Towards Bayesian Network Structure Learning on Quantum Computers Project Report

Lukas Ruettgers

January 14, 2024

1 Introduction

To understand causal relations between the measured features of a population, researchers have introduced graphical models such as Bayesian Networks to decompose the entire joint probability distribution of the data into the salient conditional dependences between the data. A real world motivation for such model to study the impacts of overdoses in nutritional supplements is provided in Appendix Given such conditional independences, learning algorithms optimize the probability distributions to fit the observations from a given dataset.

However, the assumed conditional independences mostly follow human-designed heuristics, because they are hard to learn themselves. Theoretically, there are exponentially many possible structures with different implications on the conditional independences innate to the data. Because this structure learning problem has received quite little attention in research so far, we dedicated our course project to the potential of quantum algorithms to provide a computational speed-up both on current noisy intermediate-scale quantum (NISQ) devices as on more powerful quantum computers in the future.

1.1 My contributions

While I more thoroughly describe our project teamwork in Appendix A, I summarize my independent contributions to the project as follows:

1. The entire work on the quantum algorithm for constraint-based methods (besides the slides for the presentations in our weekly seminars). Theoretical derivation and complexity analysis of a quantum gate-based unitary that models conditional independence tests.
2. I proposed a Hamming approximation of the score Hamiltonian in our QA work that effectively reduced its m-locality to 2-locality and in that way rendered the exponential number of auxiliary qubits required for mapping the Hamiltonian to QUBO superfluous.
3. Original research into prior work, topic proposal, real-world dataset collection
4. Slides and final presentation on classical algorithms, the gate-based adjacency phase and the locality reduction of the score Hamiltonian in the QA section were prepared and held by me.

Moreover, the following contributions are *partially* attributable to me:

1. Review of the theoretical background and classical algorithms for BNSL (together with Pengfei and Ziyi).

2 Problem Formulation

2.1 Bayesian Networks

To study causal relations between features of a dataset, one could model its conditional dependences by a directed acyclic graph, where each feature corresponds to a node. The distribution of each variable Y is fully determined by its parents Γ_Y, that is $P(Y = y \mid \Gamma_Y)$ defines the probability distribution of Y, which is also referred to as the *Markov Condition*. To avoid networks with more edges than necessary, we also require a valid Bayesian network to satisfy the *minimality condition*, which states that each edge in the graph is required to imply at least one conditional dependence in the data.

Note that the direction of edges does not matter when it comes to conditional independences, as Figure 1 demonstrates. If we reverted the edges in a causal chain, the conditional probabilities do not alter. However, flipping the edges of a common cause does indeed matter, as it renders the two children A and C, which are now parents of B, conditionally dependent of B. The latter structure is also known as *v-structure* and marks the only causal class that introduces a conditional independence, while all other classes are equivalent. To better grasp how a BN represents these dependences, the interested reader may refer to an example in Appendix C.1.

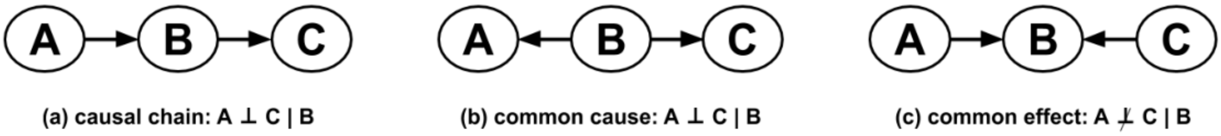

A
B
C
(a) causal chain: A ⊥ C | B
(b) common cause: A ⊥ C | B
(c) common effect: A ⊥̸ C | B

Figure 1: The three causal classes for three-node Bayesian networks. (2)

To determine conditional independences for graphs of arbitrary size, Pearl et al. (6) introduced the notion of *d-separation*. Intuitively, for any $A, B \in V$, a set $S \in V \setminus \{A, B\}$ is said to *d-separate* A and B if all causal paths are blocked by at least one node in S. While this abstract intuition suffices for the scope of the report, a more thorough explanation is provided in Appendix C.2. They also proved that the v-structures and undirected adjacencies inside a graph already fully determine the conditional dependences between the nodes. Intuitively, for each three connected nodes in the network, we can consider their induced subgraph and are only sensitive to whether they form a v-structure or not, because all other subgraphs belong to the same causal equivalence class. The interested reader may refer to Appendix C.3 for a more exhaustive explanation of the proof.

