Learning Discretized Bayesian Networks with GOMEA

Damy M.F. Ha
D.M.F.Ha@lumc.nl
Leiden University Medical Center
Leiden, The Netherlands

Tanja Alderliesten
T.Alderliesten@lumc.nl
Leiden University Medical Center
Leiden, The Netherlands

Peter A.N. Bosman
Peter.Bosman@cwi.nl
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands

ABSTRACT

Bayesian networks model relationships between random variables under uncertainty and can be used to predict the likelihood of events and outcomes while incorporating observed evidence. From an eXplainable AI (XAI) perspective, such models are interesting as they tend to be compact. Moreover, captured relations can be directly inspected by domain experts. In practice, data is often real-valued. Unless assumptions of normality can be made, discretization is often required. The optimal discretization, however, depends on the relations modelled between the variables. This complicates learning Bayesian networks from data. For this reason, most literature focuses on learning conditional dependencies between sets of variables, called structure learning. In this work, we extend an existing state-of-the-art structure learning approach based on the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) to jointly learn variable discretizations. The proposed Discretized Bayesian Network GOMEA (DBN-GOMEA) obtains similar or better results than the current state-of-the-art when tasked to retrieve randomly generated ground-truth networks. Moreover, leveraging a key strength of evolutionary algorithms, we can straightforwardly perform DBN learning multi-objectively. We show how this enables incorporating expert knowledge in a uniquely insightful fashion, finding multiple DBNs that trade-off complexity, accuracy, and the difference with a pre-determined expert network.

CCS CONCEPTS

• **Mathematics of computing** → **Bayesian networks**; • **Computing methodologies** → *Genetic algorithms.*

KEYWORDS

Bayesian networks, evolutionary algorithms, discretization, explainable AI

1 INTRODUCTION

Bayesian Networks (BNs) [13, 18] are probabilistic graphical models that model relationships between random variables under uncertainty. The relationships between variables can be depicted using a Directed Acyclic Graph (DAG). The process of optimizing the DAG for given (tabular) data, which is often called structure learning, has been extensively researched in the literature (e.g., [13, 17]) and applied to many real world applications such as in the medical domain: [7, 19, 27], geology and environmental modeling domain: [2, 14, 20], and (risk and safety) management: [10, 21, 28].

In the aforementioned domains, it is not uncommon to have a mix of discrete and continuous random variables. How to best incorporate continuous variables is however not straightforward. In the literature, there are various methods to extend discrete BNs with continuous variables. For example, a common method is to call upon a domain expert, who is tasked to pre-discretize continuous variables before structure learning or to model the continuous variables with a parametric distribution. It might however be difficult to consult a domain expert or they might not always be able to correctly model the variables. Non-parametric modelling of variables [5, 11] on the other hand, does not require expert knowledge. However, in non-parametric models, normality is usually assumed. Discretization techniques [6, 8, 9, 14, 25] offer an alternative as neither expert knowledge is required a priori, nor must the assumption of normality hold. The optimal discretization however, depends on the relations modelled between the variables, necessitating simultaneous optimization.

In this work, for the first time, a state-of-the-art structure learning approach based on the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) family of algorithms [17] is extended to jointly learn variable discretizations. The proposed Discretized Bayesian Network GOMEA (DBN-GOMEA) is compared to the state-of-the-art on randomly generated problems. When the algorithms are tasked to retrieve randomly generated ground-truth networks, it is shown that DBN-GOMEA obtains, similar or better performance than the state-of-the-art. Moreover, leveraging key strengths of EAs in multi-objective optimization, it is possible to straightforwardly perform DBN learning multi-objectively. The proposed approach is fundamentally different from e.g., [26], where a bi-objective search is performed on (proxies of) the accuracy and complexity and e.g., [1], where prior knowledge is included in the search by altering prior model probabilities according to expert knowledge. Our multi-objective approach leverages a tri-objective search to incorporate expert knowledge in a uniquely insightful fashion that enables finding multiple discretized BNs that trade-off (proxies of) the model accuracy, complexity, and difference to a pre-determined expert network.

The code is available at: https://github.com/damyha/dbn_gomea.

2 DISCRETE BAYESIAN NETWORKS

BNs [13, 18] are a class of probabilistic graphical models. A BN B is defined by a DAG G, which represents $\mathbf{X} = \{X_1, \cdots X_N\}$ random variables. Each node i in G is associated with a random variable X_i and has a (conditional) probability distribution $P(X_i|\text{pa}(X_i))$, where the probability of X_i is conditionally dependent on the parent nodes of X_i, i.e., $\text{pa}(X_i)$. In G, this relationship is modeled via a directed edge from each of the parent nodes to node i. An example of G is shown in Figure 1, where $\text{pa}(X_3) = \{X_1, X_2\}$, and X_3 is a parent of X_4. Node X_3, together with spouse X_6 are also parents of X_5. Given G and all conditional probabilities Θ, the probability of $\mathbf{X}$ can be written as a product of the individual conditional node probabilities, as is shown in Equation 1.

$$\mathrm{P}(X_1, \cdots, X_N) = \prod_{i=1}^{N} \mathrm{P}(X_i | \mathrm{pa}(X_i)) \tag{1}$$

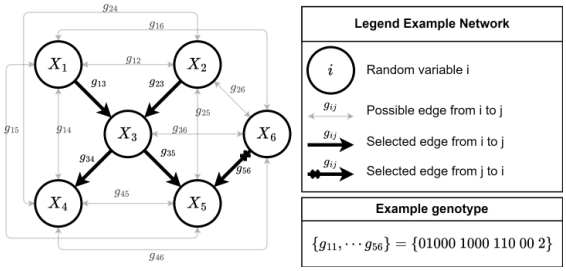

Figure 1: Example of a DAG used to represent a BN (in black) and all possible edges (in grey).

2.1 Bayesian Network GOMEA

In recent work, a state-of-the-art score-based BN structure learning algorithm was developed, called BN-GOMEA [17]. BN-GOMEA employs an Evolutionary Algorithm (EA) from the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) family. BN-GOMEA learns network structures from discrete data. It showed superior performance to other EAs, greedy hill-climbing, and tabu-list based algorithms such as Ordering-Based Search, Sparse Candidate, and Max-Min Hill-Climbing.

In BN-GOMEA, the BN structure learning problem is formulated as follows: solutions are represented as a string of discrete variables. Each variable in the string represents an edge between an arbitrary random variable i and random variable j (where $i \neq j$). The value of each variable can be 0, meaning no edge between i and j, 1, meaning a directed edge from i to j or 2, meaning a directed edge from j to i. The problem formulation results in $l_{\text{total}} = \frac{N}{2}(N-1)$ number of variables to represent all edges in the graph, where N is the total number of random variables. An example of a BN with a genotype representation is given in Figure 1. This problem formulation however, allows cyclic networks to exist. Therefore, a repair operator is used to remove cycles. To check if solutions contains a cycles, a depth-first search is executed. If a cycle is found, the last edge that completes the cycle in the depth-first search is removed.

