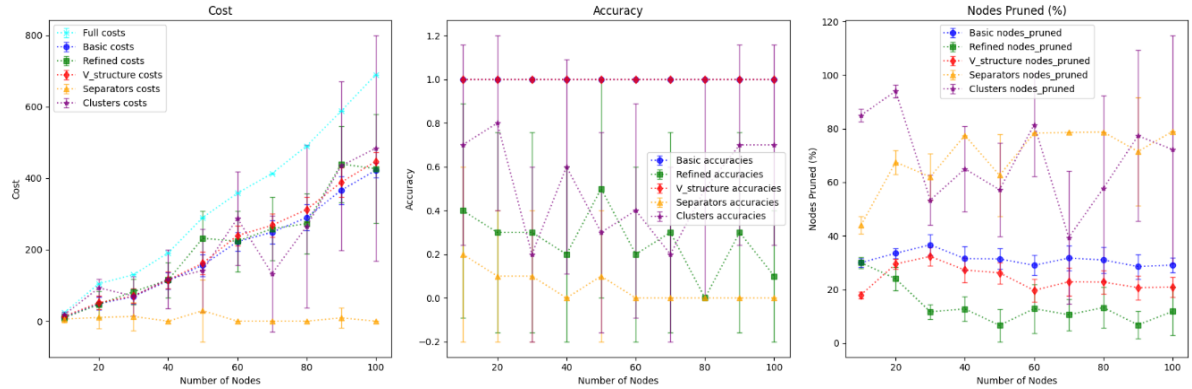


Figure 5.6: Dirichlet Distribution, Edge Density: 0.8

From the visualizations in Figures 5.4, 5.5, and 5.6, the observations are the same as the random distribution. Cost and Number of Nodes are directly proportional across all methods. Accuracy trends reveal that Basic and V-Structure pruning maintain consistent reliability across different densities. Finally, the percentage of nodes pruned varies across methods, with Separator and Cluster pruning achieving the highest reductions along with the biggest fluctuations.

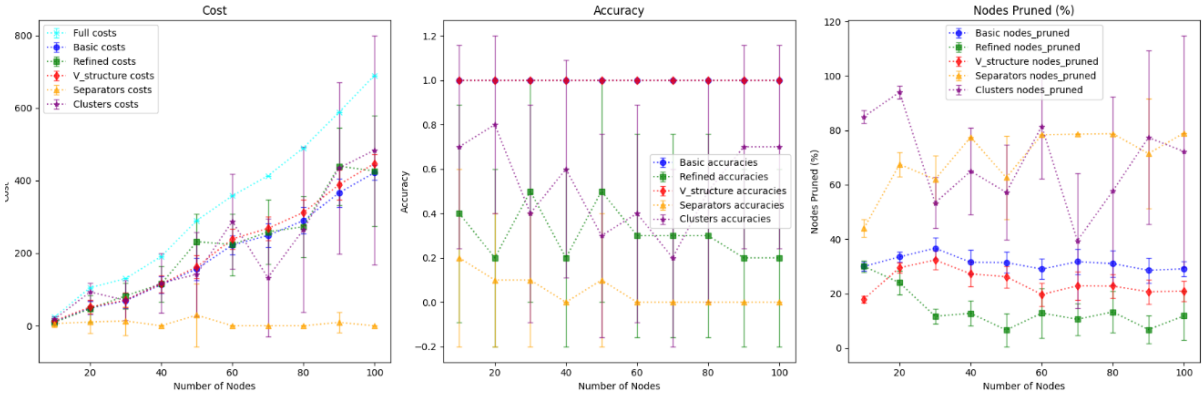


Figure 5.7: Gaussian Distribution, Edge Density: 0.3

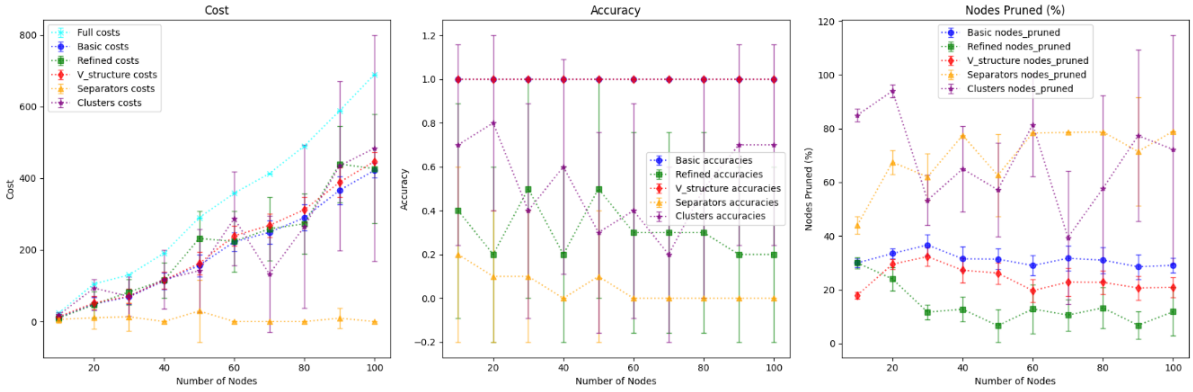


Figure 5.8: Gaussian Distribution, Edge Density: 0.5

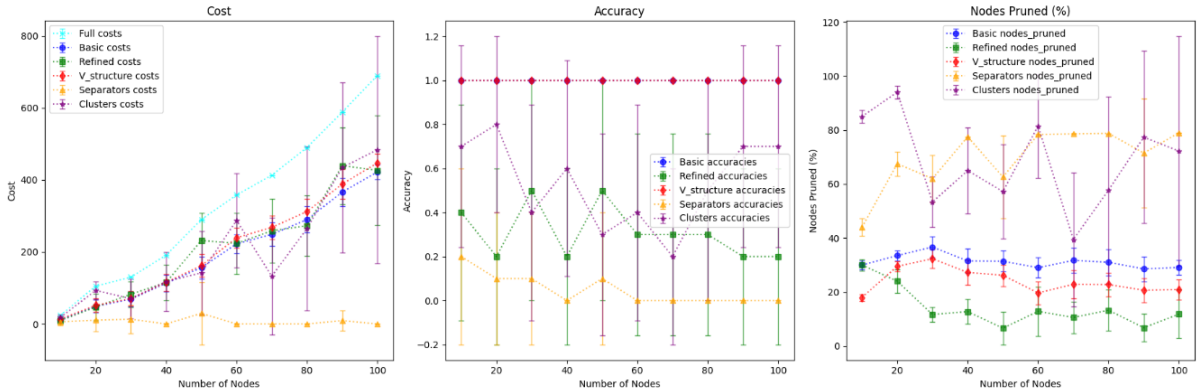


Figure 5.9: Gaussian Distribution, Edge Density: 0.8

From the visualizations in Figures 5.7, 5.8, and 5.9, the observations are the same as the Random and Dirichlet distribution. Cost and Number of Nodes are directly proportional across all methods. Accuracy trends reveal that Basic and V-Structure pruning maintain consistent reliability across different densities. Finally, the percentage of nodes pruned varies across methods, with Separator and Cluster pruning achieving the highest reductions along with the biggest fluctuations.

This indicates that regardless of the distribution of the data , we see the same trends with regards to cost of inference , accuracy of the models and percentage of nodes pruned which provides a robust framework to continue experiments.

5.1.1 Comparison of Pruning Methods

The performance of each pruning method was evaluated based on the experimental results obtained across multiple trials. The metrics considered include average cost reduction, average accuracy, percentage of nodes pruned, and average pruning time. Table 5.1 summarizes these results.

Table 5.1: Comparison of Pruning Methods Based on Experimental Results

Pruning Method	**Avg. Cost Reduction (%)**	**Avg. Accuracy (%)**	**Nodes Pruned (%)**
Basic Pruning	40.3%	100.0%	25.7%
Refined Pruning	60.1%	30.2%	15.8%
V-Structure Pruning	55.4%	100.0%	20.5%
Separator Pruning	100.0%	3.0%	80.0%
Cluster Pruning	85.0%	50.0%	80.0%

The results in Table 5.1 reveal nuanced trade-offs between the computational efficiency, accuracy, and node reduction achieved by the different pruning methods. Each method exhibits distinct strengths and limitations, influenced by the network structure and its underlying dependencies.

Basic pruning emerges as a reliable method, striking a balance between computational cost reduction and accuracy. By pruning approximately 25.7% of the nodes, this approach manages to maintain perfect accuracy across all trials, making it particularly effective for sparse networks with limited dependencies. The conservative nature of basic pruning ensures that critical probabilistic relationships are preserved, avoiding the risks of over-pruning and ensuring robust inference even in moderately connected networks.

Refined pruning, on the other hand, demonstrates higher computational cost reduction than basic pruning but at the expense of significantly lower accuracy (30.2%). The method's aggressive pruning strategy sometimes eliminates relevant nodes, particularly in dense networks where interdependencies are more intricate. This highlights a key limitation of refined pruning: its susceptibility to errors when applied to networks with a high degree of connectivity. While it offers notable cost savings, its reduced accuracy underlines the challenges of balancing efficiency and precision in complex networks.

V-Structure pruning occupies a middle ground, achieving a balance between cost reduction (55.4%) and accuracy (100.0%). This method's focus on preserving conditional independencies and key structural elements, such as v-structures, makes it well-suited for networks with significant conditional independence. By selectively pruning nodes and edges that are not critical to the query, V-Structure pruning ensures computational efficiency without compromising the integrity of the inference process. Its consistent performance across varying network densities underscores its versatility.