To abstract from specific edges that do not influence these properties, Bayesian networks are usually summarized in equivalence classes. The replacement of directed edges by undirected edges for all edges that are not part of a v-structure leads to what is called a *partially directed acyclic graph* (PDAG). However, PDAGs still not coincide with causal equivalence classes, as some of the edges can only be specified in one direction to not introduce another v-structures. By specifying their direction, we finally obtain a *complete PDAG* (CPDAG) that indeed coincides with an equivalence class, which is exemplified in the appendix (Fig. 6). For this reason, what we actually desire when we seek to determine the conditional independences and solve the BNSL problem is to obtain the adjacencies and the v-structures of the most likely CPDAG.

3 Classical Algorithms

Classical algorithms usually make two assumptions. The first one is the *causal sufficiency* of the network, which means that we assume the absence of any latent variables. The second assumption states that each node may not have more than $m \ll n$ parents to further narrow the solution space.

The classical algorithms mainly fall into two groups, the *constraint-based*, and the *score-based* algorithms. While constraint-based algorithms directly avail to the entire theoretical framework I just introduced before to determine the most likely CPDAG with conditional independence tests, score-based methods formulate the BNSL as an optimization search problem with an objective function. A full picture of the algorithm landscape is drawn in Appendix D.

3.1 Constraint-based Algorithms

To introduce the approach of constraint-based algorithms, recall our example on d-separation in Figure 5. When two variables are adjacent such as **A** and **C**, there will be no d-separating set because there is node that can block the direct edge between **A** and **C**. In fact, the opposite is also true, namely two variables are adjacent if there is no set that can d-separate them (6). We can hence determine the adjacency of to variables A and B by checking whether there is any d-separating set $S \in V \setminus \{A, B\}$ such that $A \perp B \mid S$. The conditional independence can be checked by verifying the null hypothesis $H_0 : A \perp B \mid S$ with any statistical test on the dataset. For simplicity, let us assume we have an dataset of N entries over n binary random variables. For any $A, B \in V$ and $S \in V \setminus \{A, B\}$, we denote by $N_{a,b,s}$ the absolute frequency of entires where $A = a, B = b$ and $S_i = s_i$ for all $S_i \in S$. Note that $N = \sum_{a=0}^{1} \sum_{b=0}^{1} \sum_{s=(0,\dots,0)}^{(1,\dots,1)} N_{a,b,s}$ and $\hat{P}(A = a, B = b, S = s) = \frac{N_{a,b,s}}{N}$ for the empirical estimation $\hat{P}$ of the true unknown probability density function P. A commonly used statistical test is the χ^2 test, where

$$\chi^2 = 2 \cdot \sum_{a=0}^{1} \sum_{b=0}^{1} \sum_{s=(s_1,\dots,s_q) \in \{0,1\}^q} \frac{(Expected - Observed)^2}{Expected},$$

$$Observed = N_{a,b,s} = N \cdot P(a, b \mid s), \quad Expected = \frac{N_{a,s} \cdot N_{b,s}}{N_s} = N \cdot \frac{P(a \mid s) \cdot P(b \mid s)}{P(s)}.$$

The stochastic nature of the outcome of this test reflects that our decision whether A and B are adjacent is always subject to statistical uncertainty, which underscores the usefulness of approximate algorithms close-to-optimal solution still exhibit a high likelihood of having generated the data.

After having determined the adjacencies, it remains to determine the v-structures. Clearly, we only have to consider those A, B, C where A and C are not adjacent but B is adjacent to both. Then, we need to check whether B is a source for the conditional dependence of A and C regardless of conditioning on other variables. Pearl et al. (6) showed that this is equivalent to $A \not\perp C \mid B \cup S$ for *all* $S \in V \setminus \{A, B, C\}$, which we can again check using CI tests. Actually, we don't have to conduct these CI tests again and can reuse the information on the d-separating sets of A and C. If B is not part in any of them, then the above statement holds.

In this way, we have obtained a PDAG and just have to explicitly specify the direction of edges that are already implicitly determined such as in Figure 6. This is almost exactly what the Spirtes-Glymour-Scheines (SGS) algorithm (1) does, except that it does not avail to the aforementioned trick to reuse the information from the adjacency phase to avoid another round of CI tests. Its overall complexity amounts to $O\left(n^2 \cdot 3^n\right)$ and the exact calculation is available in Appendix D.1, where I also summarized the pseudocode (Fig. 8).