BN-GOMEA, makes use of a linkage model that captures interdependencies between problem variables. A linkage model is made up of Family of Subset (FOS) elements, where each FOS element is a set of indices that indicate dependencies between the problem variables represented by those indices. The FOS elements are used during variation to effectively mix groups of variables to create fitter solutions. In [17], BN-GOMEA makes exclusively use of the linkage tree. The linkage tree is hierarchical tree structure which is learnt from the population. The Gene-pool Optimal Mixing (GOM) variation operator leverages this linkage tree. Each solution in the population undergoes GOM. First, the solution is cloned. Then the linkage tree is randomly traversed for this solution. For each FOS element in the linkage tree, a random donor solution is selected from the population. The variables, as indicated by the FOS element, are then copied to the offspring solution. If the change results in a worse fitness, the change is reverted in the offspring, otherwise the change is kept.

Other than the GOM operator, the excellent performance of BN-GOMEA can also be attributed to two other reasons. First, BN-GOMEA exploits the use of partial evaluations in combination with the linkage tree. When an offspring solution is created from a parent solution, each GOM step only changes part of the solution. It is more efficient to only recalculate the fitness contribution of variables that have changed, if the fitness function is decomposable. For typical fitness functions used with BNs, this is the case. In BN-GOMEA the decomposable BDeu score was used.

The second reason for BN-GOMEA's excellent performance is because a local search operator is additionally used. Upon initialization and after applying GOM to every solution in the population, the local search operator is applied on every solution in the population. The local search operator randomly traverses all variables of a solution and evaluates the fitness when the selected edge takes a different value, i.e., any value in $\{0, 1, 2\}$ different from the current value. During local search, only changes that result in a better fitness are accepted, otherwise the change is reverted.

At last, BN-GOMEA makes use of an Interleaved Multi-start Scheme (IMS), which runs multiple populations of various sizes side by side. The IMS avoids the user to excessively tune the population size manually. For this, the IMS ensures that a population of size n_{pop}, executes 4 generations before a population of $2 \cdot n_{\text{pop}}$ executes a single generation, starting from a base population of size 2.

3 DISCRETIZATION OF CONTINUOUS RANDOM VARIABLES IN BAYESIAN NETWORKS

3.1 DBN-GOMEA

In this work, BN-GOMEA is extended such that it can handle continuous random variables without prior discretization, i.e., the variables are discretized during structure learning. This new algorithm is dubbed Discretized Bayesian Network-GOMEA (DBN-GOMEA).

First, the BDeu score used in BN-GOMEA is replaced by a density based function, as the variables are reinterpreted in terms of density. The assumption that is made is that if data is discretized, it is uniformly distributed within that discretization. To optimize the uniform discretizations, the density of the discretizations should be maximized. As such, the log likelihood over the densities is taken as fitness function, as is shown in equation 2, where $\mathbf{x_i} \in \mathbb{R}^N$ is training sample i from a training data set and n the size of the training data set. To make sure that the density is invariant to the range of data, the data ranges are normalized to $[0, 1]$ prior to calculating the densities. This is similar to what has been done in [25]. As a penalty term, the penalty of the BIC score [24] is used, where the model complexity $\mathrm{C}(G)$ is dependent on the number of parent discretizations: $|\mathrm{pa}(X_i)|$, the number of discretizations of X_i: $|X_i|$, and n as shown in Equation 3. This results in the fitness function, displayed in Equation 4.

$$\mathrm{LL}(\mathbf{X}, G) = \prod_{i=1}^{n} \log(f_{\text{density}}(\mathbf{x_i})) \tag{2}$$

$$\mathrm{C}(G) = \sum_{i=1}^{N} |\mathrm{pa}(X_i)| \cdot (|X_i| - 1) \cdot \log(\frac{n}{2}) \tag{3}$$

$$\mathrm{fitness}(\mathbf{X}, G) = \mathrm{LL}(\mathbf{X}, G) - \mathrm{C}(G) \tag{4}$$

To discretize continuous variables, two common discretization methods are introduced, namely: Equal-Width (EW) and Equal-Frequency (EF). In EW discretization, data is split into 'k' equally ranged discretization bins. In EF discretization, data is sorted and split into 'k' equally filled discretization bins. In DBN-GOMEA, the number of discretizations 'k' is optimized by appending the discretization counts 'k' of each continuous random variable to the solution representation of BN-GOMEA, i.e., the representation is enlarged with N_c variables where N_c is the number of continuous variables.

As the solution representation is altered, the local search operator of BN-GOMEA is extended. In DBN-GOMEA, the original local search operator for the network topology is kept. However, when a solution variable is selected that represents a discretization count, the modified local search operator increases and decrease the discretization count by one, i.e., $\{k-1, k+1\}$. If the resulting number of discretizations falls outside the minimum or maximum number of discretizations, which are 2 and 15 respectively, the local search step is not executed. In this work, the minimum and maximum discretizations have been chosen to keep computation times feasible.

3.2 Post-structure Learning Discretization

Although EW and EF discretization are commonly used in the literature, in practice it is unlikely for data to be EW or EF distributed. As a consequence, an inaccurate discretization might be found. For this reason, the effect of optimizing the discretization boundaries will be investigated. This will however be done after structure learning has finished, as structure learning and discretization can become expensive when the sample size grows.

The algorithm selected for this task is the Real-Valued GOMEA (RV-GOMEA) [3, 4], which is a state-of-the-art real-valued optimization algorithm. With the network structure and number of discretizations for each continuous random variable fixed, the boundaries can be optimized using the same density fitness function. As a result, the log likelihood over the densities is potentially further optimized, without a change in complexity.

The boundary optimization in RV-GOMEA is encoded by concatenating all boundaries to be optimized into a single solution. Instead of directly optimizing the boundaries of the data, the optimization problem is reformulated by sorting the unique data values $\mathbf{u}$ of each continuous random variable and to optimize the sample indices that separate the data. By optimizing the sample indices, the flat-landscape between samples becomes equiprobable, compared to directly optimizing the boundaries. The boundary at sample index i is then calculated by taking the midway point between sample u_i and sample u_{i+1}. As RV-GOMEA optimizes real-valued problems, the solution parameters (which represent the sample indices) are rounded down.

For RV-GOMEA, the linkage tree is once again used as a linkage model.

