From inference to influence: applying causal game theory to complex security environments

Vonk, M.C.

Chapter 5

Optimization of Causal Interventions

Beyond establishing the methodological foundations of causality and game theory, this dissertation aims to equip practitioners with the necessary tools to engage with key aspects of causal reasoning. In particular, the focus is on the optimization of causal interventions, examining how they can be computed efficiently while maintaining high accuracy.

As associational queries can already be computationally demanding, causal interventions often require the computation of multiple associational queries and are therefore even more demanding. Optimizing over multiple causal interventions constitutes a particularly burdensome computational task.

This chapter provides a computationally efficient methodology to optimize over multiple causal interventions in Markovian Bayesian networks. This chapter thereby addresses RQ2: *How can optimal causal interventions be computed with high accuracy while ensuring computational efficiency?* The content of this chapter closely aligns with two peer-reviewed conference papers [249, 253].

5.1 Introduction

Before presenting the approach to optimizing causal interventions, the problem is first formally defined in terms of the concepts introduced in Chapter 3. Additionally, existing methodologies are reviewed for addressing this problem, highlighting their

strengths and limitations.

Similar to how influence diagrams make a separation between decision, utility and chance nodes, the variables in a causal Bayesian network $\boldsymbol{V}$ can be further dvidided into intervenable variables $\boldsymbol{D}$, context variables $\boldsymbol{X}$, and outcome variable Y.[1] The goal is then to optimize the interventions among intervenable variables that minimize the expected value of the outcome variable in the associated interventional distribution:

$$\boldsymbol{V}_K^*, \boldsymbol{v}_K^* = \underset{\boldsymbol{V}_K \subset \boldsymbol{D}, \boldsymbol{v}_K \in \Omega_{\boldsymbol{V}_K}}{\arg\min} \mathbb{E}[Y \mid do(\boldsymbol{V}_K = \boldsymbol{v}_K))]. \tag{5.1}$$

This is called the offline *causal global optimization* problem [4] on Markovian Bayesian networks with continuous as well as discrete variables. The problem can be separated into two parts. First, there is a necessity to compute the inference queries $\mathbb{E}[Y \mid do(\boldsymbol{V_K} = \boldsymbol{v_K}))]$ efficiently in terms of accuracy and in terms of computational cost. Second, this computation procedure should then be embedded in an optimization framework that can formulate an answer to the causal global optimization problem.

Computing interventional queries requires applying the adjustment formula from Chapter 3. When the set of adjusted variables $\boldsymbol{V}_J \subset \boldsymbol{V}$ is non-empty and continuous, the interventional distribution is given by:

$$p(Y \mid do(\boldsymbol{V}_K = \boldsymbol{v}_K)) = \int_{\Omega_{\boldsymbol{V}_J}} p(Y \mid \boldsymbol{v}_K, \boldsymbol{v}_J) p(\boldsymbol{v}_J) d\boldsymbol{v}_J. \tag{5.2}$$

Even though many Bayesian network applications require the accommodation of such continuous variables [65, 162], state-of-the-art methods for continuous or hybrid (combination of discrete and continuous) Bayesian network inference are still underdeveloped. Algorithms have been developed to conduct inference on hybrid Bayesian networks when a conditional Gaussian distribution among the variables is assumed [127]. However, assuming the parametric form of the distribution is costly, which is why much research has been dedicated to approximation by either discretizing Bayesian networks [31, 169, 168] or by approximating the distribution of the variables in the Bayesian network with a linear combination of exponentials [206] or polynomials [216], which both allow inference.

Discretization of the continuous variables enables the use of established discrete Bayesian network inference methods. Variable elimination and belief propagation are

[1] While this classification closely resembles the structure of influence diagrams, the framework considered here is technically a causal Bayesian network. This distinction arises from the assumption that decision rules are fixed, meaning the decision-maker is limited to performing hard interventions on variables rather than selecting decision rules.

well-developed exact inference methods for discrete Bayesian networks that exploit the structure of the Bayesian network to substantially reduce the computational burden. Nevertheless, even with these effective algorithms, the computational cost increases exponentially as the number of parent nodes within the network grows. Therefore, researchers often employ approximate methods such as sampling or variational inference approaches for more complex Bayesian networks. These methods are summarized by Koller and Friedman [127].

While discretization allows the use of discrete Bayesian network inference algorithms, it may lead to a loss of information, resulting in a lower accuracy of the inference query. At the same time, the computational cost of inference depends heavily on the number and positioning of bins that result from the discretization process. Or, as stated in Koller and Friedman [127], "discretization provides a trade-off between the accuracy of the approximation and cost of computation."

To address the computational challenges of Bayesian network inference after discretization, knowledge compilation [59] can be used. In knowledge compilation, information (such as the probability distribution given by a Bayesian network) is translated without loss into a format that can be queried efficiently. One of the motivations behind knowledge compilation is that by first performing a potentially computationally expensive 'compilation' step, which takes exponential time in the worst case, afterwards the result of many queries (such as inference queries) can be computed quickly. While compilation of the Bayesian network is a heuristic compression method, guided by practical effectiveness rather than formal guarantees of reduced inference complexity, it often proves much faster in practice. This property is particularly advantageous when inference is embedded within an optimization framework, where numerous inference queries must be evaluated efficiently. Specifically, the compilation of Bayesian networks into *binary decision diagrams* (BDDs) [40] is considered, as they have been shown to perform well in the context of Bayesian network inference [56].