The PC algorithm (5) improves upon this complexity by seizing the aforementioned trick and hence saves the v-structure phase. By further availing to efficient node orderings, early edge removals, and incorporation of knowledge from prior CI tests, it reduces the complexity to $O\left(n^2 \cdot \frac{(n-1)^{s_{max}-1}}{(s_{max}-1)!}\right)$. This complexity depends on the largest size of any d-separating set $s_{max} \leq n - 2$, which is hard to compare with the closed term before but in general still retains the exponential worst case complexity bound (2). The algorithms that were introduced in the subsequent years did mostly focus on loosening assumptions, such as considering hidden variables, allowing multi-modal distributions, or restricted themselves to determining a local subgraph of the entire DAG. Figure 9 visualizes the step-wise process in which the SGS algorithms and its descendents determine the final CPDAG.

3.2 Score-based Algorithms

While constraint-based algorithms iteratively construct the most likely causal equivalence class, score-based algorithms initialize a specific DAG and then traverse a search space of DAGs by either adding, removing, or reversing edges, swapping two nodes or performing other permutations under that the

DAGs remain closed. At each point, they evaluate the current DAG with an objective function that shall assess the likelihood of this graph to have generated the observed data. To remain consistent across several DAGs within the same equivalence class, an objective function should ideally assign the same score to representatives of the same class. This property is referred to as *score-equivalence* and is satisfied by most of the commonly used scores. Another property that greatly affects the computational efficiency of the search is the *decomposability* of the objective function into local terms that are each associated with a particular node in the graph (2). In this way, the aforementioned traversal steps in the search space will only require recomputing the scores for the nodes that were subject of the permutation applied in the traversal step.

The commonly used objective functions divide up into two major groups. The first groups are the Bayesian scores which enables to incorporate prior beliefs or inductive biases into the objective function. The second group comprises information-theoretic scores that allow to penalize the complexity of the current graph G. This penalty is crucial because in contrast to the constraint-based methods which explicitly ensured the minimality condition of the CPDAG, the mere optimization search does not inherently prefer more minimal DAG solutions over more complex ones. It is to the objective function to consider such aspects.

Given a naïve encoding of the DAG through its adjacency matrix, the size of the search space amounts to $2^{\Omega(n^2)}$ (2). When representing a DAG by the topological node ordering it implies, the search space size reduces to $2^{n\log(n)}$, which is however still super-exponential in n.

While a quantum formulation of constraint-based methods is more unclear, the resemblance of score-based algorithms to optimization search quantum algorithms gives natural rise to a transfer to quantum annealing or QAOA approaches as was already done in (3) and (4) respectively.

4 Quantum Algorithms

In the following section, I will firstly present and discuss a quantum formulation for the constraint-based algorithms. After that, I will review how (3) transferred the score-based algorithms to a quantum annealing instance and propose improvements on their computational complexity.

4.1 Gate-based Adjacency Phase

The computational burden of the constraint-based algorithms mainly originates from the adjacency phase, which in the worst case includes conducting an exponential number of CI test, which in turn roughly require an exponential number of arithmetic operations. Naturally, the question arises whether we can exploit the principle of state superposition in quantum systems to run this CI test in parallel for all $S \in V \setminus \{A, B\}$. For an arbitrary, but fixed A, B, we could represent each $X \in V \setminus \{A, B\}$ by a qubit $|q_X\rangle$. Then, $q_X = 0$ would indicate $X \notin S$ and $q_X = 1$ represented $X \in S$ vice versa. If we managed to formulate a comparably efficient unitary U_{CI} that conducted the CI test *uniformly* for a variable set S, then we could avail to the same strategy as Grover's algorithm and execute the unitary for the superposition of all possible states. By the aforementioned theorems of Pearl et al. (6), we then only needed to check whether the unitary is *constant*, which we could verify or disprove with $O(\sqrt{2^{n-2}})$ repeated executions. To access our dataset, consider the absolute frequency $N_{X_1=x_1,\dots,X_n=x_n}$ stored in a dedicated qubit $|q_{x_1,\dots,x_n}\rangle$. In the general case, this need for exponentially many storage units is unavoidable if we do not make further sparsity assumptions on the dataset, which however limits the applicability to real-world datasets.

Recall that the χ^2 test requires computing $N_{a,b,s}, N_{a,s}, N_{b,s}$ and N_s for each possible $A = a, B = b$, and $S = s$. If we have computed $N_{a,b,s}$, the latter three are easily obtained in constant number of additions. Finally, the terms just need to be summed up and compared with the rejection threshold to eventually output 0 or 1.