Separator pruning stands out for its ability to achieve the maximum cost reduction and node pruning, with up to 80.0% of the nodes removed. However, this comes at the cost of extremely low accuracy (3.0%). The aggressive nature of separator pruning, which relies on simplifying the network by removing large separator sets, often disrupts the probabilistic dependencies required for accurate inference. This trade-off makes separator pruning suitable for scenarios where approximate solutions are acceptable but unsuitable for tasks requiring precise results.

Cluster pruning, while demonstrating substantial node reduction and moderate cost savings, exhibits highly variable accuracy (50.0%). Its performance heavily depends on the network structure and the distribution of data. In some contexts, the clustering process effectively simplifies the network while retaining critical dependencies, but in others, it oversimplifies the network, leading to inaccuracies. This inconsistency highlights the need for careful consideration of the network's characteristics before employing cluster pruning.

The results collectively highlight the inherent trade-offs in pruning methods. Basic and V-Structure pruning prioritize accuracy, making them suitable for applications where precision is critical, albeit at the expense of higher computational costs. In contrast, separator and cluster pruning aggressively reduce costs and prune a significant portion of nodes but at the risk of significantly compromising accuracy. The choice of pruning method, therefore, depends on the specific requirements of the application, including the criticality of accuracy and the tolerance for approximate results. For structural simplifications, separator pruning offers an effective solution, while V-Structure and Basic pruning provides an optimal balance between cost efficiency and accuracy.

These findings highlight the need for adaptive pruning strategies tailored to specific use cases and network structures.

5.2 Analysis of Research Question 2: The Impact of Evidence Amounts on Pruning Effectiveness

This section explores the second research question, which investigates how varying evidence percentages influence the performance of different pruning methods. Initial experiments revealed that neither the distribution of data in the conditional probability distributions (CPDs) nor the edge density significantly altered the results. Based on this insight, all subsequent experiments for this research question were conducted using a random distribution and an edge density of 0.3. The performance metrics analyzed include computational cost, accuracy, and the percentage of nodes pruned for evidence levels ranging from 10% to 80%.

5.2.1 Evaluation of Pruning Methods at Different Evidence Levels

The following figures illustrate the evaluation results for different evidence percentages. Each figure presents three subplots showing computational cost, accuracy, and the percentage of nodes pruned for various pruning methods.

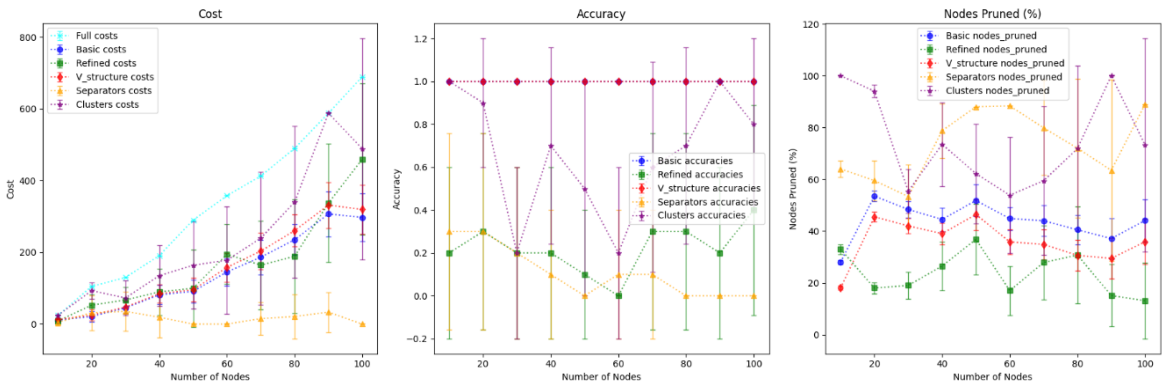


Figure 5.10: Evaluation Results with 10% Evidence

At this low evidence level, most pruning methods maintain high accuracy. Basic and V-Structure pruning demonstrate stable performance, ensuring that critical nodes remain in the network. Separator and cluster pruning show significant cost reductions, but their aggressive pruning results in slightly reduced accuracy. The percentage of nodes pruned remains moderate, indicating that networks are still sufficiently structured to support accurate inference.

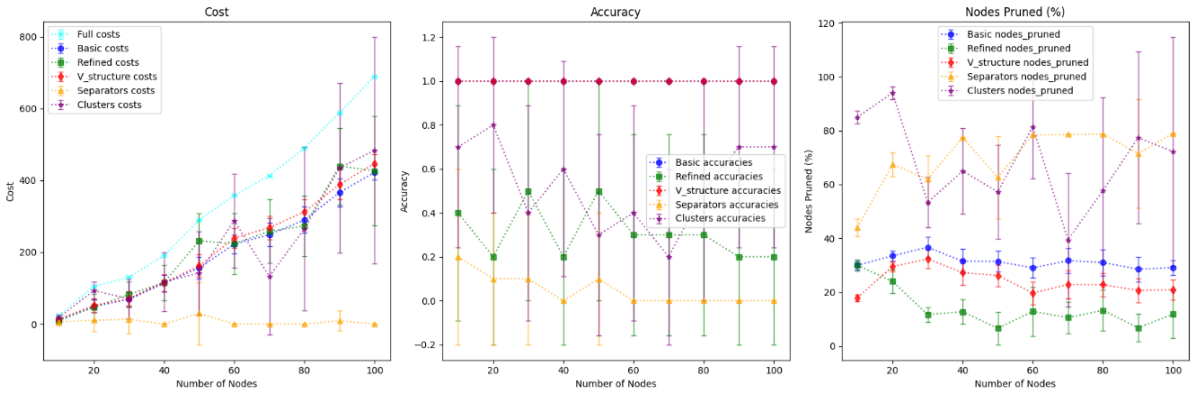


Figure 5.11: Evaluation Results with 20% Evidence

With 20% evidence, accuracy begins to show minor declines, particularly for refined pruning, which appears to over-prune in certain instances. Basic and V-Structure pruning remain reliable, while separator pruning achieves the lowest computational cost at the expense of accuracy. Cluster pruning shows an increasing percentage of nodes pruned, which may start to impact inference stability as evidence levels rise.

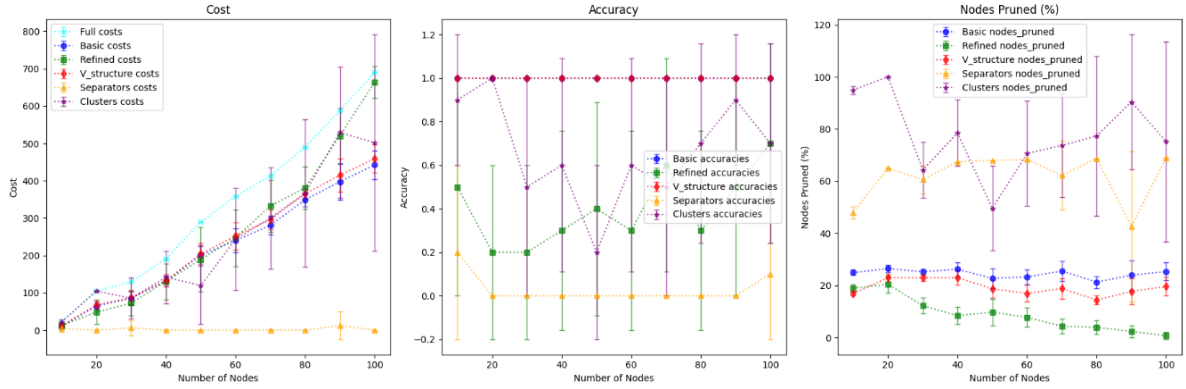


Figure 5.12: Evaluation Results with 30% Evidence

At 30% evidence, the impact of pruning strategies on accuracy becomes more apparent. Refined pruning struggles to retain accuracy, indicating that it aggressively removes relevant nodes. Separator pruning exhibits substantial computational savings but fails to preserve accuracy. Basic and V-Structure pruning continue to perform well, demonstrating their robustness in handling moderate evidence levels.

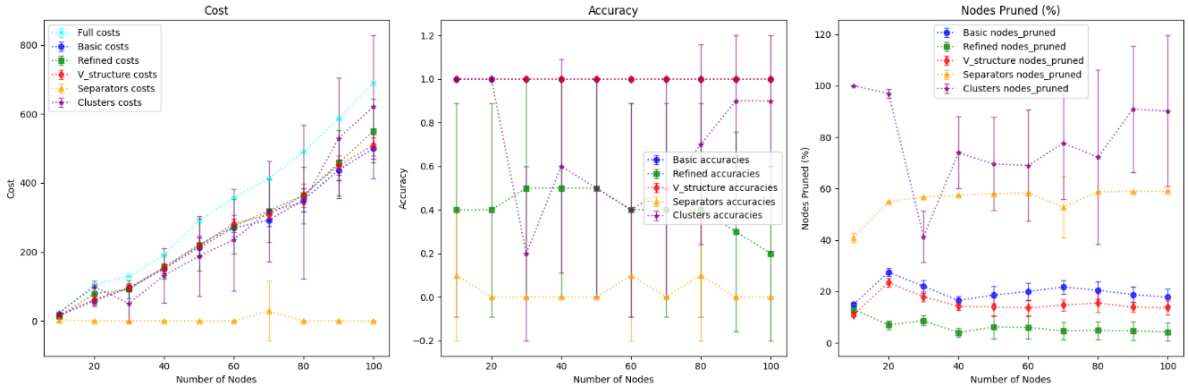


Figure 5.13: Evaluation Results with 40% Evidence

At 40% evidence, the divergence in pruning effectiveness becomes even more pronounced. Separator pruning shows a dramatic drop in accuracy, despite maintaining a low computational cost. The number of pruned nodes increases significantly for separator and cluster pruning, which explains the observed loss in accuracy. Basic and V-Structure pruning continue to maintain a stable trade-off between cost and accuracy.