The entire methodology is first discussed in Section 5.2, including discretization, knowledge compilation methods, and optimization. As discretization is associated with a loss of information, experiments are run that provide insight into the trade-off between the accuracy of the inference approximation/discretization and the cost of its computation. Further experiments were conducted embedding this framework within various optimization heuristics, in order to evaluate the performance of different algorithms when optimizing over causal interventions. The experimental setup, including the evaluation metrics and considered Bayesian networks for both experiments, is outlined in Section 5.3. The results are presented in Section 5.4, followed by concluding

remarks in Section 5.5.

5.2 Methodology

In this section, a methodology is proposed to tackle the offline causal global optimization problem on hybrid Bayesian networks. The first step is to encode discretized versions of the Bayesian networks as binary decision diagrams. These binary decision diagrams are subsequently subjected to heuristic optimization algorithms that use these efficient encodings to optimize over interventional queries.

More specifically, the entire methodology, including evaluation metrics, is specified in Figure 5.1.[2] To allow a different number and positioning of discretized bins, different types of discretization methods are considered: two unsupervised approaches,

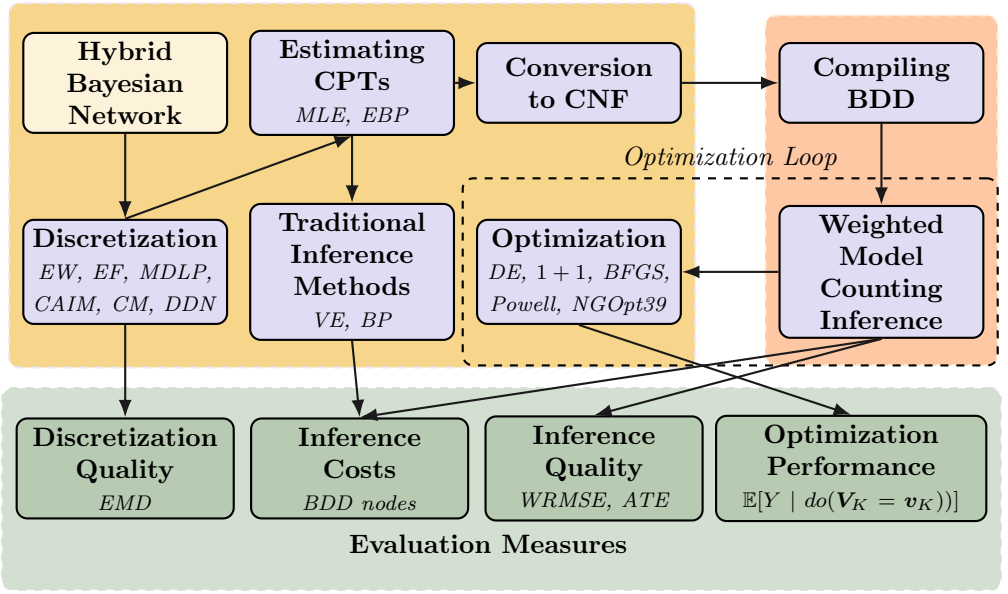


Figure 5.1: Methodology for offline causal global optimization on hybrid Bayesian networks. Python-based components (yellow) handle discretization, estimating CPTs, traditional inference, conversion to CNF formulas, and optimization. C-based components (orange) compile and query BDDs for efficient interventional inference. Evaluation measures (green) assess discretization quality, inference cost, inference accuracy, and optimization performance.

[2]Implementation of the methodology is available on `https://github.com/sebastiaanbrand/bn-dd`.

equal width (EW) and *equal frequency* (EF) binning, as well as the supervised *minimum description length principle* (MDLP) binning method, *class-attribute interdependence maximization* (CAIM) discretization [133], *ChiMerge* (CM) discretization [124] and *dynamic discretization* (DDN) [168]. Conditional probability tables of discretized Bayesian networks are inferred via *maximum likelihood estimation* (MLE) as well as via a *Bayesian method with adjusted empirical Bayes priors* (EBP). Subsequently, the discretized Bayesian networks are encoded as *conjunctive normal form* (CNF) formulas, which are then compiled into BDDs. Inference with BDDs first requires performing a computationally expensive compilation step, after which computing inference queries can be done in time linear in the size of the BDD [57]. As an empirical demonstration, a range of black-box optimization algorithms from the literature [19] is applied,[3] including *random search* as a baseline, *differential evolution* [235] (DE), *(1+1) evolutionary strategy* [32], *BFGS* [39], *Powell's method* [186] and the *NGOpt39 optimizer* [160].

In applying the methodology to a variety of hybrid Bayesian networks, several findings are reported. First, the computation time of BDD-based inference is compared with traditional inference methods such as *variable elimination* (VP) and *belief propagation* (BP). Second, the trade-offs between the quality of the discretization/inference and the computational cost of BDD-based inference algorithms is studied. Finally, the performance of several heuristic optimization algorithms in addressing the offline causal global optimization problem within this BDD-based inference framework is reported.

To assess the trade-off between the quality of discretization/inference and the cost of knowledge compilation, a concept known in multi-objective optimization as the *Pareto front* is used to visualize the results of various considered approaches. A Pareto front represents the set of non-dominated solutions where improving one objective would result in degrading another. The evaluation involves measuring discretization quality in terms of the *earth mover's distance* (EMD) and quantifying knowledge compilation cost by considering the number of nodes in the BDD. Additionally, for non-causal networks, the quality of conditional queries (if ground truth is available) is measured using the *weighted root mean squared error* (WRMSE). For causal Bayesian networks, such additional quality evaluation is done via the *percentage error* of the average treatment effect (PE ATE). Finally, the optimization is evaluated by the objective function (interventional query) with respect to the number of evaluations.