3.3 Bayesian Method

In [6], a discretization method is proposed that finds a discretization Λ that maximizes a likelihood score, given a BN structure. The method uses Bayes rule to maximize: $P(\Lambda) \cdot P(D|\Lambda)$, where $P(\Lambda)$ is the prior of a discretization policy and $P(D|\Lambda)$ the probability of the data given the discretization policy. The likelihood is formulated by making assumptions, of which one assumption is that the prior probability of a discretization boundary between two unique consecutive sample values is proportional to their difference.

Maximization of the likelihood is implemented via dynamic programming. By doing pre-calculations, the discretization runtime is reduced to $O(r \cdot n^2)$, where r is a constant and n the sample size.

In [6] furthermore, this Bayesian discretization method is combined with a structure learning algorithm. The combination of both algorithms is dubbed LDBN in this work. In LDBN, the structure is learnt by first applying EW discretization on all continuous random variables, where the number of discretizations is selected to be the largest number of instantiations amongst all discrete random variables. After EW discretization, an arbitary random variable is selected as a starting point. The remaining random variables are randomly selected and sequentially added to the BN structure as child nodes. An edge between the new child node and potential parent node(s) materializes when the K2 score of the network improves after adding the edge. If a new edge is added to the network, the Bayesian discretization method is applied on all nodes that fall within the Markov blanket of the new node. All nodes within the Markov blanket are sequentially discretized in a random order.

As both the structure learning algorithm as well as the Bayesian discretization contain randomness, LDBN runs the structure learning and discretization algorithm multiple times. We kindly refer to [6] for more details.

4 MULTI-OBJECTIVE LEARNING

A major limitation of Single-Objective (SO) BN learning is that the weight of the complexity term is not straightforward to set [26]. Furthermore, the obtained model might not be trusted by an expert when the expert has their own beliefs. Taking a Multi-Objective (MO) perspective can offer a solution to these problems. First, using MO search does not require the user to know the penalty factor a priori. Furthermore, the search returns many networks out of which a domain expert can choose a network that matches (partly) with their own prior belief or discover new knowledge this way.

A straightforward way to do MO search is to optimize (a proxy of) accuracy and model complexity as in e.g., [26], where an EA is used, performing MO search relatively straightforward. For GOMEA, multi-objective variants also exist that are direct extensions of the SO versions, necessitating no further adaptions to e.g., DBN-GOMEA's genotype or operator. Using the density function as is, an MO version of the problem can be created straightforwardly by making Equation 2 and Equation 3 separate objectives. See Section 4.1 for more on this. A subset of the networks obtained from this MO search can then be shown to an expert, who decides which network is most appropriate, observing the fit to the data and matching with their own beliefs. The inclusion of expert knowledge however, could provide additional guidance to the search. In this work, not only are the density and complexity optimized as separate objectives,

the difference with an a priori determined expert network is also optimized. To this end, the Kullback-Leibler (KL) divergence is used as a distance between a candidate network and an expert network. The KL divergence is shown in Equation 5, where $P(\mathbf{X})$ is the probability distribution of the expert network, $Q(\mathbf{X})$ the probability distribution of a candidate network, and $\mathcal{X}$ the sample space. The KL divergence is 0 when two probability distributions are identical, and is larger than 0 otherwise. An example of the objective space of a resulting MO run is shown in Figure 2.

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(\mathbf{X}) \log \left(\frac{P(\mathbf{X})}{Q(\mathbf{X})} \right) \tag{5}$$

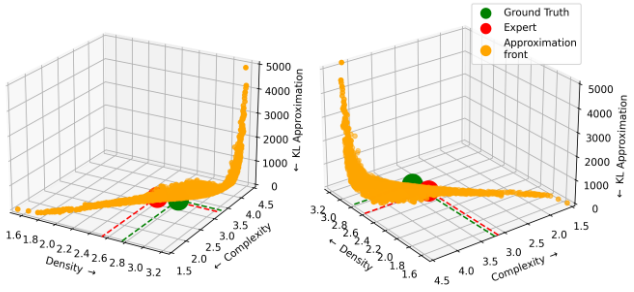

Figure 2: Impression (rotated visualizations) of the approximation front of a multi-objective run together with the objective values of the ground truth and expert solutions. The x, y, and z axis are on a log-log-linear scale.

4.1 MO-DBN-GOMEA

For the MO optimization, Multi-Objective Gene-pool Optimal mixing Evolutionary Algorithm (MO-GOMEA) [16] is used. MO-GOMEA is a state-of-the-art multi-objective EA that is also part of the GOMEA family. It can similarly exploit partial evaluations for enhanced efficiency. MO-GOMEA uses domination-based optimization, i.e., it uses the concept of Pareto dominance to find better solutions. A solution is said to Pareto dominate another solution if it is not worse in any objective and better in at least one objective.

Given m objectives that need to be optimized, a population in MO-GOMEA is partitioned into c clusters of equal sizes. For each cluster, a linkage model is learnt. In this work, the linkage tree is used, which is similar to the linkage tree in Section 2.1. Each cluster is evolved using the respective linkage model. A select number of clusters (specifically m) with the (respective) highest average objective values are selected to optimize the individual objective functions in a SO setting using the SO GOM operator. For the remaining clusters the MO GOM operator is used. Different from the SO GOM operator, the MO GOM operator accepts a solution when any of the following holds: 1) the GOM altered solution dominates the unaltered solution, 2) the altered solution has the same objective values, 3) the altered solution is not dominated by any solution in the elitist archive. The elitist archive is an archive of non-dominated solutions found during the optimization. In this work, an elitist archive size of 10,000 is used to collect as many solutions as possible during the search, while balancing computation time.

Similar to BN-GOMEA, MO-GOMEA uses the IMS to manage its population size. However, the IMS in MO-GOMEA additionally manages the number of clusters c in a population. The population size starts at 8 and is multiplied by 2 for every new population size. The number of clusters starts at $m + 1$, and is incremented by 1 for every new population. For more details, we kindly refer to [16].

In this work, MO-GOMEA is slightly extended by making MO-GOMEA capable of solving discrete problems over binary problems only. For this, the suggestions proposed in [16] are followed. The most important change made, is replacing the binary linkage tree in MO-GOMEA with the discrete linkage tree of [17]. Furthermore, the Bayesian network structure learning, as proposed in Section 2 is integrated into MO-GOMEA. The new structure learning algorithm is dubbed MO-DBN-GOMEA.

5 EXPERIMENTS AND RESULTS

5.1 Network Generation

In this work, randomly generated BN structures and probability distributions are used to assess the performance of the algorithms. For this, the network generator algorithm of [12] is used to generate random BNs. Probability distributions are generated using a method described below. Data sets are sampled from the ground truth networks and given to the algorithms. In [12], random BN structures are generated under constraints. The maximum number of parent random variables are chosen to be 6 and the maximum number of edges in a network are set to be at most 40% of all possible edges l. These constraints have been chosen, such that networks can be evaluated within reasonable time.