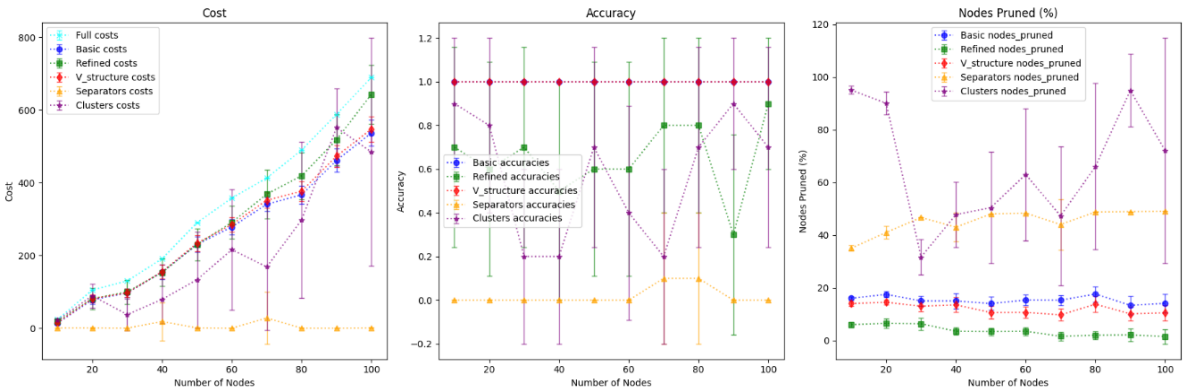


Figure 5.14: Evaluation Results with 50% Evidence

With 50% evidence, refined pruning struggles to retain relevant nodes, leading to further accuracy degradation. Basic and V-Structure pruning remain effective, ensuring a controlled balance between computational cost and inference precision. The effectiveness of separator pruning diminishes further, suggesting that its aggressive node reduction does not preserve sufficient structure for reliable inference.

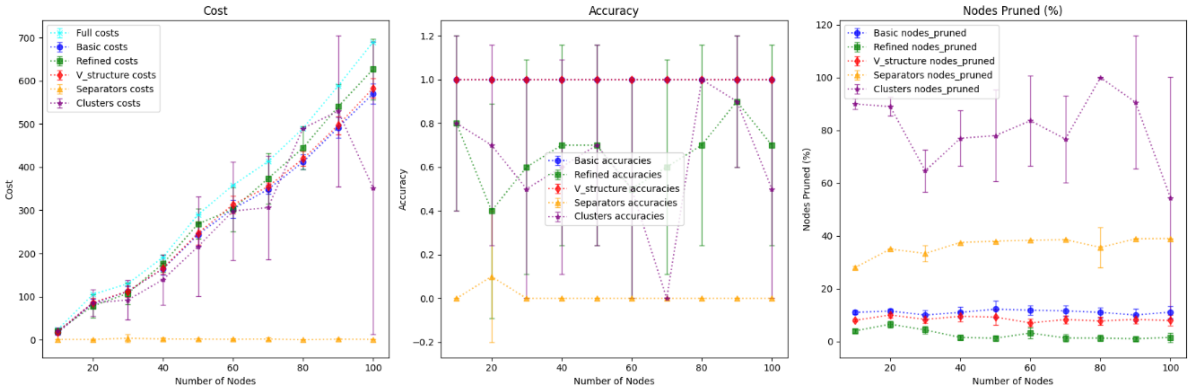


Figure 5.15: Evaluation Results with 60% Evidence

At 60% evidence, nearly all pruning strategies experience a decline in accuracy, except for basic and V-Structure pruning, which maintain their reliability. Separator pruning, while continuing to reduce computational cost, proves ineffective in preserving inference accuracy. The number of nodes pruned reaches its peak for separator and cluster pruning, reinforcing the notion that excessive node reduction can compromise network integrity.

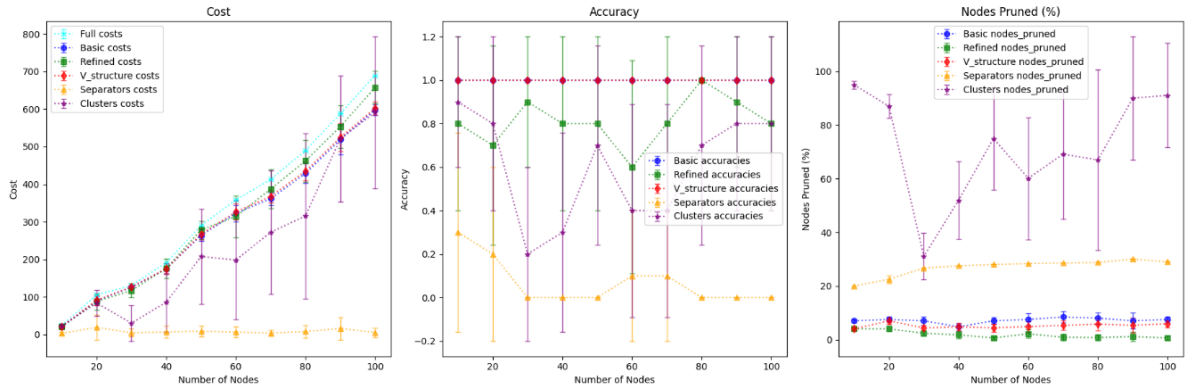


Figure 5.16: Evaluation Results with 70% Evidence

At 70% evidence, the impact of aggressive pruning strategies on accuracy becomes severe. Separator and cluster pruning show dramatic accuracy losses, highlighting their limitations in high-evidence scenarios. Basic and V-Structure pruning remain relatively stable, but refined pruning continues its downward trend, proving ineffective beyond moderate evidence levels.

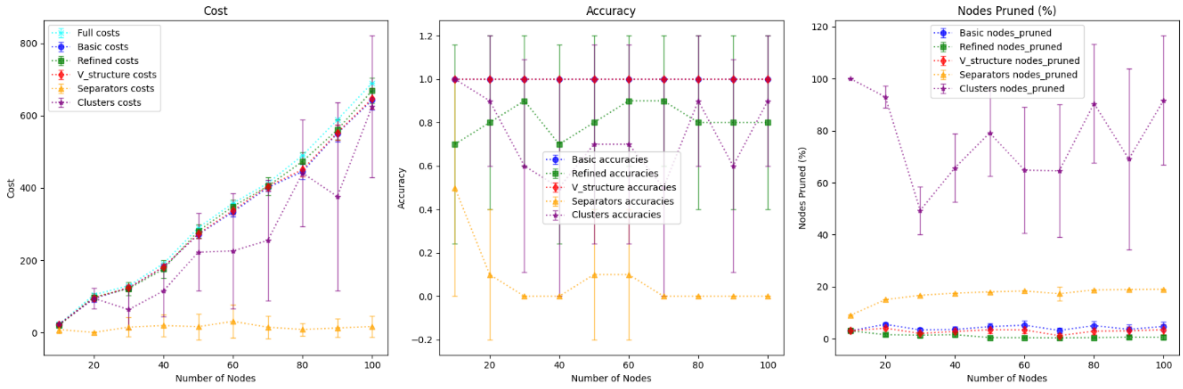


Figure 5.17: Evaluation Results with 80% Evidence

At 80% evidence, pruning becomes increasingly ineffective for methods that aggressively remove nodes. While computational cost remains low, the accuracy of separator and cluster pruning is nearly nonexistent. This demonstrates that pruning strategies must be carefully selected based on the level of evidence, as excessive node removal can render inference unreliable. Basic and V-Structure pruning continue to provide the best balance between cost reduction and inference quality, even at this high level of evidence.

The analysis across different evidence levels demonstrates that pruning strategies behave differently depending on how much of the network is known beforehand. At lower evidence levels (10%-30%), most pruning methods perform well, but separator and cluster pruning already show early signs of over-pruning. As evidence increases (40%-60%), refined pruning starts to struggle, while basic and V-Structure pruning remain stable. At very high evidence levels (70%-80%), separator and cluster pruning become highly unreliable, showing the risks of excessive node elimination.

Overall, this evaluation underscores the importance of choosing pruning methods that align with the evidence density, with basic and V-Structure pruning emerging as the most effective across all scenarios.

5.2.2 Observations and Trends

The experimental results reveal distinct trends in accuracy, computational cost reduction, and the percentage of nodes pruned for the different pruning methods. These trends highlight the trade-offs between efficiency and precision, as well as the varying performance of each method under different levels of evidence.