4.1.1 Computing $N_{a,b,s}$ for variable S

The complexity of our unitary therefore mainly depends on how we compute the $N_{a,b,s}$. To that end, visualize the data $N_{a,b,x_1,\ldots x_{n-2}}$ stored in an n-dimensional tensor T. On variable input $|q_{X_1}\rangle, \ldots, |q_{X_{n-2}}\rangle$, we could independently check $q_{X_i} = 1$ for each i. If this holds, then $X_i \in S$, so we must distinguish $X_i = 0$ or $X_i = 1$ in the final χ^2 term. However, if $q_{X_i} = 0$ and hence $X_i \notin S$, we could sum up all values along this dimension to the 0 index of X_i. After all absolute frequencies have been summed up along the dimensions i with $X_i \notin S \cup \{A, B\}$, each $N_{a,b,s}$ can be accessed by setting all other indices to 0.

The above additions require integer arithmetic on the quantum circuit, which we fix over a field $\mathbb{F}_{2^k-1}$. The absolute frequencies in $[0, \ldots, N]$ could be scaled down and rounded to $\mathbb{F}_{2^k-1}$ by preprocessing on classical computers. If we provide k qubits for each element in this imaginary tensor, then we can surely avoid any overflow during the aforementioned computations. Appendix E.1.2 provides a more exhaustive consideration of integer arithmetic and numerical stability.

This will necessitate $(n-2)\cdot 2^{n-1}$ additions in the worst case and this algorithm directly transfers to our unitary formulation and is again summarized in Figure 10. In total, this conditional computation includes at most $(n-2)\cdot 2^{n-1}$ additions and therefore $O(k\cdot(n-2)\cdot 2^{n-1})$ gates, which mainly comprise CNOT gates, as I describe more thoroughly in Appendix E.1.3.

Having obtained $N_{a,b,s}$, $N_{a,s}, N_{b,s}$ and N_s can be sequentially computed for each a, b, s, which necessitates only a constant number of further *workbench* qubits and unavoidably $O(2^n)$ further additions in the worst case. Equivalence transformations that are detailed in Appendix E.1.2 reveal that the computation of the χ^2 simplifies to the computation of $\frac{N_s N_{a,b,s}^2}{N_{a,s}\cdot N_{b,s}}$. Evaluating this term in the numerically most stable way will require each workbench qubit unit to comprise $3k$ qubits.

4.1.2 Resource Estimation

All in all, $n-2$ qubits encode the current subset S. From the derivation in the last chapter, $k \cdot 2^n$ data qubits are necessary to store the preprocessed data. Finally, we use a constant number of qubits to perform the intermediate computations for each a, b, s and then repeatedly add the result term to another storage that tracks the overall sum. We need $O(k)$ of these workbench qubits.

Regarding the number of gates, the computation of $N_{a,b,s}$ can be done by conditioning the addition of each $N_{i_1\ldots i_n}$ on a CNOT gate. Because addition of k bit values requires $O(k)$ CNOT gates, this procedure entails $O(2^{n-1}\cdot n\cdot k)$ CNOT gates in total. Afterwards, we sequentially iterate over all possible value combinations in $s \in \{0,1\}^{n-2}$, check whether $s_j = 0$ for all $s_j \notin S$ and then conduct the three multiplication and one division operation with $O(k^2)$ gates each. For conditioning these arithmetic operations on whether $s_j = 0$ holds for all $s_j \notin S$, we unfortunately require Toffoli gates of up to $n-2$ qubits, which will remain extremely costly to map to quantum hardware even in upcoming decades.

Overall, the quantum circuit for the CI unitary requires roughly $O(n \cdot k^2 \cdot 2^n)$ gates. For the reasonable assumption that $k \in O(n)$, this culminates in $O(n^3 \cdot 2^n)$ gates. A schematic overview of the unitary and its complexity is drawn in Figure 11. The $O(2^{0.5n})$ repeated evaluations of this unitary for Grover's search finally scales this complexity further. Figure 12 juxtaposes both the classical algorithm with our unitary and compares their complexity in each step.