[3]The experiments use implementations from the Nevergrad library, an open-source platform for optimization available at `https://github.com/facebookresearch/nevergrad`.

While the datasets to which the methodology is applied, along with the evaluation measures, are further discussed in Section 5.3, the discretization, parameter learning, BDD encoding, and optimization process are discussed in more detail in this section. A general limitation in this area is the lack of benchmark datasets with known ground truths that follow general continuous distributions. To address this, we use synthetic data generated from continuous distributions with known ground truths, as well as observational data with corresponding experimental counterparts, both of which enable meaningful validation.

5.2.1 Discretization and Parameter Learning Methods

The discretization process serves to partition the state space Ω_{X_i} of a continuous random variable X_i into disjoint bins $\{B_j \mid j = 1, \ldots, m\}$ such that $\bigcup_j B_j = \Omega_{X_i}$. Every bin B_j is associated with a real number $g(B_j)$ denoting the value of the interval. In real-world applications, the state space of the random variable is unknown but is based on the sample data. The value associated with each bin B_j corresponds to the sample mean of the samples that are included in the bins, $\frac{1}{|B_j|} \sum_{x_i \in B_j} x_i$, in which $|B_j|$ denotes the number of $x_i \in B_j$.

The equal width discretization method partitions the state spaces Ω_{X_i} into bins of equal width. The equal frequency discretization approach divides the samples into quantiles. Both are unsupervised methods and require a parameter specifying the number of bins into which the original state space should be partitioned.

In addition to these two unsupervised discretization methods, four supervised discretization methods are used. First, the entropy error-based approach, dynamic discretization [168] is considered,[4] specifically developed for Bayesian network inference. Second, the minimum description length principle discretization [71] is employed, which iterates through potential cut-points recursively to minimize information entropy with respect to a chosen target variable. Third, ChiMerge [124] is applied, a discretization technique that continuously merges fine intervals based on the χ^2 statistic. Fourth, class-attribute interdependence maximization [133] is used, which discretizes the continuous variables intending to maximize interdependency with the target variable [49]. The latter three supervised discretization methods have been chosen because they performed well on a variety of discretization tasks [77].

Discretization of a continuous Bayesian network is followed by parameter learning, which involves the estimation of the CPTs. In this section, the maximum likelihood

[4]The implementation available at `https://github.com/PCiunkiewicz/dynamic-discretization` is used, adopting the parameter settings deemed most optimal by the implementer.

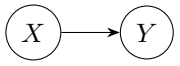
estimate and the Bayesian method with adjusted empirical Bayes type 2 maximum likelihood priors [121, 85] are considered. In the latter, the prior is initially estimated through MLE but refined by substituting 0 probability values with a minimal value (0.0001). This adjusted prior is subsequently used to infer the posterior CPTs with the data. While the maximum likelihood estimates are sufficient to conduct inference on non-causal datasets, the causal datasets require the Bayesian approach to prevent any violations of the positivity assumption as described in Chapter 3 [267]. The differences in results between both methods are discussed together with all the results of the experiments in the Section 5.4.3.

5.2.2 BDD Encoding and Weighted Model Counting

Binary decision diagrams [40] are rooted directed acyclic graphs which represent Boolean functions $f : \{0,1\}^n \to \{0,1\}$, although by storing additional information outside the BDD they can also be used to represent pseudo-Boolean functions $f : \{0,1\}^n \to \mathbb{R}$. Two important properties of BDDs are their ability to compactly represent many functions by identifying redundancies, and their support for efficient operations (i.e. polynomial-time in the size of the BDD), such as computing marginal probabilities.

The joint probability distribution given by a BN is effectively a function of the form $f : \{0,1\}^n \to \mathbb{R}$ and can thus be encoded in a BDD. This is done by encoding each CPT entry in a small Boolean expression, from which a BDD can then be built using primitive BDD operations for logical and ($\wedge$), or ($\vee$), not ($\neg$), etc. As an example, consider the BN given in Figure 5.2. To capture the (integer) values of X and Y, Boolean variables $\{x_0, y_0, y_1\}$ are introduced, while unique probabilities are related to Boolean variables θ_i. As an example of the encoding of specific CPT entry, $P(Y = 2 \mid X = 0) = 0.4$ is encoded as $(\neg x_0 \wedge y_1 \wedge \neg y_0) \Rightarrow \theta_2$, where $\neg x_0$ corresponds to $X = 0$ and $y_1 \wedge \neg y_0$ corresponds to $Y = 2_{\text{dec}} = 10_{\text{bin}}$. The relationship $\mathsf{val}(\theta_2) = 0.4$ is stored outside of the BDD.

Computing marginal or conditional probabilities from a BDD that encodes a joint probability distribution can be done using so-called *weighted model counting* [46]. During weighted model counting the BDD is traversed, relevant probabilities are gathered along the way, and each node is visited at most once, resulting in a computation time linear in the size of the BDD. To compute interventional queries, the *do*-operator has been implemented through the adjustment formula (Equation 5.2) that utilizes the efficiently computed marginal and conditional distributions.