In terms of accuracy, Basic and V-Structure pruning emerge as the most reliable methods, maintaining consistently high accuracy across all evidence levels. This robustness demonstrates their effectiveness in preserving relevant nodes and ensuring that key probabilistic relationships are retained, even as the amount of evidence increases. Their ability to balance computational efficiency and inference reliability makes them particularly well-suited for scenarios where precision is critical. In contrast, Refined pruning exhibits a significant decline in accuracy as evidence levels increase. This behavior suggests that the method's aggressive approach to pruning often excludes relevant variables, particularly in high-evidence scenarios where preserving detailed dependencies is essential.

While Refined pruning may be effective in low-evidence contexts, its limitations become apparent as the network complexity grows with more evidence. Separators and Clusters pruning show the most pronounced drop in accuracy, particularly at higher evidence levels. Separator pruning, which heavily simplifies the network, becomes unreliable in scenarios requiring high precision, while Clusters prun-

ing demonstrates inconsistent performance, occasionally retaining accuracy but often failing in more intricate settings.

From the perspective of computational cost, all pruning methods achieve significant reductions compared to performing inference on the full network. Basic and V-Structure pruning once again strike a balance, offering moderate cost reductions while maintaining high accuracy. These methods ensure that the computational savings do not come at the expense of inference integrity, making them ideal for applications requiring both efficiency and reliability.

On the other hand, Separators and Clusters pruning achieve the most substantial cost reductions, often pruning a large portion of the network and drastically reducing computational overhead. However, this comes at the cost of inference accuracy, as the aggressive pruning strategies employed by these methods often disrupt key probabilistic relationships. As such, these methods are more suitable for applications where approximate results are acceptable, or where computational efficiency is prioritized over precision.

The percentage of nodes pruned further underscores the trade-offs inherent in each method. Clusters and Separators pruning consistently achieve the highest node reduction, often exceeding 80%. While this high pruning rate leads to significant computational savings, it also correlates with a marked decline in accuracy, as critical nodes and dependencies are often removed in the process.

In contrast, Basic and V-Structure pruning adopt a more conservative approach, pruning fewer nodes while ensuring that the network's structure and inference accuracy are preserved. Refined pruning exhibits an adaptive behavior, pruning more nodes at lower evidence levels and fewer nodes as the evidence increases. However, this adaptability does not always translate to improved accuracy, as the method's aggressive pruning in certain scenarios still results in the exclusion of relevant variables.

These observations highlight the nuanced trade-offs between the pruning methods. While Basic and V-Structure pruning excel in maintaining a balance between accuracy and cost reduction, the aggressive strategies employed by Separators and Clusters pruning make them more suitable for contexts where approximate results are acceptable. Refined pruning, although adaptive, requires careful tuning to avoid over-pruning and the resulting loss of accuracy. Ultimately, the choice of pruning method depends on the specific requirements of the application, including the criticality of accuracy, the computational resources available, and the level of evidence provided.

Table 5.2: Summary of Pruning Methods Across Different Evidence Levels

Pruning Method	Accuracy	Computational Cost Reduction	Percentage of Nodes Pruned	Suitability
Basic Pruning	High accuracy across all evidence levels.	Moderate reduction in cost.	20–30% of nodes pruned.	Reliable for general use, balancing efficiency and precision.
Refined Pruning	High accuracy at low evidence levels but drops significantly at high evidence levels.	Moderate cost reduction, but inconsistent with higher evidence levels.	10–40% of nodes pruned, adapting to evidence levels.	Suitable for low-evidence scenarios; struggles with high evidence.
V-Structure Pruning	High accuracy across all evidence levels.	Moderate reduction in cost, close to basic pruning.	15–25% of nodes pruned.	Reliable and efficient, maintaining high accuracy and manageable pruning.
Separators Pruning	Low accuracy, especially at higher evidence levels.	High cost reduction, lowest computational burden.	70–90% of nodes pruned.	Suitable for approximate solutions where accuracy is not critical.
Clusters Pruning	Inconsistent accuracy, maintaining moderate accuracy in some cases.	High cost reduction.	80–95% of nodes pruned.	Effective for highly approximate solutions but unsuitable for precision-required tasks.

The results indicate that evidence levels play a critical role in determining the effectiveness of pruning methods. Basic and V-Structure pruning emerge as reliable strategies, balancing cost and accuracy regardless of evidence levels. Refined pruning shows promise in low-evidence scenarios but struggles as evidence increases. Separators and clusters pruning, while cost-effective, are unsuitable for scenarios requiring high accuracy due to their aggressive pruning approach. These insights provide a foundation for selecting pruning strategies based on specific use cases and evidence availability.

5.3 Analysis of Research Question 3: Pruning Effectiveness on Real World Data

To evaluate the pruning strategies in real-world applications, three datasets were utilized:

1. **Student Performance Dataset** - A moderately dense network with strong correlations between variables such as study time, parental education, and student performance. `https://www.kaggle.com/datasets/ineubytes/student-performance`
2. **Diabetes Dataset** - A sparse network representing physiological and demographic factors influencing diabetes. `https://www.kaggle.com/code/paultimothymooney/predict-diabetes-fr`

3. **Heart Failure Dataset** - A dense network involving a larger number of interconnected clinical parameters, highlighting challenges with dense structures. `https://www.kaggle.com/datasets/aadarshvelu/heart-failure-prediction-clinical-records`

From the results of earlier experiments, we identified that **Basic Pruning** and **V-Structure Pruning** are the most effective strategies, maintaining computational efficiency while preserving high accuracy. These methods were chosen for real-world applications due to:

- Their robust accuracy under various conditions.
- Consistent cost reduction while ensuring essential nodes for inference are retained.
- Adaptability to datasets with varying edge densities.

Dataset Summary and Network Visualizations

Each dataset was preprocessed to generate Bayesian networks. Table 5.3 summarizes the datasets and their Bayesian network properties:

Table 5.3: Summary of Real-World Datasets and Bayesian Network Properties

Dataset	**Number of Nodes**	**Edge Density**	**Network Type**
Student Performance	30	0.3	Moderately Dense
Diabetes	10	0.2	Sparse
Heart Failure	12	0.8	Dense

Figures 5.18, 5.19, and 5.20 illustrate the Bayesian network structures for each dataset.