The probability distributions are randomly generated by first separating the random variables into discrete and continuous variables. The number of discrete variables is set to 10% with a minimum of at least one discrete variable per ground truth network. Each random variable, whether discrete or continuous, is then randomly assigned between 2 and 6 discretizations, e.g., if a random variable is randomly assigned 5 discretizations, the possible values are: $\{1, 2, 3, 4, 5\}$. A discrete probability table is then generated for each random variable, that maps the possible parent values to a probability of a specific discretization value. In this work, the probability tables are generated in three ways: EW, EF or random probability distributions. For EF probability distributions, the probability of sampling any value is equiprobable. For EW and random probability distributions, random probability tables are generated.

Discrete samples can now be retrieved. For the continuous variables however, the discrete probability tables must be converted to continuous probability distributions. For this, a mapping is generated that maps each discrete value to a specific range of continuous values. Continuous samples can be obtained by uniformly sampling from this range. For example, if a continuous random variable has 3 discretizations with ranges: $[1.0, 2.0\rangle$, $[2.0, 2.5\rangle$, $[2.5, 3.5\rangle$, and a discrete value of 2 is sampled, a continuous sample is produced by uniformly sampling from $[2.0, 2.5\rangle$. The sample ranges are designed to be adjacent to each other and non-overlapping. For EW probability distributions, the sample ranges are set to be equally spaced. For EF and random probability distributions, the sample ranges are determined randomly.

5.2 Metrics

The performance of the algorithms is assessed using various metrics. In terms of network structure metrics, two metrics are used. First, the accuracy is defined as the sum of the number of correctly identified edges TP and correctly identified absent edges TN of a candidate network structure as a percentage of the sum of the total number of edges l_{total} of the ground truth network (see Equation 6). A correctly identified edge is defined as an edge in the ground truth network which also appears in the candidate network without regarding the directionality of the edge. Note that this definition is different from other works such as e.g., [15]. A correct absent edge is defined as an absent edge in both the ground truth network and candidate network. The sensitivity is defined as the number of correctly identified edges TP as a percentage of the total number of edges in the ground truth network: l_{edges} (see Equation 7).

$$\text{Accuracy} = \frac{TP + TN}{l_{\text{total}}} \tag{6}$$

$$\text{Sensitivity} = \frac{TP}{l_{\text{edges}}} \tag{7}$$

To assess the quality of the discretizations, the KL divergence with respect to the ground truth network is used. The KL divergence was already introduced in Section 4.

5.3 Single-Objective Scalability

5.3.1 Single-Objective Scalability in Terms of Sample Size. The scalability in terms of sample size is shown for various algorithms in Figure 3. For this, 30 ground truth networks, each having 8 random variables, are generated with EW, EF and random probability distributions. To asses the KL divergence, 50.000 test samples are generated. DBN-GOMEA with EW, EF, and Bayesian Discretization (BD), as well as the structure learning algorithm from [6] (LDBN) are considered in these experiments. DBN-GOMEA-EF and DBN-GOMEA-EW are run on an Intel E5-2690 where each run uses a single core with 2GB of memory and 24 hours of computation time. The Bayesian discretization's memory requirements scale with $O(n^2)$. Therefore, LDBN and DBN-GOMEA-BD are run on a (newer) E5-4650 with 20GB of memory per run and 24 hours of computation time. Due to computational constraints, it was not possible to run all algorithms on the E5-4650.

Figure 3 shows that for EW, EF, and random probability distributions, in general, DBN-GOMEA with the appropriate discretization techniques finds better network structures as well as better KL divergence when the sample size grows. Only DBN-GOMEA-EW obtains perfect network retrieval for EW problems, given at least 6400 samples. DBN-GOMEA-EF does not achieve perfect retrieval for EF problems, for any tested sample size. Note however, that the EF data is sampled from the ground truth network and thus not perfectly EF distributed. Hence; when using EF discretization, the boundaries are not optimal. Interestingly, LDBN in general, shows worse performance than DBN-GOMEA in terms of network metrics, KL divergence or in some cases in both. LDBN also runs out of memory when there are too many samples. DBN-GOMEA-BD also runs out of memory. For small sample sizes however, except for the EW problems, it can find better network structure and similar or better KL divergence compared to DBN-GOMEA-EW and DBN-GOMEA-EF.

Number of samples	DBN-GOMEA-EW	DBN-GOMEA-EF	DBN-GOMEA-BD	LDBN
200	1.57 ± 0.58	**0.94** ± 0.21	1.62 ± 0.89	3.43 ± 1.27
400	1.45 ± 0.59	**0.83** ± 0.19	**1.16** ± 0.78	3.39 ± 1.17
800	1.38 ± 0.61	**0.73** ± 0.18	**0.94** ± 0.65	3.37 ± 1.24
1600	1.32 ± 0.64	**0.64** ± 0.17	**1.02** ± 0.73	2.66 ± 1.48
3200	1.26 ± 0.64	**0.57** ± 0.18	1.82 ± 1.43	-
6400	1.21 ± 0.66	**0.50** ± 0.17	-	-
12800	1.16 ± 0.66	**0.46** ± 0.17	-	-
25600	1.09 ± 0.63	**0.43** ± 0.18	-	-
51200	1.07 ± 0.62	**0.41** ± 0.18	-	-

Table 1: The average KL divergence values and standard deviation to the ground truth networks of various algorithms and for different sample sizes. In bold are the best KL values and those statistically not different from it. The ground truth networks have random probability distributions.

To test for differences in the KL divergence, a Mann–Whitney U statistical test is performed. Results obtained from ground truth networks with random probability distributions are investigated. In table 1, the mean ± the standard deviation of the KL divergence is shown. Numbers in bold have the best average KL divergence or are statistically not different from the best. An alpha value of 0.05 is used, with a Bonferroni correction of 63 as 27 tests are performed and 36 more tests will be performed later on. Table 1 shows that DBN-GOMEA-EF is the overall best, with DBN-GOMEA-BD performing similar on problems with few samples.

5.3.2 Single-Objective Scalability in Terms of Random Variables. The scalability in terms of number of random variables in a network is shown in Figure 4. For this, 30 ground truth networks, with random probability distributions, are generated per ground truth network size. Each run was performed using 500 training samples, on a single core of an AMD Genoa 9654, with 2GB of memory and a computation budget of 24 hours.