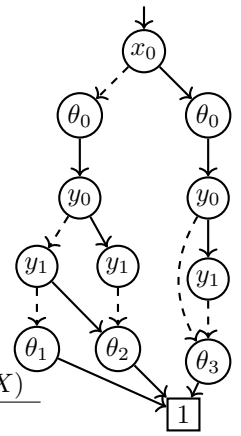


(a) BN structure

$P(X=0)$	$P(X=1)$
0.5	0.5

X	$P(Y=0\|X)$	$P(Y=1\|X)$	$P(Y=2\|X)$
0	0.2	0.4	0.4
1	0.33	0.33	0.33

(b) Conditional probability tables



(c) BDD

θ_i	val(θ_i)
θ_0	0.5
θ_1	0.2
θ_2	0.4
θ_3	0.33

(d) θ_i values

Figure 5.2: An example Bayesian network (a, b) and the corresponding BDD (c). The probabilities associated with the Boolean variables θ_i are shown in (d). In the BDD, solid (dashed) edges correspond to positive (negative) assignments. For clarity, edges to the 0-terminal are omitted in (c).

5.2.3 Optimization

Given the above discretization and encoding mechanisms, interventional queries can be computed and the causal global optimization problem (Equation 5.1) can be addressed heuristically. To illustrate this point, a variety of optimization algorithms from the *black-box optimization* literature [19], implemented in the Nevergrad [190] platform, are selected. The set of algorithms includes: (a) random search as a baseline, (b) two evolutionary algorithms (differential evolution [235] (DE) and (1+1) evolutionary strategy [32] (OnePlusOne)), (c) local search methods (BFGS [39, 73, 84, 215] and Powell's method [186]), (d) the algorithm wizard 'NGOpt39' [160], which automatically selects an optimizer based on problem characteristics such as the number of variables.

5.3 Experimental Setup

In applying the methodology depicted in Figure 5.1 to a range of Bayesian networks, the quality and efficiency of the resulting inference as well as the performance of the optimization algorithms are evaluated. Section 5.3.1 introduces the measures used to evaluate the quality of the discretization or inference and the measure used to assess

the computational cost of inference with decision diagrams. The evaluation procedure for the optimization process is also detailed. The specifics of the causal and non-causal Bayesian networks, including which networks are subjected to which experiments, are outlined in Section 5.3.2.

5.3.1 Evaluation Measures

First, different measures to assess the quality of discretization and inference are discussed, followed by a measure to evaluate the computational costs. Finally, the evaluation of the optimization algorithm is discussed.

Measuring the quality of discretization and inference

While f-divergences measure differences between probability distributions on the same measurable space [210], they are unsuitable for comparing a discretized state space and its continuous counterpart. Instead, the Wasserstein distance is used, specifically the Euclidean first-moment Wasserstein distance or earth mover's distance, to assess discretization quality as it is a common metric to compare (multivariate) distributions [255, 205, 12]. The earth mover's distance quantifies the dissimilarity between two probability distributions by measuring the minimum 'cost' to transform one distribution into the other.

A high-quality discretization does not necessarily imply that a query of interest can be computed accurately. Fortunately, as the synthetic BNs used in the experiments possess specific distributions that permit exact inference methods, access to the conditional inference queries is available. To evaluate inference quality, the conditional expected value of the original Bayesian network ($\mathbb{E}[Y \mid X]$) is compared to its discretized counterpart ($\mathbb{E}_{disc}[Y \mid X]$) using the weighted root mean squared error, where the weights adjust for the probability of the conditioned-on variables. For the causal Bayesian networks, the percentage error of the average treatment effect is used. The reader is referred to Appendix B.3 for a detailed description of these evaluation measures.

Measuring the computational costs of inference

As outlined in Section 5.2, inference using binary decision diagrams reduces to weighted model counting, which takes time linear in the size of the BDD. Therefore, the number of BDD nodes is considered a proxy for the computational costs of inference. Although BDDs can potentially grow exponentially in the size of the Bayesian network, they

typically remain smaller, enabling more scalable inference compared to traditional methods like variable elimination or belief propagation.[5]

The reported inference time for BDDs includes both compilation and weighted model counting. The runtime of inference with traditional methods is compared to the inference time with BDDs. Since VE and BP are implemented in Python and weighted model counting in C++, comparing their runtimes directly is inappropriate. Instead, scalability is assessed by measuring the time speed-up (seconds) as the number of bins in the Bayesian network increases.

Measuring optimization performance

For each optimization algorithm and each Bayesian network, 10 independent runs are performed of 2000 evaluations of the objective function each, where an evaluation consists of calculating the expected value of the outcome variable given an intervention set (see Equation 5.2), which is to be minimized in all networks considered in this paper (see Equation 5.1).

To represent the problem within the optimization algorithms, each node eligible for intervention is assigned a value between 0 and the number of bins if an intervention is performed; for convenience, a negative value is used to indicate the absence of intervention on that node. Within the Nevergrad library, such a representation gets translated to a real-valued one to allow continuous optimization algorithms to tackle this problem as well [190]. To track the optimization performance, IOHexperimenter [63] is used, a benchmarking module from the IOHprofiler environment [246], which allows us to track the optimization process fully. The expected value of the interventional distribution across 2,000 evaluations, averaged over 10 runs is reported.

5.3.2 Bayesian Network Description

In this section, the specifications of the non-causal and causal Bayesian networks that are subject to experimentation are highlighted. As the optimization experiments aim to embed the BDD-inference framework within an optimization loop, the BNs used in these experiments are generally a bit more expansive than those used solely in inference experiments. First, some general statistics for the Bayesian networks are summarized in Table 5.1, followed by a detailed description of each network. Finally, Table 5.2 presents the experimental characteristics of each Bayesian network.