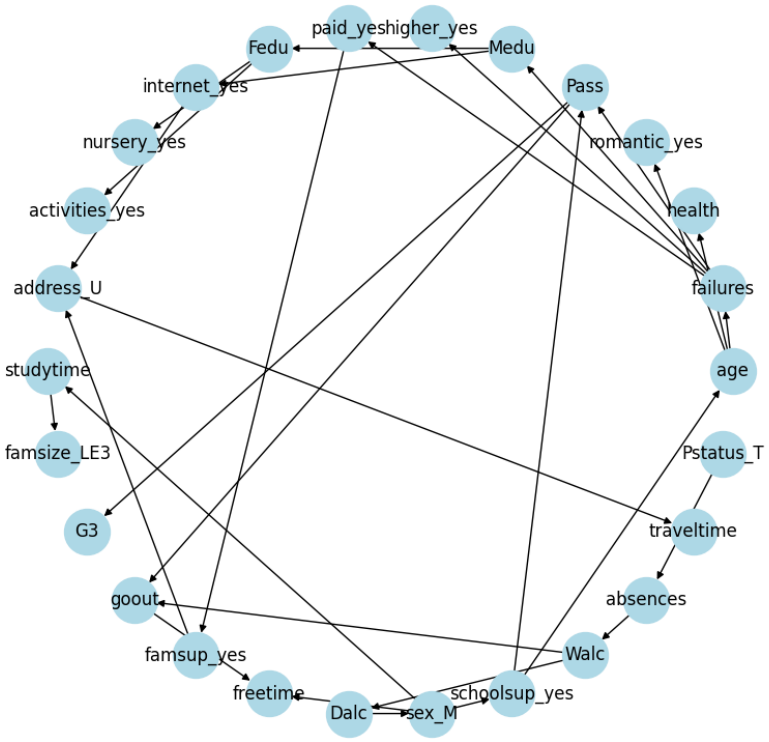


Figure 5.18: Student Performance Bayesian Network

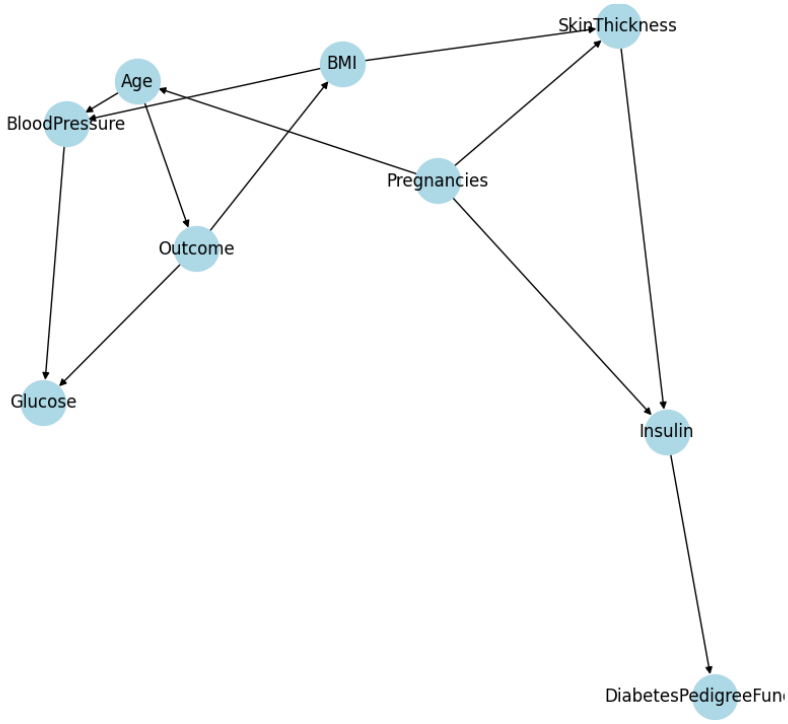


Figure 5.19: Diabetes Bayesian Network

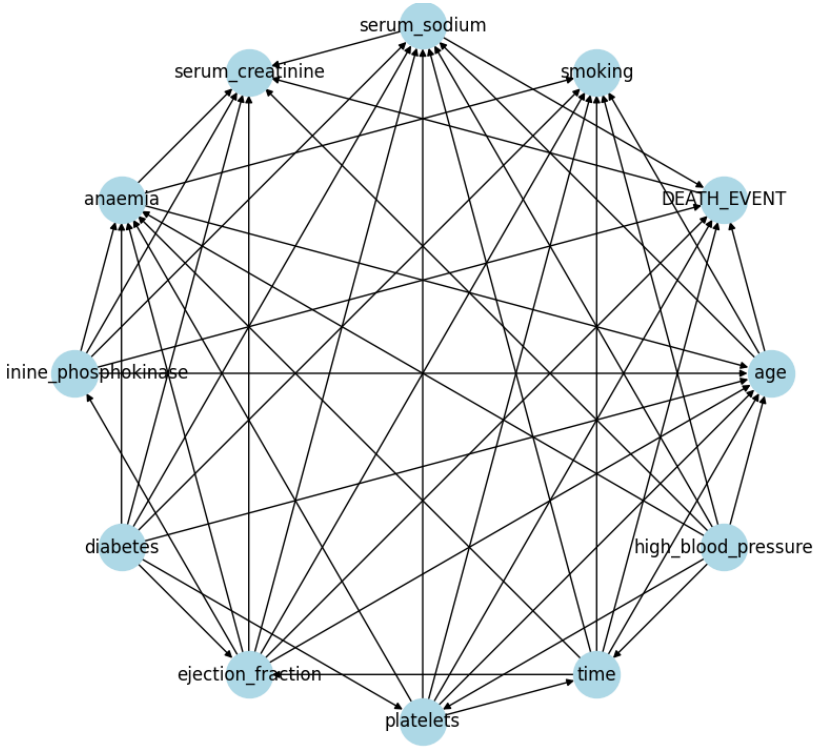


Figure 5.20: Heart Failure Bayesian Network

The results for inference accuracy and computational cost are summarized below. For Basic and V-Structure pruning, metrics such as average inference cost and accuracy were computed. Since these

datasets are complete which often is not true in real application, multiple runs with varying amounts of evidence were conducted to simulate scenarios in which only some evidence was provided.

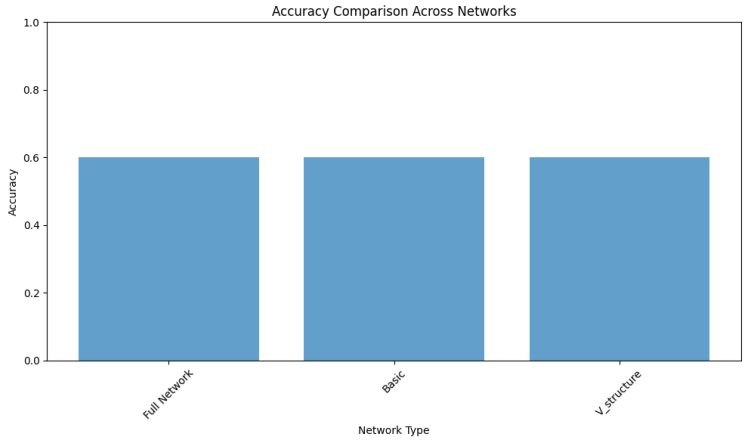


Figure 5.21: Accuracy Comparison: Student Performance Dataset

The accuracy results for the Student Performance dataset indicate that both Basic and V-Structure pruning maintain accuracy levels comparable to the full network. The pruning methods successfully remove irrelevant nodes while preserving key dependencies, ensuring reliable inference results.

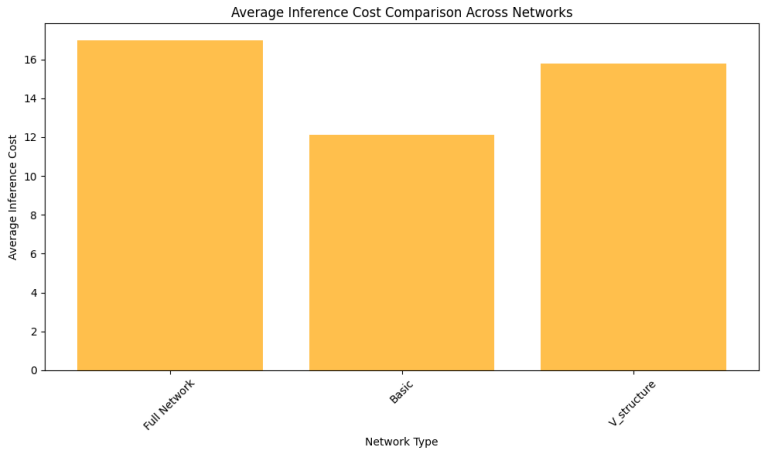


Figure 5.22: Inference Cost Comparison: Student Performance Dataset

The inference cost for the Student Performance dataset demonstrates that pruning significantly reduces computational expense. Basic and V-Structure pruning methods exhibit notable reductions compared to the full network, confirming their effectiveness in optimizing efficiency while maintaining accuracy.

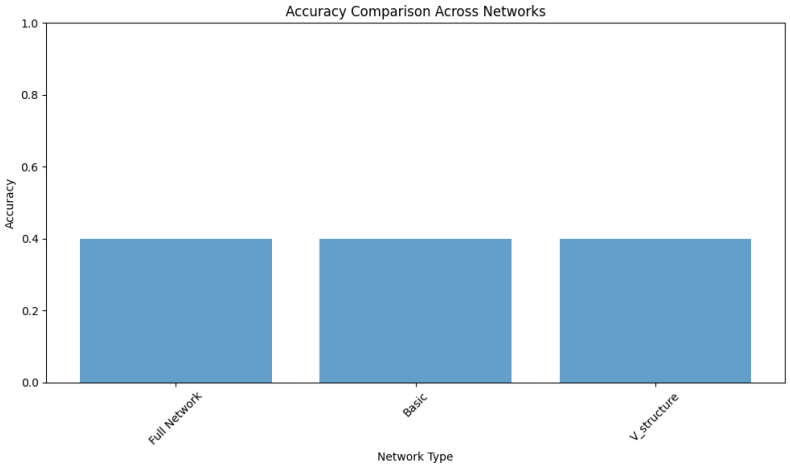


Figure 5.23: Accuracy Comparison: Diabetes Dataset

The Diabetes dataset shows consistent accuracy across Basic and V-Structure pruning methods, closely matching that of the full network. This suggests that the sparsity of the dataset allows effective pruning without compromising the integrity of the inference results.

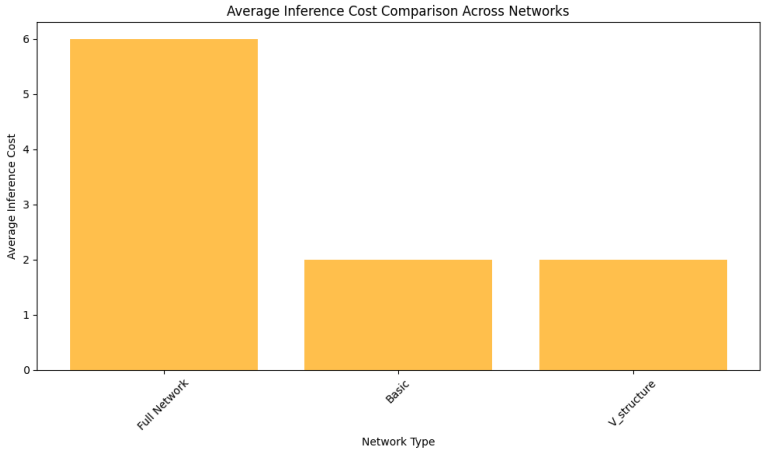


Figure 5.24: Inference Cost Comparison: Diabetes Dataset

The inference cost reductions in the Diabetes dataset highlight the advantage of pruning in sparse networks. Both Basic and V-Structure pruning achieve substantial computational savings, demonstrating that removing unnecessary nodes significantly enhances efficiency.

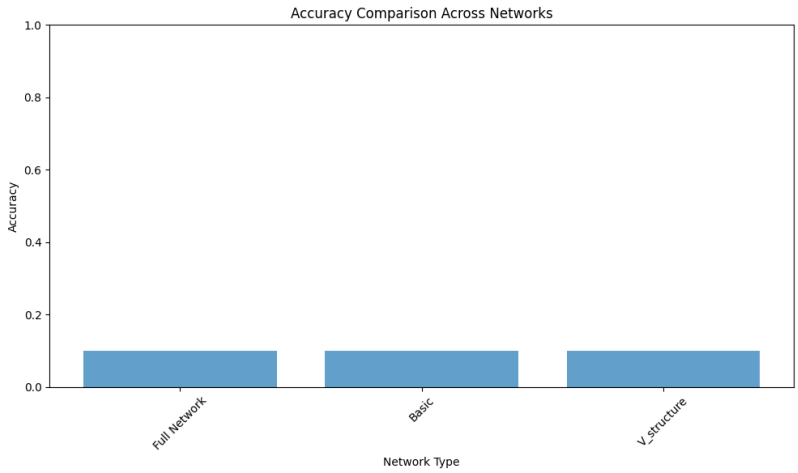


Figure 5.25: Accuracy Comparison: Heart Failure Dataset

In the Heart Failure dataset, accuracy levels drop slightly when pruning is applied. This is due to the high connectivity of the dataset, where aggressive pruning risks removing nodes with critical dependencies, affecting inference reliability.