Figure 4 shows that DBN-GOMEA-BD obtains more accurate and more sensitive network structures when there are few nodes. However, after more than 12 nodes, the networks become less accurate compared to the ones obtained by the other algorithms. At the same time, the time it takes to find the best solution also nears the computation budget of 24 hours. This begs the question if DBN-GOMEA-BD needs more time to converge. The network accuracy obtained by the other algorithms seems to be similar, especially after 12 nodes.

Despite having similar network structures, the KL divergence does seem to differ per algorithm. To test for statistical differences in the KL divergence, a Mann–Whitney U statistical test is performed. In Table 2, the mean ± the standard deviation of the KL divergence is provided. Numbers in bold have the best average KL divergence or are statistically not different from the best. An alpha value of 0.05 is used, with a Bonferroni correction of 63 as 27 tests were performed previously and 36 more tests are performed in Table 2.

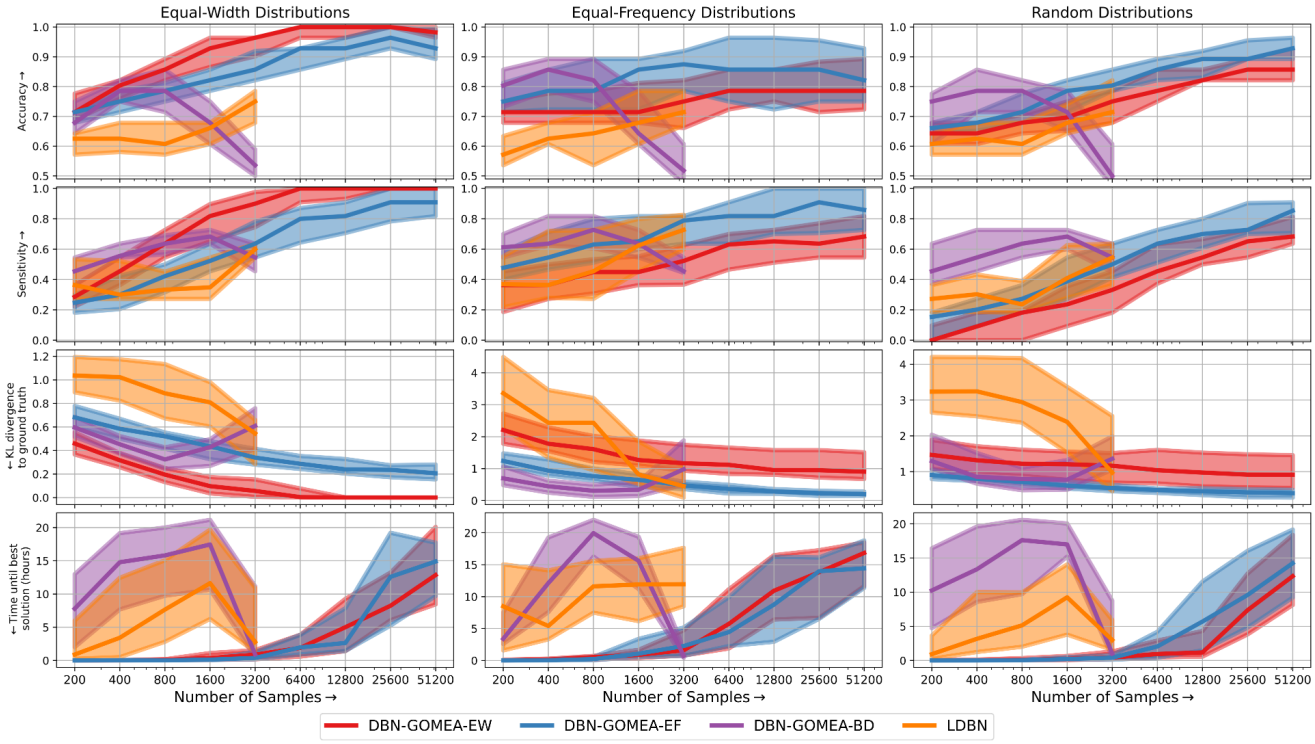

Figure 3: Scalability in terms of sample size for 30 random networks with 8 random variables having EW, EF and Random probability distributions. The solid lines are medians, while the shaded areas encompass the first and third interquartile ranges. The arrows on the y-axis point in the direction of improvement per metric.

Random Variables	DBN-GOMEA-EW	DBN-GOMEA-EF	DBN-GOMEA-BD	LBDN
4	**0.58** ± 0.45	**0.29** ± 0.14	**0.45** ± 0.40	1.11 ± 0.81
6	0.98 ± 0.56	**0.51** ± 0.16	**0.57** ± 0.40	2.31 ± 0.86
8	1.26 ± 0.51	**0.78** ± 0.18	**1.09** ± 0.63	3.04 ± 1.05
10	2.11 ± 0.69	**1.15** ± 0.22	**1.89** ± 1.16	4.42 ± 1.35
12	2.57 ± 0.91	**1.54** ± 0.25	3.02 ± 1.06	4.63 ± 1.07
14	3.29 ± 0.93	**1.98** ± 0.28	4.49 ± 1.32	6.42 ± 1.46
16	3.47 ± 0.82	**2.32** ± 0.29	5.69 ± 1.25	6.97 ± 1.73
18	4.32 ± 1.06	**2.76** ± 0.30	6.95 ± 1.43	7.85 ± 1.74
20	4.66 ± 0.82	**3.14** ± 0.33	8.33 ± 1.77	8.35 ± 1.52

Table 2: The average KL divergence ± the standard deviation of various algorithms optimized on 30 ground truth networks with 500 samples and random probability distributions for various number of random variables. The best KL scores and the statistically insignificant results are marked in bold.

Table 2 shows that, similar to Table 1, DBN-GOMEA-EF is amongst the best in terms of KL divergence. Table 2 also shows that DBN-GOMEA-BD is amongst the best in terms of KL divergence when the number of random variables is small. LDBN and DBN-GOMEA-EW perform relatively poorly.

5.4 Post-Structure Learning Discretization

Figure 3 and Figure 4 have shown that DBN-GOMEA-EW and DBN-GOMEA-EF can retrieve accurate networks within relatively short time compared to the other algorithms. DBN-GOMEA-EW and DBN-GOMEA-EF, however, do not optimize the discretization as granularly as e.g., BD. To investigate the effect of doing post-structure learning discretization, all network structures of Figure 3, obtained using DBN-GOMEA-EW and DBN-GOMEA-EF on ground truth networks with random probabilities, are once more discretized. The discretizations are optimized with RV-GOMEA and the BD. RV-GOMEA is tasked to optimize the density fitness function (Equation 4) within a budget of 24 hours and is ran on a E5-2690 with 2GB of memory. The BD algorithm is ran on a E5-4650 with 20GB of memory.