[5]The Python implementation of pgmpy [11] is used for VE and BP to compare runtimes between BDD-inference and traditional inference.

Table 5.1: The general characteristics of all Bayesian networks. These include both synthetic and real datasets, varying in sample sizes, network complexity, and structural properties. The maximum number of parents (in-degree) serves as a proxy for the computational cost of inference. Due to the size of the Mehra and Arth networks and the 64GB RAM constraint, pruned but computationally equivalent versions were used for evaluation [23].

Dataset	Kind	Variants	Samples	Network		Max
				nodes	edges	parents
LG	synthetic	36	100-5000	5	4	2
NM	synthetic	8	100-500	2	1	1
CQ	synthetic	1	2500	3	3	2
Lalonde	real	1	2676	10	17	9
MC	synthetic	1	4000	12	15	6
Arth*	real	1	5000	107	150	17
Toy	synthetic	1	1000	3	2	1
Climate	real	1	293	8	11	3
Mehra*	real	1	5000	24	71	9

Linear Gaussian (LG) Bayesian network. Samples are drawn from a linear Gaussian Bayesian network [175] with random variables X_1, X_2, X_3, X_4, X_5. In total, 36 inference experiments were conducted, varying in sample size (N) and distribution parameters. To ensure a balanced experimental design, Sobol sequences were employed [78]. Detailed experimental specifications are provided in Tables B.3 and B.4 of Appendix B.2. The computational costs in terms of the number of nodes in the BDD is drawn against the WRMSE and against the earth mover's distance in Figure 5.5a, 5.5c and Figure 5.5b, respectively.

Normal mixture (NM) Bayesian network. Samples from a normal mixture Bayesian network are generated using a two-node Gaussian mixture model, for the purpose of conducting inference experiments. In this network, X_1 follows a Bernoulli distribution and $P(X_2|X_1)$ is Gaussian, based on similar experiments by Neil et al. [168]. Details on sample sizes and distribution parameters are listed in Table B.5 of Appendix B.2.

Causal quadratic (CQ) Bayesian network. In the context of the inference experiments, data is sampled from a quadratic data-generating process. The confounder Z is distributed normally and has a quadratic effect on outcome variable Y while also affecting treatment variable T. For the full specifications of the distribution of this experiment [176], the reader is referred to Appendix B.2. The computational costs

of inference have been set out against the percentage error of the ATE in the Pareto front of Figure 5.5d.

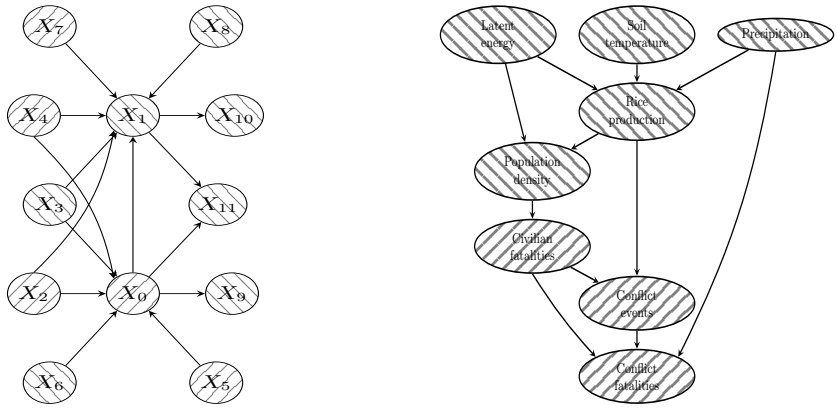
Lalonde causal Bayesian network. The Lalonde causal dataset is a real causal dataset in which the effect of temporary employment on income is studied [135], given confounding variables. Since both an observational [64] and an experimental dataset [135] are available, the non-parametric estimates of the average treatment effect can be compared with the difference in means in the observational and experimental datasets. The comparative analysis of computational costs of inference is presented alongside the percentage error of the ATE in the Pareto front depicted in Figure 5.5e.

Mixed confounding (MC) Bayesian network. To support both discretization and optimization experiments, samples are drawn from a synthetic dataset with mixed confounding, as detailed in the Csuite benchmarking causal datasets [79]. This dataset, depicted in Figure 5.3a, includes both continuous and discrete variables that causally influence multiple nodes in the graph in a non-linear manner. Figure 5.5f presents the computational costs of inference against the earth mover's distance within a Pareto front.

After discretizing the continuous variables into 30 bins, two optimization experiments are conducted on the outcome variables X_{10} and X_{11}, using the minimal intervention set $\{X_2, X_3, X_4, X_5, X_6, X_7, X_8\}$ [141]. Figures 5.6a and 5.6b show that DE and NGOpt19 outperform other optimization algorithms.

Arth Bayesian network. This large Gaussian Bayesian network, sourced from the GeneNet package and featured in bnlearn, contains plant expression data [173]. Due to the enormous size of the networks, a computationally equivalent pruned version of the network is compiled where every node is discretized into 6 bins in order to reduce compilation time [23]. A total of 6 intervention variables have been chosen for the optimization procedure. The results of the optimization experiments are shown in Figure 5.6d.