4.1.3 Complexity comparison with classical algorithms

While the above unitary achieves a slight advantage in case of the tedious execution of each CI test in the SGS algorithm, the node orderings, and early edge removals used in the PC algorithm reduces the classical complexity down to $O\left(n^2 \cdot \frac{(n-1)^{s_{max}-1}}{(s_{max}-1)!}\right)$, which seems to provide a better complexity upper bound in practice than our gate-based quantum algorithm, especially when we assume a fixed maximum number of parents m. However, it is fair noting that the early edge removal in the PC algorithm is more error-prone than the SGS algorithm, because falsely removed edges in earlier CI tests can disturb the correctness of later CI tests (2). But the stochastic nature of BNSL still makes this instability acceptable in practical applications. For this reason, a theoretical quantum improvement over the constraint-based algorithms was not achieved. However, it was an enriching challenge to design such an algorithm and investigate potentials for complexity improvements.

4.2 Score-based Quantum Annealing

Since score-based methods already accord to the spirit of an optimization search, its transfer to a Quadratic Unconstrained Binary Optimization (QUBO) problem for simulated annealing or quantum annealing leads over a solidly paved path. Naïvely, one could directly encode the adjacency matrix of a DAG by binary variables d_{ij}, but more sophisticated encodings that avail to the implicit topological node ordering of a DAG can reduce the amount of variables from $O(n^2)$ to $O(n \log n)$. In the classical score-based method, we could ensure that the traversal steps remain closed under the subclass of DAGs and therefrom ensure that the current candidate graph in question is always a valid DAG. In annealing, we do not have this freedom and must ensure in other ways that optimal solutions will not violate the constraints of a DAG. One could verify the DAG condition by adding more *topological* variables rij that encode whether $i < j$ in total topological ordering the DAG implies. The QUBO formulation in (3) availed to this very encoding. Firstly, we introduce their approach, and then suggest improvements to reduce the required number of qubits.

4.2.1 Original Hamiltonian

In total, the Hamiltonian that the authors define writes as

$$H(\mathbf{d}, \mathbf{y}, \mathbf{r}) = H_{\text{score}}(\mathbf{d}) + H_{\text{max}}(\mathbf{d}, \mathbf{y}) + H_{\text{trans}}(\mathbf{r}) + H_{\text{consist}}(\mathbf{d}, \mathbf{r}),$$

where $H_{\text{score}}(\mathbf{d})$ represents *any* decomposable score $H_{\text{score}}(\mathbf{d}) = \sum_{i=1}^{n} H^{(i)}_{\text{score}}(\mathbf{d})$ of the graph that is encoded by the adjacency variables $\mathbf{d}$.

To penalize graphs that are no valid DAGs, the Hamiltonian terms $H_{\text{trans}}(\mathbf{r})$ and $H_{\text{consist}}(\mathbf{d}, \mathbf{r})$ check whether the topological variables $\mathbf{r}$ encode a transitive order that is consistent with the adjacency variables $\mathbf{d}$.

Last but not least, the authors avail to the well-known assumption of a constant maximum number m of parents each node may have, and penalize graphs that exceed this limit by $H_{\text{max}}(\mathbf{d}, \mathbf{y})$. To that end, they introduce additional slack variables $\mathbf{y}$, where each slack variable set $0 \leq \mathbf{y}_i \leq m$ encodes a non-negative integer. While $\mathbf{y}_i$ can close the gap between the in-degree d_i of node i and the upper bound m when $d_i \leq m$, this is not possible in the converse case as $\mathbf{y}_i$ can not take negative values.

All in all, this QUBO encoding comprises $\frac{3n(n-1)}{2} + n\log_2(m+1) = \underbrace{n(n-1)}_{\mathbf{d}} + \underbrace{\binom{n}{2}}_{\mathbf{r}} + \underbrace{n\log_2(m+1)}_{\mathbf{y}}$ variables. The interested reader may refer to the exact definitions of the Hamiltonian terms in Appendix E.2. There, we will also discuss lower-bounds on the penalty weights to ensure that the ground energy state truly encodes a valid DAG (Appendix E.2.5).