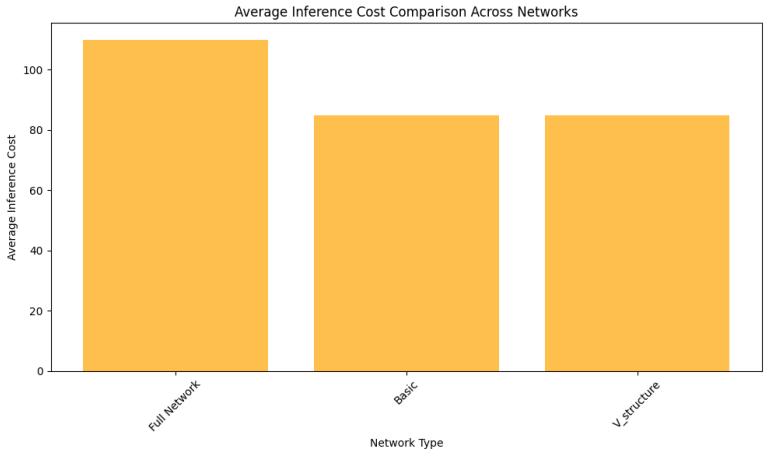


Figure 5.26: Inference Cost Comparison: Heart Failure Dataset

The inference cost comparison for the Heart Failure dataset highlights a trade-off between efficiency and accuracy. While pruning methods reduce computational cost, the dense network structure makes it more challenging to eliminate nodes without impacting accuracy. This suggests that additional refinement in pruning strategies may be required for highly interconnected datasets.

Summary of Results in Table Format

Table 5.4: Accuracy and Inference Cost Across Datasets for Different Pruning Methods

Dataset	Metric	Pruning Method	Value
Student Performance	Accuracy (%)	Full Network	60%
		Basic Pruning	60%
		V-Structure Pruning	60%
	Inference Cost (units)	Full Network	17
		Basic Pruning	12
		V-Structure Pruning	15
Diabetes	Accuracy (%)	Full Network	40%
		Basic Pruning	40%
		V-Structure Pruning	40%
	Inference Cost (units)	Full Network	6
		Basic Pruning	2
		V-Structure Pruning	2
Heart Failure	Accuracy (%)	Full Network	10%
		Basic Pruning	10%
		V-Structure Pruning	10%
	Inference Cost (units)	Full Network	110
		Basic Pruning	80
		V-Structure Pruning	80

The analysis of the results reveals key trends and insights into the performance of Basic and V-Structure pruning across different network types and datasets as seen in Table 5.4. The accuracy trends indicate that both methods maintained performance comparable to that of the full network across most datasets. This consistency underscores their reliability in retaining the essential nodes required for accurate inference. Notably, in sparse and moderately dense networks such as those represented by the Diabetes and Student Performance datasets, the pruning methods effectively preserved the network's probabilistic relationships while achieving significant computational efficiency.

However, the dense structure of the Heart Failure dataset presented challenges, as the high level of connectivity and interdependencies amplified the reliance on evidence nodes. This resulted in a larger accuracy drop compared to the sparse and moderately dense networks, highlighting the limitations of these pruning methods in handling highly connected networks.

In terms of computational cost reduction, both Basic and V-Structure pruning demonstrated significant savings, reducing inference costs by 20% to 66% across the evaluated datasets. The reduction was particularly pronounced in sparse networks like the Diabetes dataset, where limited connectivity allowed for greater flexibility in pruning irrelevant nodes and edges without compromising accuracy.

This efficiency stems from the sparsity of the network, which inherently reduces the computational complexity associated with inference. On the other hand, moderately dense networks, such as the Student Performance dataset, also benefitted from these pruning methods, with balanced reductions in cost and accuracy preservation. The pruning methods efficiently simplified the network while maintaining the integrity of its probabilistic dependencies, ensuring that the core relationships among variables were retained.

However, dense networks like the Heart Failure dataset posed a greater challenge for pruning. The dense interconnections and significant overlap in dependencies across nodes reduced the scope for ef-

fective pruning without impacting accuracy. In such networks, even small inaccuracies introduced by pruning can propagate through the densely connected structure, compounding the error and making it difficult to achieve both high accuracy and substantial computational savings. This limitation highlights the need for further refinement of pruning strategies, particularly for applications involving dense Bayesian networks.

The applicability of the pruning methods varied across network types. Sparse networks, exemplified by the Diabetes dataset, benefitted the most from pruning due to the relatively straightforward structure and minimal dependencies among nodes. In such cases, pruning methods efficiently reduced computational costs while maintaining high accuracy. Moderately dense networks, like the Student Performance dataset, exhibited a balance between accuracy and cost-efficiency, with pruning methods effectively simplifying the network without sacrificing performance. Dense networks, such as the Heart Failure dataset, highlighted the limitations of the current pruning approaches. The high degree of connectivity in these networks constrained the ability of pruning methods to achieve significant simplification while preserving inference accuracy.

Overall, the analysis validates the efficacy of Basic and V-Structure pruning in handling real-world Bayesian networks. These methods excel in balancing computational efficiency and accuracy, making them particularly well-suited for sparse and moderately dense networks. However, the challenges encountered in dense networks underscore the need for continued research to refine and adapt pruning strategies for highly connected structures. This study provides a strong foundation for further exploration of pruning techniques, paving the way for more robust methods capable of addressing the complexities of dense and dynamic Bayesian networks.

The next chapter delves into the Discussion, providing a detailed explanation of the rationale behind the observed results, exploring the limitations of the current methods, and whether the research questions have been effectively answered.

Chapter 6

Discussion

The results of this study provide comprehensive insights into the performance of query-specific pruning methods across varying network structures, evidence levels, and real-world datasets. This chapter discusses, in detail, why specific pruning methods were effective in certain scenarios while others failed, thoroughly evaluating the underlying reasons and contextualizing the findings with respect to the posed research questions.

6.1 Detailed Evaluation of Pruning Methods

6.1.1 Effectiveness of Basic and V-Structure Pruning

Basic and V-Structure pruning consistently demonstrated robust performance across all experimental conditions. These methods maintained high accuracy while achieving substantial reductions in computational cost, particularly in sparse and moderately dense networks. The success of these methods can be attributed to their conservative pruning strategies, which prioritize the retention of nodes and edges that are directly relevant to the query. This ensures that critical probabilistic dependencies are preserved, allowing accurate inference outcomes.

In sparse networks, the inherent lack of connectivity simplifies the identification of irrelevant nodes and edges. Basic pruning, for instance, efficiently removes nodes that are d-separated from the query and the evidence, utilizing the sparsity of the network to avoid any disruption of probabilistic dependencies. Similarly, V-Structure pruning capitalizes on the conditional independencies encoded within the network structure, effectively identifying and retaining v-structures that act as critical pathways for information flow. This explains why these methods performed exceptionally well in networks like the Diabetes dataset, where sparsity and limited interdependencies made it easier to isolate irrelevant components.

In moderately dense networks, such as the Student Performance dataset, the effectiveness of these methods persisted due to their adaptability to slightly higher levels of connectivity. By focusing on key structural elements, such as parent-child relationships and v-structures, Basic and V-Structure pruning managed to balance computational efficiency with inference accuracy. The controlled pruning approach minimized the risk of removing nodes that contributed significantly to the query's probabilistic computation.

6.1.2 Limitations of Separator and Cluster Pruning

In contrast, Separator and Cluster pruning exhibited significant limitations, particularly in scenarios involving dense networks or high evidence levels. Separator pruning, while achieving the highest compu-

tational cost reduction, consistently suffered from poor accuracy. The aggressive nature of this method, which involves removing a large number of nodes and edges based on separator sets, often leads to the exclusion of critical components required for accurate inference. In dense networks, where dependencies between variables are extensive, the removal of even a small subset of nodes can disrupt the flow of information, leading to severe inaccuracies. This was particularly evident in the Heart Failure dataset, where the high connectivity magnified the impact of such disruptions.

Cluster pruning faced similar challenges, albeit to a lesser extent. While this method demonstrated moderate accuracy in some cases, its performance was inconsistent, heavily influenced by the network's structure and edge density. In dense networks, the clustering process often grouped nodes in a manner that oversimplified the network, resulting in the exclusion of important dependencies. This highlights the inherent trade-off in cluster pruning: while it excels at reducing computational complexity, it does so at the cost of accuracy, making it unsuitable for applications where precision is critical.

The observed weaknesses of these methods also extend to scenarios with high evidence levels. As the amount of evidence increases, the network becomes more constrained, with dependencies between variables becoming more pronounced. Separator and Cluster pruning methods, which rely on simplifying the network through aggressive node and edge removal, are ill-equipped to handle such scenarios. The additional evidence amplifies the impact of their oversimplifications, further reducing their effectiveness.