Toy. This dataset [4] contains a three-node $X \to Z \to Y$ Bayesian network discretized into 100 bins. While not interesting from an optimization perspective, it benchmarks inference quality post-discretization, focusing on possibly-optimal minimal intervention set [141] $\{Z\}$. Most optimization algorithms quickly converge to the



(a) Mixed Confounding BN **(b)** Climate Causal BN

Figure 5.3: Bayesian network structures corresponding to the mixed confounding and climate experiments. The hybrid nodes are distinguished by interior line patterns: dashed lines oriented from the top-left to bottom-right denote nodes with *continuous* values, whereas those from the top-right to bottom-left signify nodes with *discrete* values.

optimal solution in the discretization, $Y = -1.866$, which is near the known exact solution $Y = -1.856$.

Climate. The dataset comprises 294 samples, tapping into the interplay between climate and conflict as depicted in Figure 5.3b. It includes conflict, climate, environmental, and demographic data at the municipal level in southeastern Iraq. An understanding of the variables and the details of the empirically verified causal structure can be found in Malekovic et al. [153].

The outcome variable is the number of conflict fatalities. The interventions in consideration are precipitation, rice production, and population density, although direct intervention may be challenging. Indirect policy measures such as water management and development projects can be targeted to increase water availability and balance demographic distribution, respectively.

Figure 5.6e shows that DE and NGOpt19 are the best-performing algorithms. The tables in Figure 5.6f indicate best-found objective values and intervention values compared to the sample means in the dataset.

Mehra. This conditional linear Gaussian BN from bnlearn [213] explores the correlation between air pollution and health outcomes [247]. Due to its considerable size, the compilation of BDD is restricted to those segments of the network pertinent to

the optimization task [23]. The results can be seen in Figure 5.6c, where the identified best-performing algorithm is random search, closely followed by DE and NGOpt39.

Table 5.2: The experimental characteristics of the Bayesian networks. The table indicates whether each dataset is used for evaluating discretization quality (Disc), inference quality (Inf), or optimization performance (Opt). Inference quality evaluation is only possible when the ground truth of inference queries is available. For these inference experiments, the corresponding evaluation metric is reported. For optimization experiments, the number of discretization bins and the number of intervention variables considered are specified.

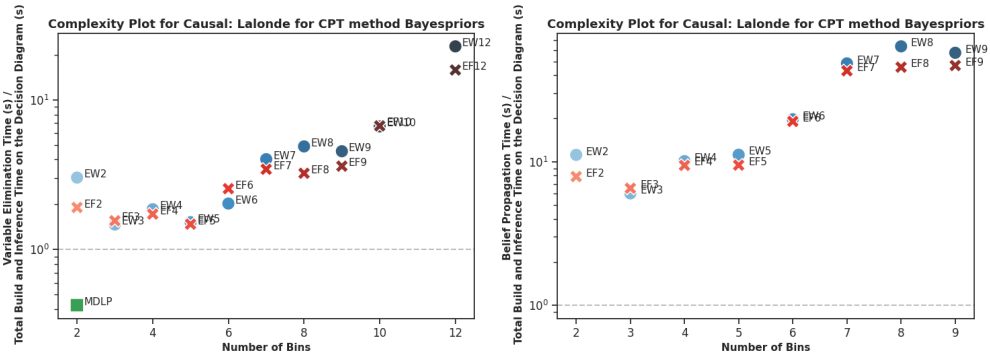
Dataset	Experiments	Inference comparison	Discretization bins	Intervention variables
LG	Disc/Inf	WRMSE	*NA*	*NA*
NM	Disc/Inf	WRMSE	*NA*	*NA*
CQ	Disc/Inf	PE ATE	*NA*	*NA*
Lalonde	Disc/Inf	PE ATE	*NA*	*NA*
MC	Disc/Opt	*NA*	30	7
Arth	Disc/Opt	*NA*	6	5
Toy	Opt	*NA*	100	1
Climate	Opt	*NA*	20	3
Mehra	Opt	*NA*	4	8

5.4 Results

First, scalability results are presented, comparing inference speed using binary decision diagrams against conventional approaches. Subsequently, Pareto fronts show the trade-off between computational cost and the quality of discretization and inference. Finally, the optimization algorithm's performance is analyzed across experiments by examining the average objective function value relative to the number of evaluations.

5.4.1 Scalability of Inference Method

In Figure 5.4, the speedup of Bayesian network inference via binary decision diagrams is compared to inference with variable elimination (VE) (Figure 5.4a) and belief propagation (BP) (Figure 5.4b) for the Lalonde Bayesian network. The Lalonde network has been chosen since it has a relatively high maximum in-degree, a proxy for the computational costs of inference. The fact that inference with binary decision diagrams becomes at least over 10 times faster than VE or BP as the number of bins increases underscores a notable improvement in scalability (in fact, for BP this is true for over 5 bins).



(a) BDD speedup versus VE **(b)** BDD speedup versus BP

Figure 5.4: The speedup plots for using binary decision diagrams as opposed to VE (a) or BP (b) for inference for the Lalonde experiment. The red crosses refer to EF binning, the blue circles represent the EW binning, and the MDLP binning is indicated by a green square. As the number of bins increases, using decision diagrams is more than 10 times as fast as both VE and BP.