4.2.2 Reducing m-locality to any $k < m$

By definition of decomposability, we desire that the score term for each node i satisfies $H^{(i)}_{\text{score}}(\mathbf{d}) \equiv H^{(i)}_{\text{score}}(\Gamma_i) \equiv s_i(\Gamma_i)$, where s_i is the decomposable score function and Γ_i the parent set of i encoded by $\mathbf{d}$. Instead of a naïve n-local formulation, the authors availed to the *inclusion-exclusion principle* to realise this term with m-locality. They do only verify the former condition, namely if each $j \in J$ is a parent of i by expanding the weight of each set by the weights of all its subsets with alternating sign such that the terms of all subsets eventually cancel out and satisfy the overall condition

$$H^{(i)}_{\text{score}}(\mathbf{d}) \equiv \sum_{\substack{J \subset V \setminus \{i\} \\ |J| \leq m}} w_i(J) \underbrace{\prod_{j \in J} d_{ji}}_{J \subset J^*} \equiv s(J^*),\ w_i(J) = \sum_{l=0}^{|J|} (-1)^{|J|-l} \sum_{K \subset J, |K| = l} s_i(K).$$

However, m-locality still necessitates $O(n^m)$ auxiliary variables to map this term to a QUBO instance. In view of the limited resources of NISQ devices, I wondered whether we could achieve a sufficiently accurate approximation of the score Hamiltonian that could reduce the locality of the Hamiltonian down to any $k < m$. My idea was to allow each set $J \in V \setminus \{i\}$ to contribute to the overall score term of the true parent set J^*, but weight each score $s(J)$ by the resemblance to J^* in the 1-local *hamming distance* $d(J, J^*) := n - 1 - \sum_{j \in J} d_{ji} + \sum_{j \notin J}(1 - d_{ji})$.

For any $1 < k$, we can now scale the score of each J with a weight $\left(\frac{(n-d(J,J^*))}{n}\right)^k$ that decreases polynomially in k with increasing distance $d(J, J^*)$ from the true parent set J^* and finally obtain our Hamming approximation of the original score term

$$\hat{H}^{(i)}_{\text{score}}(d_i) \equiv \sum_{J \subset V \setminus \{i\}} \left(\frac{(n - d(J, J^*))}{n}\right)^k \cdot s_i(J).$$

Here, k should be chosen to obtain an optimal trade-off between preserving the approximation accuracy of the original score term and improving upon the efficiency of the required variables for the QUBO formulation. Because the number of sets with Hamming distance d grows roughly exponentially in n up to $d = \frac{n}{2}$, choosing k too small might give distant parent sets a too strong influence on the score for the actual parent set. We compare different choices of k with the original term and show that this approximation also smoothens the optimization landscape.

4.2.3 Regularization alternatives to H_{max}

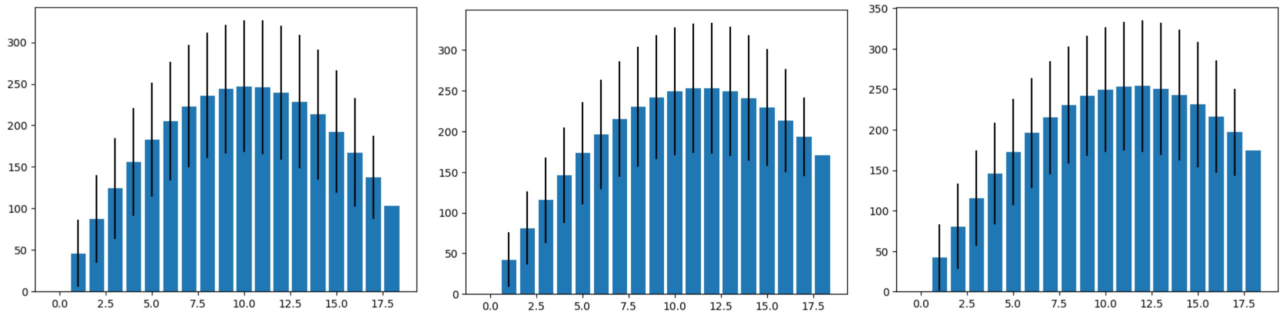
To further reduce the required number of variables to map the BNSL to QUBO, we can also replace the slack variables $\mathbf{y}$ by a *regularization* term $H_{\text{norm}} = \lambda \|d\|_k$ that penalizes the graph complexity with a hyperparameter λ, where $\|d\|_k = \left[\sum_{i,j} |d_{ij}|\right]^k$ is the 1-norm of the graph's adjacency matrix to the power of our locality parameter k.

5 Experiments

To compare the Hamming approximation with the original score Hamiltonian, we collected 200 samples out of a toy data set with $n = 4$ binary variables. Instances of $n = 4$ were the largest we could achieve. Problem instances of $n = 5$ proved to be too hard for current quantum hardware simulators, as the number of qubits amounts to 45 and is further increased by the number of required auxiliary variables $O(n^m)$ to map the originally m-local term to a QUBO instance.