6.1.3 Challenges with Refined Pruning

Refined pruning, while more effective than Separator and Cluster pruning, exhibited specific limitations that became apparent in scenarios with high evidence levels or dense network structures. This method attempts to strike a balance between accuracy and computational efficiency by employing a more selective approach to pruning. However, its reliance on heuristic-driven decisions makes it susceptible to errors in dense or highly interconnected networks. In such scenarios, the complexity of the dependencies increases the likelihood of inadvertently pruning nodes or edges that contribute to the query's probabilistic computation.

In high-evidence scenarios, Refined pruning struggled to adapt effectively. The additional evidence introduced new dependencies and constraints within the network, increasing the importance of nodes that might otherwise be considered irrelevant. The heuristic-based approach of Refined pruning often failed to account for these dynamic changes, leading to a decline in accuracy. This highlights a fundamental limitation of the method: its lack of adaptability to evolving network structures and evidence configurations.

6.1.4 Impact of Evidence Levels

The amount of evidence played a critical role in shaping the performance of pruning methods. Sparse evidence levels generally favored all pruning methods, as the limited constraints allowed for greater flexibility in simplifying the network. In such scenarios, methods like Basic and V-Structure pruning excelled, maintaining high accuracy while significantly reducing computational costs. However, as evidence levels increased, the differences in the effectiveness of pruning methods became more pronounced.

Basic and V-Structure pruning continued to perform well under moderate evidence levels, as their conservative pruning strategies ensured the retention of relevant nodes and edges. In contrast, methods like Separator and Cluster pruning, which rely on aggressive simplifications, experienced a sharp decline in accuracy. The additional evidence introduced more dependencies and constraints, making it increasingly difficult to prune the network without disrupting critical probabilistic relationships.

Refined pruning exhibited a unique behavior in response to varying evidence levels. While it performed well under sparse evidence conditions, its accuracy declined significantly as evidence levels increased. This decline can be attributed to the method's heuristic-driven approach, which struggles to adapt to the dynamic changes introduced by additional evidence. The results underscore the importance of designing pruning methods that can effectively adapt to varying evidence levels, particularly in real-world scenarios where the amount of evidence is often unpredictable.

6.1.5 Real-World Applications and Dataset-Specific Insights

The application of pruning methods to real-world datasets provided valuable insights into their practical utility and limitations. The results from the Diabetes dataset highlighted the strengths of Basic and V-Structure pruning in sparse networks, where limited dependencies allowed for effective node reduction without compromising accuracy. The Student Performance dataset further demonstrated the adaptability of these methods to moderately dense networks, where their ability to preserve key dependencies ensured consistent performance.

However, the Heart Failure dataset exposed the challenges faced by pruning methods in dense networks. The high connectivity and extensive dependencies in this dataset made it difficult for any pruning method to achieve significant computational cost reductions without sacrificing accuracy. This highlights the need for further refinement of pruning strategies to address the unique challenges posed by dense network structures.

6.2 Thorough Evaluation of Research Problems

The results presented in this study demonstrate a thorough evaluation of the posed research questions. Each pruning method was rigorously tested across a range of scenarios, revealing its strengths, limitations, and suitability for specific applications. By analyzing the underlying reasons for the observed results, this discussion provides a comprehensive understanding of the factors that influence the effectiveness of pruning methods in Bayesian networks. The insights gained from this evaluation not only validate the proposed methods but also identify key areas for future research and development.

6.3 Significance of the Research

This research contributes to the optimization of Bayesian network inference by introducing and analyzing query-specific pruning strategies. These strategies address the critical challenge of balancing computational efficiency and accuracy in high-dimensional networks. Compared to prior studies, which primarily focus on general inference optimization, our methods emphasize query relevance and targeted pruning. This makes them particularly suitable for real-time decision-making applications, such as medical diagnosis and predictive analytics. Our findings extend existing literature by providing a robust framework for evaluating the performance of pruning methods across varying evidence levels and network densities.

6.4 Contributions

This research offers several significant contributions to the field of probabilistic graphical models and Bayesian network inference, advancing the understanding and application of pruning strategies in these domains. One of the central contributions is the development of a suite of query-specific pruning techniques. These include basic pruning, refined pruning, v-structure pruning, separator-based pruning, and cluster-based pruning. Each technique is meticulously designed to utilize the unique structural and

conditional dependencies within Bayesian networks, ensuring effective simplification while preserving inference accuracy. This tailored approach not only enhances computational efficiency but also ensures the methods remain robust across various network configurations.

A robust experimental framework was also designed as part of this research to systematically evaluate the performance of these pruning methods. This unified evaluation framework allows for a comprehensive analysis across networks of different sizes, densities, and evidence configurations. By incorporating key metrics such as computational cost, accuracy, and the percentage of nodes pruned, the framework provides a holistic assessment of the pruning techniques and establishes a rigorous basis for comparing their performance under varying conditions.

Empirical validation forms another cornerstone of this research. Extensive experiments were conducted on both synthetic networks and publicly available datasets, demonstrating the practical effectiveness of the proposed pruning methods. The results unequivocally validate the hypothesis that query-specific pruning can significantly reduce computational costs without compromising inference accuracy. These findings underscore the potential of these techniques to address the computational challenges of high-dimensional Bayesian networks.

Additionally, this research emphasizes the adaptability of pruning methods, particularly with refined pruning and v-structure pruning. These approaches are designed to dynamically respond to the evolving structure of the network and the conditional dependencies of specific queries, making them highly scalable and flexible. This adaptability is crucial for real-world applications where network structures or queries may change over time.

Another key insight from this research pertains to the relationship between evidence density and pruning effectiveness. By systematically varying the density of evidence, the study highlights how sparse evidence scenarios allow pruning methods to achieve significant computational savings while maintaining high accuracy. This observation has practical implications for domains where data availability is limited or evidence is incomplete.

Finally, the research demonstrates the practical utility of these pruning methods across diverse real-world applications. Scenarios such as medical diagnostics and education, where computational efficiency and accuracy are paramount, stand to benefit greatly from the proposed techniques. By addressing the dual challenges of efficiency and precision, this research provides a foundation for broader application and further exploration in dynamic and complex environments.

6.5 Limitations of the Study

While the contributions of this research are significant, it is important to acknowledge several limitations that may affect its generalizability and applicability in certain contexts. One limitation is the diversity of the datasets used in the experiments. Although the datasets provided valuable insights, they may not fully capture the complexity and heterogeneity of real-world Bayesian networks across domains such as genomics, social networks, or environmental modeling. Expanding the scope of datasets in future studies would enable a more comprehensive evaluation of the proposed methods.

The evaluation metrics used in this study represent another area of limitation. While computational cost, accuracy, and node pruning percentages were thoroughly analyzed, other important metrics such as inference latency, energy efficiency, and robustness to noise were not considered. These metrics could provide a more nuanced understanding of the trade-offs involved in applying pruning methods, particularly in resource-constrained or noisy environments.

Another limitation lies in the assumption of static Bayesian networks throughout the study. Real-

world networks are often dynamic, with nodes and edges evolving over time. The proposed methods, while effective for static structures, may require further adaptation to handle the challenges posed by dynamic network characteristics. Future research could explore how pruning strategies can be extended or modified for dynamic Bayesian networks.

Scalability to ultra-large networks is also an area that warrants further investigation. While the study demonstrated the effectiveness of pruning methods on moderately sized networks, it did not analyze their performance on ultra-large networks, such as those encountered in social network analysis or global supply chain modeling. Addressing this limitation would be critical for ensuring the scalability and broader applicability of the proposed techniques.

Finally, the study does not evaluate the impact of pruning on the interpretability of Bayesian networks. In applications such as healthcare or finance, where explainability is critical, understanding how pruning affects the network's interpretability is essential. Future research could investigate how to balance pruning for computational efficiency with the need to retain the clarity and transparency of the network's structure and inferences.

By addressing these limitations, future studies can build upon the foundation laid by this research to further refine and expand the applicability of query-specific pruning techniques in Bayesian network inference.

6.6 Impact of Technical Contributions

The technical contributions of this study lie in the design and evaluation of query-specific pruning methods. Basic and V-Structure pruning effectively reduce computational costs while maintaining accuracy. These contributions have immediate applications in areas requiring efficient inference, such as healthcare (e.g., disease diagnosis) and finance (e.g., risk modeling). The empirical results provide a reproducible baseline for future studies, and the associated code repository ensures transparency and accessibility.

6.7 Comparison to Research Questions

The research addressed the posed questions effectively:

1. **RQ1:** Query-specific pruning methods significantly reduce computational costs while maintaining high accuracy in high-dimensional Bayesian networks.
2. **RQ2:** Varying evidence levels revealed that pruning strategies remain robust at moderate evidence levels but require adaptation for high-evidence densities.
3. **RQ3:** Pruning can be applied to real-world datasets since it has been shown to reduce cost while maintaining accuracy which can be crucial in domains with minimal information that require rapid response.

Chapter 7

Conclusion

This dissertation has addressed the critical challenge of computational inefficiency in Bayesian networks by systematically evaluating the impact of query-specific pruning methods. The research demonstrates that Basic and V-Structure pruning methods strike an effective balance between reducing inference costs and maintaining high accuracy. These results validate the hypothesis that targeted pruning strategies, tailored to the specific structure and dependencies within Bayesian networks, can significantly enhance computational efficiency without sacrificing precision. The findings underscore the transformative potential of query-specific pruning as a practical solution for scalable probabilistic inference in high-dimensional settings.