The effect of optimizing the discretization after completing structure learning is shown in Figure 5. Figure 5 also shows the original discretization obtained with DBN-GOMEA-EW and DBN-GOMEA-EF as a reference. Note that the time until the best found solution in Figure 5 does not include the original 24 hours of structure learning. Figure 5 shows that when RV-GOMEA is applied (purple and orange), the median KL divergence improves compared to not doing post-structure learning discretization (red and blue) regardless whether the structure was obtained using EW or EF discretization. The BD method however, seems to perform poorly when there are not enough samples in combination with having inaccurate structures (as seen from Figure 3). After 12800 samples, BD runs out of memory.

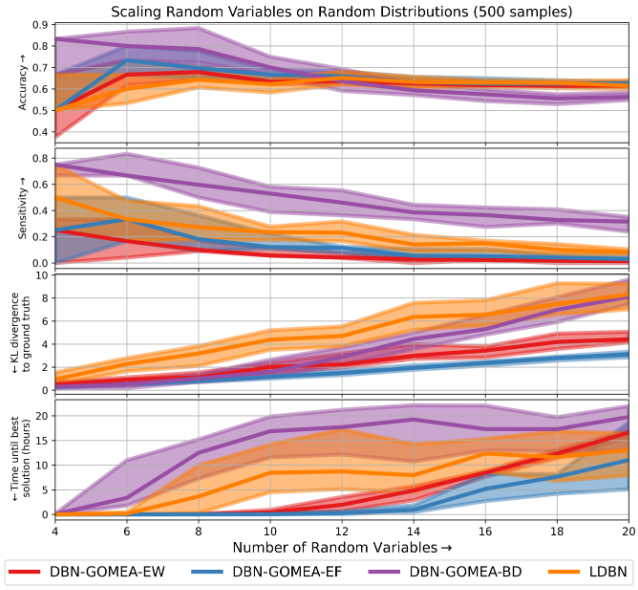

Figure 4: The scalability in terms of number of random variables. For each number of random variables on the x-axis, 30 ground truth networks were generated with random probability distributions. The solid lines are medians, while the shaded areas encompass the first and third interquartile ranges. The arrows on the y-axis point in the direction of improvement per metric.

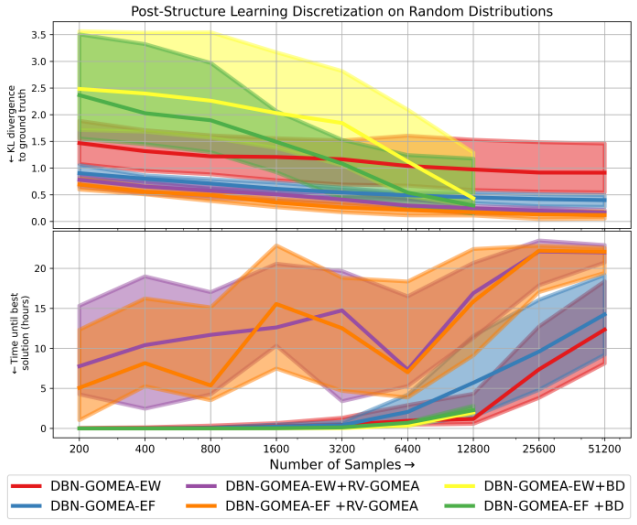

Figure 5: Optimizing the discretization after structure learning. The networks of Figure 3, obtained using DBN-GOMEA-EW and DBN-GOMEA-EF, are further optimized. The lines indicate the median, while the shaded regions encompass the first and third quartiles. The arrows on the y-axis point in the direction of improvement per metric.

5.5 Multi-Objective Experiment

To test the robustness of the MO search, ground truth networks are generated along with expert networks. The expert networks are a simulated representation of what a domain expert believes is the ground truth. The expert networks are randomly generated based on the ground truth networks. In this experiment, the expert networks are configured to know 50% of the edges of the ground truth network, i.e., 50% of l_{edges}. This is similar to what has been done in [1]. Additional to this, the experts networks are also configured to believe in edges that do not appear in the ground truth network. The number of incorrect edges is also set to 50% of l_{edges}. The incorrect edges are randomly selected. In this process, networks with cycles are rejected until an acyclic network is found. For the continuous variables, the expert networks also need to determine how the random variables are discretized. For each continuous random variable, the expert network randomly makes between 2 and 4 discretizations. How the data is discretized, i.e., where boundaries are put, is also random.

For the MO search, MO-DBN-GOMEA with EW and EF discretization are used. In an explainable AI setting, proposed networks that are too complex might be less likely to be accepted by the expert. For this reason, proposed solutions with a complexity (Equation 3) larger than 10 times the expert network are assigned a constraint value proportionate to the difference in complexity. This threshold however, is problem and expert dependent. In this experiment, it only serves as an example. The number of maximum discretizations is also decreased from 15 to 9, as experts are unlikely to accept complex discretizations. SO algorithms DBN-GOMEA-EW and DBN-GOMEA-EF are also ran for comparison with the same number of maximum discretizations.

The results of the MO search on 30 randomly generated ground truth networks with 10 random variables and random probability distributions is shown in Figure 6 for various sample sizes. Each run was performed on a single core of an AMD Genoa 9654, with 2GB of memory and a computation budget of 24 hours. In the top two rows of Figure 6, the highest obtained network accuracy is shown with respect to the ground truth and expert networks. For the MO algorithms, the most accurate solutions in the elitist archive are displayed per run. For the SO algorithms, the best found solution's network accuracy is shown. Interestingly, both MO algorithms are able to obtain networks with better accuracy compared to the SO algorithms for the tested sample sizes. In the case of the ground truth network accuracy, the gap between the MO and SO algorithms, does shrink when the sample size grows. In terms of accuracy to the expert network, the MO algorithms perform better than the SO algorithms, as the SO algorithms do not optimize towards the expert network (which is explicitly done in the MO setting).

The best found KL divergence is also shown in Figure 6 on a log scale. For the MO algorithms, the best KL divergence is shown amongst all solutions with the highest network accuracy, not the entire elitist archive. For the SO algorithms, the best found solution's KL divergence is shown. Interestingly, both the MO and SO algorithms using EF discretization obtain similar KL divergence to the ground truth network. This is not the case for the KL divergence to the expert network as once again, the SO algorithms do not optimize towards the expert network.