Firstly, we observed how the landscape of the objective function changes when approximating the exact score term H_{score} by $\hat{H}_{\text{score}}$ for both $k = 1, 2$. Secondly, we executed the quantum annealing optimization for all three score formulations and compared the quality of the final optimal solutions obtained. To better visualize statistics in the high-dimensional space of QUBO variables, we formulated two distances metrics to which we projected the space:
Spin distance: The spin distance measures the hamming distance across *all* variable groups $\mathbf{d}, \mathbf{r}, \mathbf{y}$.
DAG distance: Here, the hamming distance is merely measured between the adjacency variables $\mathbf{d}$.

We then grouped all valid variable configurations by their DAG distance and spin distance to the true, optimal solution and computed the average, standard deviation, and minimum energy with regard to the objective function. Figure 2 shows how the standard deviation and the mean differs between the true Hamiltonian and their Hamming approximations. While the 1-local approximation becomes inaccurate when we approach large distances from the ground state 2a, the 2-local approximation still achieves a good approximation in this domain and exhibits only a slightly increased standard deviation for extreme distances 2c. This is reasonable because the most graphs gather in the middle and hence better approximate the overall variance, while more extreme distances are only obtained for few graphs whose statistics are hence subject to larger variance. The plots of the minimum energy demonstrate a similarly good approximation quality for $k = 2$ (Fig. 16) and can be viewed in the Appendix F along



(a) 1-local $\hat{H}_{\text{score}}$

(b) Original H_{score}

(c) 2-local $\hat{H}_{\text{score}}$

Figure 2: Average scores for parent sets J grouped in increasing order by their spin distance to the true parent set J^*. The black bars indicate the standard deviation of the scores in each set.

with other figures omitted here for compactness. For the DAG distance, the Hamming approximation also smoothens the solution landscape and gradually decreases the energy towards the ground state, which we believe to evoke a more stable optimization procedure as it removes local minima (Fig. 13).

To demonstrate the sensitivity of the solution distribution to the regularization parameter λ when replacing the slack variables $\mathbf{y}$ by our regularization term, we executed our algorithm with the 2-local Hamming approximation score on toy data for different λ and juxtapose the optimal results in Figure 15. The optimal result for $\lambda = 0$ (Fig. 15b) suggests further conditional dependences that are not present in the ground truth equivalence class (Fig. 15a), as it is not punished for additional model complexity. A slight increase to $\lambda = 1$ already seems to punish the graph too strongly for proposing a direct conditional dependence between 2 and 3 (Fig. 15c). This underscores the necessity of fine-tuning the penalty terms to achieve promising results.

We also executed the original QA algorithm on real world data. To that end, we availed to a dataset that comprises medical records of Intensive Care Unit (ICU) patients in US hospitals, which were collected in accordance to the privacy requirements for personal data. The dataset contains physiological, demographic, and medical information and is publicly available at https://www.kaggle.com/c/widsdatathon2020/data. For our first experiment, we considered the three diseases cirrhosis, diabetes mellitus, and leukemia, between which no correlation has been suggested by medical research. This is also the solution that the original Hamiltonian attributes the lowest energy state to. In 1000 shots, this lowest state was obtained with a noticeable probability of around 1.3% (Fig. 14a), while the second-best state that suggested a conditional dependence between leukemia and diabetes mellitus was measured with a slightly higher probability (Fig. 14b). Another experiment result for four variables is depicted in Figure 18. Our code is publicly available at https://github.com/zoeabcd/BNSL.git and also attached to this report.

6 Conclusion

The current state of quantum hardware and simulators are still too limited to suggest whether our Hamming approximation remains favourable for larger problem instances. The given Hamiltonian also requires long-range interactions between many qubits, which raises the question whether formulations with more sparse embeddings are feasible. Future research could also optimize the qubit encoding from the very beginning and encode a DAG by its implicit topological order and thence reduce the number of required variables to $O(n \log n)$, as more sophisticated score-based algorithms have already done.

References

[1] Clark Glymour, Peter Spirtes, and Richard Scheines. Independence relations produced by parameter values in causal models. *Philosophical Topics*, 18(2):55–70, 1990.

[2] Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of bayesian network structure learning. *Artificial Intelligence Review*, pages 1–94, 2023.