5.4.2 Trade-off Computational Cost and Quality of Discretization and Inference

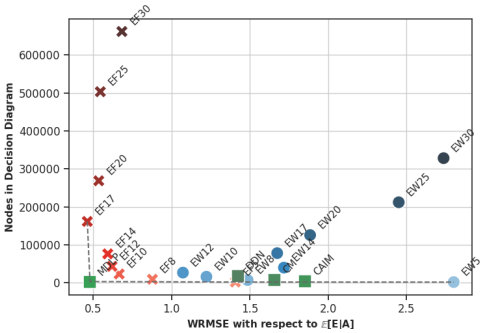
While all Pareto fronts are available at Zenodo,[6] a representative selection across all Bayesian networks and measures is highlighted in Figure 5.5. These Pareto fronts clearly demonstrate that increasing the number of bins results in a reduction of the earth mover's distance but an increase in computational costs. Simultaneously, the WRMSE and the percentage error of the ATE decrease as the number of bins rises, up to a certain number of bins.

To facilitate the interpretation of results across various experiments, the Pareto fronts have been condensed into the heatmaps presented in Tables B.1 and B.2 of Appendix B.1. All the experiments yield the following four key findings.

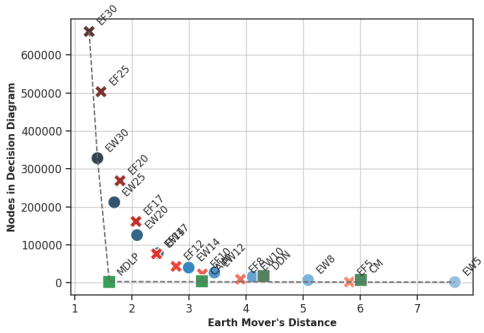
First, the solutions with the lowest earth mover's distance to the original BN are the most-binned solutions as can be observed in Figures 5.5b and 5.5f. In general, it can be observed that the earth mover's distance decreases when the number of bins used to discretize the BN increases, but the distance reduction becomes lower as the number of bins grows larger.

Second, the difference between inference results when estimating the CPTs with maximum likelihood estimate or the Bayesian method with adjusted empirical Bayes type 2 maximum likelihood priors is negligible. This similarity is evident from the

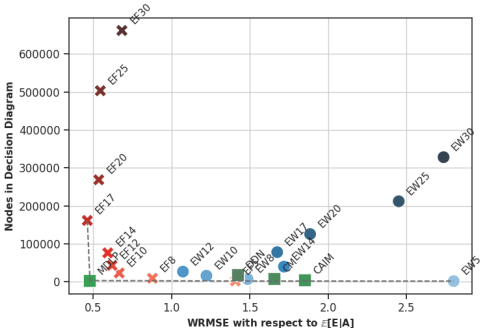
[6] `https://zenodo.org/records/11202314`



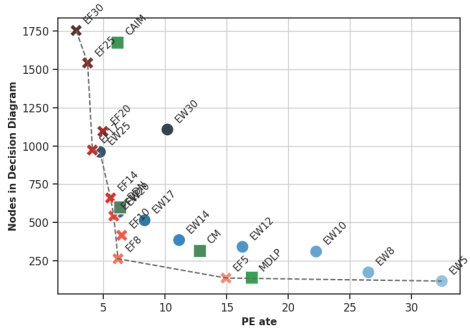
(a) WRMSE for the linear Gaussian experiment 9 with CPT method MLE.



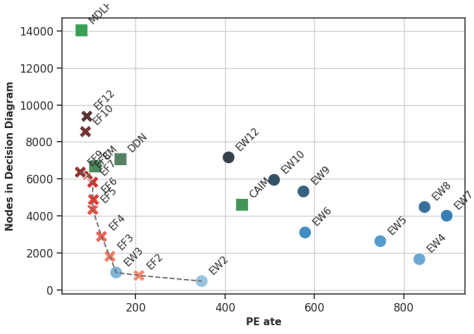
(b) Earth mover's distance for the linear Gaussian experiment 9.



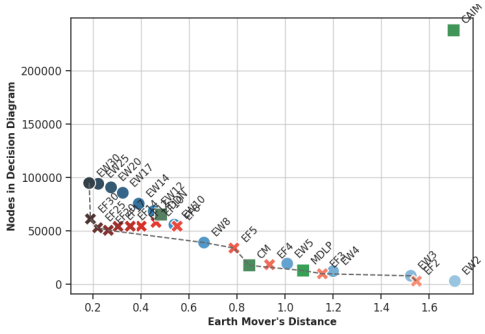
(c) WRMSE for the linear Gaussian experiment 9 with CPT method EBP.



(d) Percentage error of the ATE for the causal quadratic DGP.



(e) Percentage error of the ATE for the Lalonde dataset.



(f) Earth mover's distance for the mixed confounding dataset.

Figure 5.5: Number of nodes in the BDDs versus various evaluation measures for several discretization approaches with different parameter settings, per dataset. Approaches representing trade-offs between objectives (axes, both to be minimized) lie on the Pareto front (dashed line).

plots in Figure 5.5a and 5.5c, and supported by the data in Table B.2 in Appendix B.1. Additional discrepancy plots on Zenodo[7] further illustrate this negligible difference.

Third, the WRMSE and the PE decrease when adding bins up to a certain number of bins whereafter it increases again, indicating overfitting in data-sparse areas of the root variable. The Pareto fronts of Figure 5.5c, 5.5d, and 5.5e show that the bending point differs per experiment. Generally, more available samples or simpler BN structures lead to the solution with the lowest error being often a more intensely-binned solution.

Finally, it can incidentally be observed that one of the supervised discretization methods dominates the other solutions (as in the case with CM and MDLP in Figure 5.5f). However, no supervised discretization method performs exceptionally well across all experiments on the considered measures.