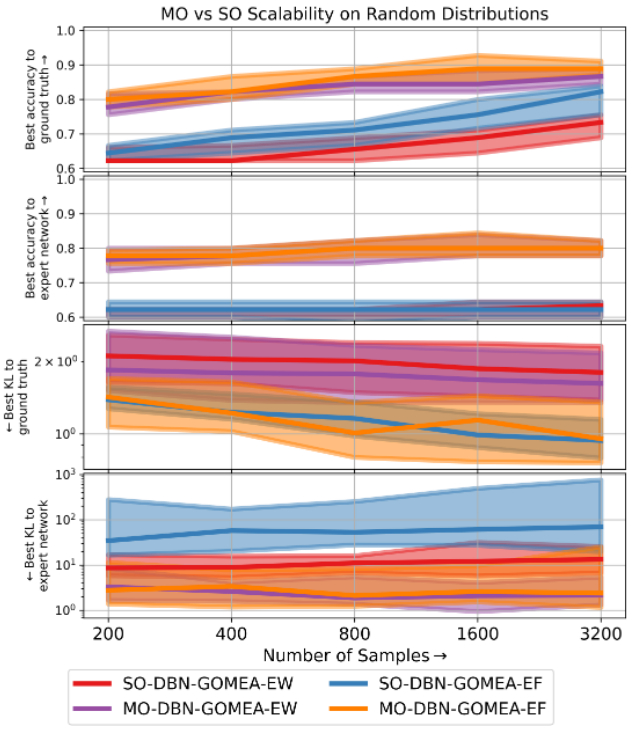

Figure 6: MO vs SO scalability in terms of sample size on ground truth networks with 10 nodes, random probability distributions, and random expert networks. The solid lines are medians, while the shaded areas encompass the first and third quartiles. The arrows on the y-axis point in the direction of improvement per metric.

6 DISCUSSION

A state-of-the-art of the art structure learning algorithm based on the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) was extended with discretization-based methods to handle continuous data. In this work, DBN-GOMEA made use of the linkage tree. In the encoding of a solution, the network variables can take three values, namely: $\{0, 1, 2\}$. When e.g., Equal Width (EW) or Equal Frequency (EF) discretization is applied, the number of discretizations is also encoded in the solution. The number of discretizations can take a value between 2 and a maximum value for every continuous variable. As the network variables and binning variables have an unequal number of values they can assume, the linkage tree tends to cluster discretization variables together, in the lower parts of the linkage tree. Mixing the network variables and discretization variables could make the optimization even faster, as graphically, discretization variables and edges are structurally related. For this, normalizing the mutual information could help.

In Section 5.3.1, the effect of the sample size on the network accuracy was shown for randomly generated networks with 10 random variables. It was shown that even for large sample sizes, DBN-GOMEA-EF was unable to fully re-obtain the EF ground truth network. Conversely, DBN-GOMEA-BS obtained better KL divergence than DBN-GOMEA-EF for smaller sample sizes. This suggests that more sophisticated discretization techniques, compared to EW and EF discretization, might be required to retrieve the full ground truth network.

An interesting approach would be to employ a mixed-integer algorithm, such as [22], which handles both integer and real-valued variables. The mixed-integer approach could encode the network structure using integers (as is done in DBN-GOMEA), while encoding the discretization with real values.

In this work, a multi-objective Bayesian network learning algorithm was also introduced. Currently, only EW and EF discretization have been ran, as Bayesian discretization is too expensive to run, especially when evaluating complex networks. A multi-objective mixed-integer approach, such as [23], could also be interesting for this problem.

In some real-world domains, blindly trusting machine learning models is not acceptable from a legal aspect. The multi-objective approach proposed in this work however could be useful when used as an advisory model, as it provides the possibility to inspect multiple possible models and trade-off between complexity and accuracy. Exploring the potential added value of our approach from an explainable AI perspective, by having domain experts interact with the found works, is therefore interesting future work.

7 CONCLUSION

In this work, for the first time, a full Bayesian network learning algorithm based GOMEA is presented, which jointly discretizes continuous variables during structure learning. In the single-objective case, the proposed algorithm (DBN-GOMEA) obtains similar or better results than the state-of-the-art when tasked to retrieve randomly generated ground-truth networks. Moreover, leveraging a key strength of EAs, the Bayesian network learning is brought to the multi-objective domain. It was shown how this enables incorporating expert knowledge in a uniquely insightful fashion, finding multiple discrete Bayesian networks that trade-off complexity, accuracy, and the difference with a pre-determined expert network.

8 ACKNOWLEDGMENTS

This research is part of the research programme Open Competition Domain Science-KLEIN with project number OCENW.KLEIN.111, which is financed by the Dutch Research Council (NWO). Furthermore, we thank NWO for the Small Compute grant on the Dutch National Supercomputer Snellius.

REFERENCES

[1] Hossein Amirkhani, Mohammad Rahmati, Peter J. F. Lucas, and Arjen Hommersom. 2017. Exploiting Experts' Knowledge for Structure Learning of Bayesian Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 11 (2017), 2154–2170. https://doi.org/10.1109/TPAMI.2016.2636828

[2] Tomas Beuzen, Lucy Marshall, and Kristen D. Splinter. 2018. A comparison of methods for discretizing continuous variables in Bayesian Networks. *Environmental Modelling & Software* 108 (2018), 61–66. https://doi.org/10.1016/j.envsoft.2018.07.007

[3] Anton Bouter, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. 2017. Exploiting Linkage Information in Real-Valued Optimization with the Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Berlin, Germany) *(GECCO '17)*. Association for Computing Machinery, New York, NY, USA, 705–712. https://doi.org/10.1145/3071178.3071272

[4] Anton Bouter and Peter A. N. Bosman. 2023. A Joint Python/C++ Library for Efficient yet Accessible Black-Box and Gray-Box Optimization with GOMEA. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*

(Lisbon, Portugal) *(GECCO '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 1864–1872. https://doi.org/10.1145/3583133.3596361

[5] Anna V. Bubnova, Irina Deeva, and Anna V. Kalyuzhnaya. 2021. MIxBN: library for learning Bayesian networks from mixed data. *Procedia Computer Science* 193 (2021), 494–503. https://doi.org/10.1016/j.procs.2021.10.051 10th International Young Scientists Conference in Computational Science, YSC2021, 28 June – 2 July, 2021.

[6] Yi-Chun Chen, Tim A. Wheeler, and Mykel J. Kochenderfer. 2017. Learning Discrete Bayesian Networks from Continuous Data. *J. Artif. Int. Res.* 59, 1 (2017), 103–132.

[7] Luis M. de Campos, Andrés Cano, Javier G. Castellano, and Serafín Moral. 2011. Bayesian networks classifiers for gene-expression data, In 2011 11th International Conference on Intelligent Systems Design and Applications. *International Conference on Intelligent Systems Design and Applications, ISDA*, 1200–1206. https://doi.org/10.1109/ISDA.2011.6121822

[8] Usama M Fayyad and Keki B Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, Chambery, France, 1022–1029.