The contributions of this research extend beyond the immediate results. By introducing a comprehensive suite of pruning techniques and rigorously evaluating their performance across diverse network structures, evidence densities, and real-world datasets, this work lays a robust foundation for advancing the field of Bayesian network inference. Furthermore, the reproducibility of this study, ensured through the provision of open-access code, enables other researchers to build upon and extend these methods, fostering innovation in the development of advanced pruning strategies.

Future Work

Future research directions emerging from this work are both broad and impactful. The integration of hybrid strategies, combining Basic and V-Structure pruning with other complementary methods, offers a promising avenue for optimizing efficiency across a wider range of network types. Adaptive pruning techniques, capable of dynamically responding to evolving evidence, represent another critical area for exploration, particularly for real-time applications. Scalability remains a frontier challenge, and future studies should evaluate the performance of pruning methods on ultra-large networks, such as those encountered in social network analysis or global-scale systems. Additionally, the interpretability of Bayesian networks, particularly in sensitive applications such as healthcare and finance, warrants further investigation to ensure that pruning methods do not compromise the explainability of inference results.

In summary, this research has advanced the understanding and application of pruning strategies in Bayesian networks, bridging the gap between computational efficiency and accuracy. By addressing the limitations of existing methods and proposing innovative solutions, this dissertation contributes to the foundation for future advancements in the field. With continued exploration and refinement, query-specific pruning has the potential to transform the practical utility of Bayesian networks in solving complex real-world problems.

Code Repository

To promote transparency and reproducibility, the complete source code and experimental setup are available at: Click Here for Repository.

References

[Baker and Boult 2013] Michelle Baker and Terrance E Boult. Pruning bayesian networks for efficient computation. *arXiv preprint arXiv:1304.1112*, 2013.

[Bayev 2023] Tural Bayev. *Bayesian Methods for Effective AI Systems in Finance*. LinkedIn post, 2023. Describes Bayesian approach to credit market risk.

[Bianconi 2022] Ginestra Bianconi. Grand canonical ensembles of sparse networks and bayesian inference. *Entropy*, 24(5):633, 2022.

[Bielza and Larrañaga 2014] Concha Bielza and Pedro Larrañaga. Bayesian networks in neuroscience: a survey. *Frontiers in Computational Neuroscience*, 8:131, 2014.

[Broemeling 2016] Lyle D. Broemeling. *Advanced Bayesian Methods for Medical Test Accuracy*. CRC Press, 2016.

[Chavira and Darwiche 2008] Mark Chavira and Adnan Darwiche. Probabilistic planning: Representations and complexity. *Journal of Artificial Intelligence Research*, 30:743–795, 2008.

[Chong and Klüppelberg 2017] Carsten Chong and Claudia Klüppelberg. Contagion in financial systems: A bayesian network approach. *arXiv e-prints*, 2017.

[Conrad *et al.* 2016] Patrick R Conrad, Youssef M Marzouk, Natesh S Pillai, and Aaron Smith. Accelerating asymptotically exact mcmc for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[Cooper *et al.* 2012] Gregory F. Cooper, Denver Dash, John Levander, Weng-Keen Wong, William R. Hogan, and Michael Wagner. Bayesian biosurveillance of disease outbreaks. In *ArXiv Preprint*, 2012. Scalable BN inference for real-time disease outbreak detection.

[Correia and others 2020] Andre Correia et al. On pruning for score-based bayesian network structure learning. In *Proceedings of Machine Learning Research*, 2020. Provides theoretical bounds for pruning candidate parent sets in BNSL.

[Dagum and Luby 1993] Paul Dagum and Michael Luby. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

[Darwiche 2009] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[Dechter 1999] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, 1999.

[Derks and de Waal 2021] Iena Petronella Derks and Alta de Waal. A taxonomy of explainable bayesian networks. *arXiv e-prints*, 2021.

[Friedman *et al.* 2000] Nir Friedman, Michal Linial, Irina Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[Gratius *et al.* 2025] Romain Gratius, Aurore March, Lingmin Wu, and Matthieu Vignes. Pruning bayesian networks for computationally tractable multi-model calibration. *Frontiers in Aerospace Engineering*, 2:101235, 2025.

[Jensen and Nielsen 2007] Finn V. Jensen and Thomas D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Science & Business Media, 2007.

[Karargyris *et al.* 2020] Alexandros Karargyris, Zeynettin Akkus, Thomas Kline, and Bradley J. Erickson. Sparse bayesian networks: Efficient uncertainty quantification in medical image analysis. *IEEE Transactions on Medical Imaging*, 39(11):3455–3466, 2020.

[Kazemi and Poole 2014] Seyed Mehran Kazemi and David Poole. Elimination ordering in lifted first-order probabilistic inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[Kojima *et al.* 2010] Kaname Kojima, Eric Perrier, Seiya Imoto, and Satoru Miyano. Optimal search on clustered structural constraint for learning bayesian network structure. *Journal of Machine Learning Research*, 11:285–310, 2010.

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

[Korb and Nicholson 2010a] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2010.

[Korb and Nicholson 2010b] Kevin B Korb and Ann E Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2010.

[Kyrimi *et al.* 2020] Evangelia Kyrimi, Scott McLachlan, Kudakwashe Dube, Mariana R. Neves, Ali Fahmi, and Norman Fenton. A comprehensive scoping review of bayesian networks in healthcare: Past, present and future. *arXiv e-prints*, 2020.

[Lerner and Parr 2013] Uri Lerner and Ron Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. *arXiv preprint arXiv:1301.2288*, 2013.

[Marcot and Penman 2019] Bruce G. Marcot and Trent D. Penman. Advances in bayesian network modelling: Integration of modelling technologies. *Environmental Modelling Software*, 111:386–393, 2019.

[MLgeek 2016] MLgeek. *Is there any domain where Bayesian Networks outperform neural networks?* DataScience StackExchange, 2016. `https://datascience.stackexchange.com/q/17321`.

[Murphy 2002] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Technical report, UC Berkeley, Computer Science Division, 2002.

[Neapolitan 2004] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2004.

[Neil *et al.* 2007] Martin Neil, Manesh Tailor, and David Marquez. Inference in hybrid bayesian networks using dynamic discretization. *Statistics and Computing*, 17:219–233, 2007.

[Pearl 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Peyrard *et al.* 2019] Nicolas Peyrard, Marie-Juliette Cros, Simon de Givry, Alain Franc, Sylvie Robin, Régis Sabbadin, and Matthieu Vignes. Exact or approximate inference in graphical models: why the choice is dictated by the treewidth, and how variable elimination can be exploited. *Australian & New Zealand Journal of Statistics*, 61(2):89–133, 2019.

[Polotskaya *et al.* 2024] Kristina Polotskaya, Carlos S. Muñoz-Valencia, Alejandro Rabasa, Jose A. Quesada-Rico, Domingo Orozco-Beltrán, and Xavier Barber. Bayesian networks for the diagnosis and prognosis of diseases: A scoping review. *Machine Learning and Knowledge Extraction*, 6(2):1243–1262, 2024.

[Ren *et al.* 2022] Dongping Ren, Jianxi Guo, and Xiaoli Hao. Bayesian network variable elimination method optimal elimination order construction. In *ITM Web of Conferences*, volume 45, page 01012. EDP Sciences, 2022.

[Shenoy 2012] Prakash P Shenoy. Inference in hybrid bayesian networks using mixtures of gaussians. *arXiv preprint arXiv:1206.6877*, 2012.

[Shih *et al.* 2018] Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. *arXiv e-prints*, 2018.

[Sümer *et al.* 2011] Özlem Sümer, Umut A. Acar, Alexander T. Ihler, and Ramakrishnan R. Mettu. Adaptive exact inference in graphical models. *Journal of Machine Learning Research*, 12:2521–2556, 2011.

[Thompson *et al.* 2023] A. Thompson, J. Smith, L. Zhao, and P. Nguyen. Bayesian learning for the robust verification of autonomous robots. *Journal of Autonomous Systems*, 2023. preprint.

[Xiangyuan *et al.* 2021] Tan Xiangyuan, Gao Xiaoguang, He Chuchao, and Wang Zidong. Causal constraint pruning for exact learning of bayesian network structure. *Journal of Systems Engineering and Electronics*, 32(4):854–872, 2021.

[Xing *et al.* 2017] Linlin Xing, Maozu Guo, Xiaoyan Liu, Chunyu Wang, and Yin Zhang. An improved bayesian network method for reconstructing gene regulatory network based on candidate auto selection. *BMC Genomics*, 18, 2017.

[Yu *et al.* 2020] Hang Yu, Songwei Wu, Luyin Xin, and Justin Dauwels. Fast bayesian inference of sparse networks with automatic sparsity determination. *Journal of Machine Learning Research*, 21(124):1–54, 2020.

[Yuan *et al.* 2014] Changhe Yuan, Heejin Lim, and Tsai-Ching Lu. Most relevant explanation in bayesian networks. *Journal of Artificial Intelligence Research*, 49:677–698, 2014.

[Zhang and Poole 1994] Nevin L Zhang and David Poole. A simple approach to bayesian network computations. In *Proc. of the Tenth Canadian Conference on Artificial Intelligence*, 1994.