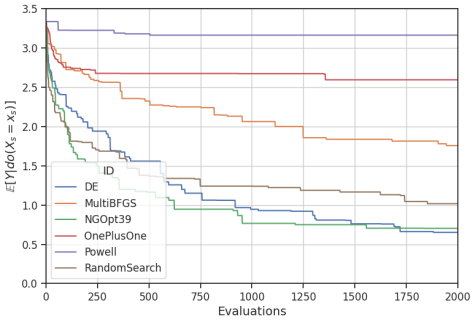
5.4.3 Optimization Performance

This section evaluates the performance of the optimization algorithms. Results from the previous section indicate that high-binned equal frequency discretizations yield among the most accurate query results. Therefore, these discretizations are used, reducing the bin count only when required to stay within the 64GB RAM constraint during BDD compilation. Details about the number of bins in the discretization and the number of interventional variables are displayed in Table 5.2, while the results are presented in Figure 5.6.

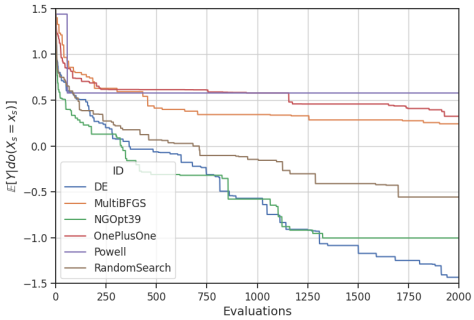
The more local methods (BFGS, Powell, OnePlusOne) perform rather poorly. Only DE and NGOpt are able to outperform the random search baseline, suggesting that the underlying optimization problem might be multimodal. This might also be connected to the choice of internal problem representation selected from Nevergrad, as working directly on the discrete variables might be more suitable for these local search methods. While this points to a need for further examination of the specifics of the optimization procedure, the results nevertheless illustrate that in general these problems *contain sufficient structure* that heuristic black-box optimizers can improve over the performance of random search.

As can be observed from Table 5.6f, with interventions tailored to increasing rice supply/production, equitable population management leading to a more balanced demographic distribution, and increased precipitation or related water management interventions, the expected value of conflict fatalities can be reduced by 85.9% with

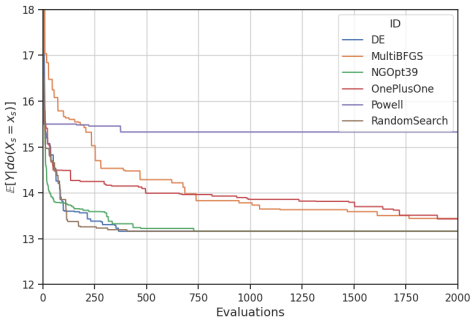
[7] `https://zenodo.org/record/8211601`



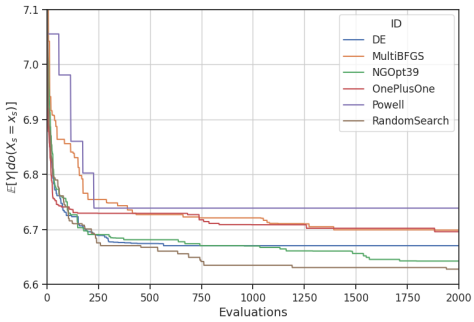
(a) Mixed Confounding: $Y = X_{10}$



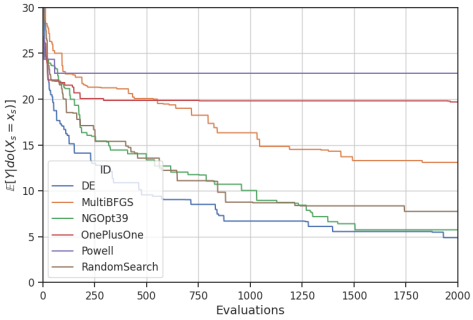
(b) Mixed Confounding: $Y = X_{11}$



(c) Mehra



(d) Arth



(e) Climate

	$\mathbf{y}^*$	$\bar{\mathbf{y}}$
Conflict Fatalities	4.89	34.6

	$\mathbf{x}^*$	$\bar{\mathbf{x}}$
Rice Production	36.9	13.0
Population Density	215	811
Precipitation	25.0	19.8

(f) Climate Dataset: Best-found solutions

Figure 5.6: The best-found expected value of the interventional distributions of 6 optimization algorithms over 2000 evaluations, averaged over 10 runs for the 4 considered datasets (a-e). For the climate dataset, the table in (f) corresponds to found objective $\mathbf{y}^*$ and intervention values $\mathbf{x}^*$ compared to sample mean values $\bar{\mathbf{y}}$, $\bar{\mathbf{x}}$.

respect to the mean.

5.5 Conclusion and Future Work

This chapter proposes a methodology to optimize causal interventions in hybrid causal Bayesian networks using only observational data. The methodology consists of discretizing the hybrid Bayesian network and encoding it into a binary decision diagram. Once the binary decision diagram is compiled, query costs become negligible, allowing the deployment of heuristic optimization algorithms that optimize for the optimal intervention. Given that discretization of a Bayesian network entails information loss, benchmarking this approach against established approximate inference methods, such as sampling or variational techniques, constitutes a promising direction for future research [127]

The estimates in Table 5.6f demonstrate the practical potential of the methodology, though they rely on a preliminary causal structure under the i.i.d. assumption introduced in Chapter 3. Chapter 6 further shows that causal estimates remain attainable when this assumption is relaxed. Extending the proposed methodology to account for causal spillovers or multiple strategic actors would enhance its relevance in complex, real-world scenarios.