[9] Nir Friedman and Moises Goldszmidt. 1996. Discretizing Continuous Attributes While Learning Bayesian Networks. In *Proceedings of the Thirteenth International Conference on Machine Learning*, Lorenza Saitta (Ed.). Morgan Kaufmann, San Francisco, CA.

[10] Seyedmohsen Hosseini and Dmitry Ivanov. 2020. Bayesian networks for supply chain risk, resilience and ripple effect analysis: A literature review. *Expert Systems with Applications* 161 (2020), 113649. https://doi.org/10.1016/j.eswa.2020.113649

[11] Katja Ickstadt, Bjöorn Bornkamp, Marco Grzegorczyk, Jakob Wieczorek, Malik R. Sheriff, Hernáan E. Grecco, and Eli Zamir. 2011. Nonparametric Bayesian Networks. In *Bayesian Statistics 9*. Oxford University Press, Oxford, United Kingdom. https://doi.org/10.1093/acprof:oso/9780199694587.003.0010 arXiv:https://academic.oup.com/book/0/chapter/141642184/chapter-ag-pdf/45787762/book_1879_section_141642184.ag.pdf

[12] Jaime S. Ide, Fabio G. Cozman, and Fabio T. Ramos. 2004. Generating Random Bayesian networks with constraints on induced width. In *Proceedings of the 16th European Conference on Artificial Intelligence* (Valencia, Spain) *(ECAI'04)*. IOS Press, NLD, 353–357.

[13] Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, Cambridge, Massachusetts.

[14] Mariana D.C. Lima, Silvia M. Nassar, Pedro Ivo R.B.G. Rodrigues, Paulo J. Freitas Filho, and Carlos M.C. Jacinto. 2014. Heuristic Discretization Method for Bayesian Networks. *Journal of Computer Science* 10, 5 (2014), 869–878. https://doi.org/10.3844/jcssp.2014.869.878

[15] Zhifa Liu, Brandon Malone, and Changhe Yuan. 2012. Empirical Evaluation of Scoring Functions for Bayesian Network Model Selection. *BMC Bioinformatics* 13 (2012), S14. https://doi.org/10.1186/1471-2105-13-S15-S14

[16] Ngoc Hoang Luong, Han La Poutré, and Peter A.N. Bosman. 2018. Multi-objective Gene-pool Optimal Mixing Evolutionary Algorithm with the Interleaved Multi-start Scheme. *Swarm and Evolutionary Computation* 40 (2018), 238–254. https://doi.org/10.1016/j.swevo.2018.02.005

[17] Kalia Orphanou, Dirk Thierens, and Peter A. N. Bosman. 2018. Learning Bayesian Network Structures with GOMEA. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Kyoto, Japan) *(GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 1007–1014. https://doi.org/10.1145/3205455.3205502

[18] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[19] Casper Reijnen, Evangelia Gogou, Nicole C. M. Visser, Hilde Engerud, Jordache Ramjith, Louis J. M. van der Putten, Koen van de Vijver, Maria Santacana, Peter Bronsert, Johan Bulten, Marc Hirschfeld, Eva Colas, Antonio Gil-Moreno, Armando Reques, Gemma Mancebo, Camilla Krakstad, Jone Trovik, Ingfrid S. Haldorsen, Jutta Huvila, Martin Koskas, Vit Weinberger, Marketa Bednarikova, Jitka Hausnerova, Anneke A. M. van der Wurff, Xavier Matias-Guiu, Frederic Amant, ENITEC Consortium, Leon F. A. G. Massuger, Marc P. L. M. Snijders, Heidi V. N. Küsters-Vandevelde, Peter J. F. Lucas, and Johanna M. A. Pijnenborg. 2020. Preoperative risk stratification in endometrial cancer (ENDORISK) by a Bayesian network model: A development and validation study. *PLOS Medicine* 17, 5 (05 2020), 1–19. https://doi.org/10.1371/journal.pmed.1003111

[20] Rosa F. Ropero, Silja Renooij, and Linda C. van der Gaag. 2018. Discretizing environmental data for learning Bayesian-network classifiers. *Ecological Modelling* 368 (2018), 391–403. https://doi.org/10.1016/j.ecolmodel.2017.12.015

[21] Akbar Rostamabadi, Mehdi Jahangiri, Esmaeil Zarei, Mojtaba Kamalinia, and Moslem Alimohammadlou. 2020. A novel Fuzzy Bayesian Network approach for safety analysis of process systems; An application of HFACS and SHIPP methodology. *Journal of Cleaner Production* 244 (2020), 118761. https://doi.org/10.1016/j.jclepro.2019.118761

[22] Krzysztof L. Sadowski, Dirk Thierens, and Peter A.N. Bosman. 2018. GAMBIT: A Parameterless Model-Based Evolutionary Algorithm for Mixed-Integer Problems. *Evolutionary Computation* 26, 1 (03 2018), 117–143. https://doi.org/10.1162/evco_a_00206 arXiv:https://direct.mit.edu/evco/article-pdf/26/1/117/1547009/evco_a_00206.pdf

[23] Krzysztof L. Sadowski, Dirk Thierens, and Peter A. N. Bosman. 2021. Optimization of multi-objective mixed-integer problems with a model-based evolutionary algorithm in a black-box setting. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 227–228. https://doi.org/10.1145/3449726.3459521

[24] Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics* 6, 2 (1978), 461–464.

[25] Joe Suzuki. 2014. Learning Bayesian Network Structures When Discrete and Continuous Variables Are Present. In *Probabilistic Graphical Models*, Linda C. van der Gaag and Ad J. Feelders (Eds.). Springer International Publishing, Cham, 471–486.

[26] Ting Wu, Hong Qian, Ziqi Liu, Jun Zhou, and Aimin Zhou. 2023. Bi-Objective Evolutionary Bayesian Network Structure Learning via Skeleton Constraint. *Front. Comput. Sci.* 17, 6 (2023), 13 pages. https://doi.org/10.1007/s11704-023-2740-6

[27] G. Zhao, Q. Feng, C. Chen, Z. Zhou, and Y. Yu. 2022. Diagnose Like a Radiologist: Hybrid Neuro-Probabilistic Reasoning for Attribute-Based Medical Image Diagnosis. *IEEE Transactions on Pattern Analysis &; Machine Intelligence* 44, 11 (2022), 7400–7416. https://doi.org/10.1109/TPAMI.2021.3130759

[28] Zhipeng Zhou, Xinhui Yu, Zeyu Zhu, Dequn Zhou, and Haonan Qi. 2023. Development and application of a Bayesian network-based model for systematically reducing safety risks in the commercial air transportation system. *Safety Science* 157 (2023), 105942. https://doi.org/10.1016/j.ssci.2022.105942