[3] Bryan O'Gorman, Ryan Babbush, Alejandro Perdomo-Ortiz, Alán Aspuru-Guzik, and Vadim Smelyanskiy. Bayesian network structure learning using quantum annealing. *The European Physical Journal Special Topics*, 224:163–188, 2015.

[4] Vicente P Soloviev, Concha Bielza, and Pedro Larrañaga. Quantum approximate optimization algorithm for bayesian network structure learning. *Quantum Information Processing*, 22(1):19, 2022.

[5] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

[6] Thomas Verma and Judea Pearl. Causal networks: Semantics and expressiveness. In *Machine intelligence and pattern recognition*, volume 9, pages 69–76. Elsevier, 1990.

A Thorough description of our teamwork and contributions

My team members were Mingxi Xie, Pengfei Zhu, Ziyi Xie, and Kairui Ding. At the start of the project, I initially reviewed the prior research on quantum algorithms formulations and proposed to dedicate our project to the BNSL problem for two reasons. Firstly, there turned out to be only one paper on a Quantum Annealing (QA) formulation of the BNSL problem (3) and another paper that availed to a Quantum Approximate Optimization Algorithm (QAOA) (4) to solve the problem. Secondly, the problem captivated me when we analysed graphical probabilistic models in our Machine Learning course.

After our group agreed on zeroing in on this topic, I first provided a review paper to drill into the theoretical background of the problem itself and the classical algorithms formulated so far, which are divided into *score-based* and *constraint-based* algorithms. At the beginning, most members of our group only saw potential in the score-based methods, for the aforementioned two papers also grounded their QA and QAOA approach on the optimization problem formulation that identifies the score-based algorithms. However, I was also curious on whether quantum algorithms could achieve any improvements over the constraint-based classical algorithms, because no research was done so far on this problem. We agreed that daring a constraint-based formulation is more prone to failure in contrast to the score-based methods, for which the aforementioned papers already achieve successful formulations. Moreover, the QA paper did not conduct any experiments on their Hamiltonian for lack of powerful quantum hardware and hence allowed more room for further contributions of our group.

Therefore, we first decided to split up into two groups, to allow me to independently explore the more constraint-based methods, while the others exploited the given foundation of the score-based methods to obtain results with a higher chance of success. During my exploration of the constraint-based methods, I independently formulated a quantum-gate based circuit for a unitary that models the conditional independence tests, which constitute the core of the constraint-based methods. During our weekly in-class seminars, Kairui helped me to create the slides for the presentations. I eventually culminated in a unitary that theoretically exhibited a significant advantage over a baseline classical algorithm, but still lacked behind a more advanced classical approach. Our group therefore decided to again join forces to advance our contribution to the Quantum Annealing formulation of the BNSL problem.

At that point, my teammates managed to reimplement the Hamiltonian proposed in the QA paper (3) and ran some toy experiments on it. However, we have in that sense only reproduces work that was already done by other researches and did not achieve own contributions on the quantum formulation of score-based methods. I decided to improve upon the major efficiency drawback that the authors already stressed in the conclusion of their paper, namely the m-locality of their Hamiltonian. To map this Hamiltonian to a Quadratic Uncontrained Binary Optimization (QUBO) problem, this would roughly entail additional auxiliary qubits whose number is exponential in the upper bound of allowed parents per node.

I came up with an approximation scheme of the Hamiltonian that could reduce the m-locality to any $k < m$ and hence achieve 2-locality without any additional qubits. As Pengfei verified in experiments, this approximation indeed seems to coarsely preserve the optimality conditions of the original Hamiltonian and also has the nice side-effect to smoothen the solution landscape. For the final experiments, I also researched into real-world datasets and provided a dataset from US hospitals along with example problem instances to the implementation team.

A.1 Independent contributions

To wrap up, the following contributions were achieved independently by myself:

1. I initially researched into the prior work on quantum algorithms for BNSL and proposed to dedicate our project to this problem.

2. The entire work on the constraint-based algorithms besides the slides for the presentations in our weekly seminars. Specifically, I derived a quantum gate-based unitary that models conditional independence tests and could greatly reduce the computational complexity of the classical constraint-based algorithms. To that end, I estimated its resources, and analysed complexity lower bounds of classical constraint-based algorithms to assess whether there is an actual theoretical improvement in the complexity.

3. In our work on the score-based methods, I proposed an approximation of the score Hamiltonian that effectively reduced its m-locality to 2-locality and in that way rendered the exponential number of auxiliary qubits required for mapping the Hamiltonian to QUBO superfluous. Furthermore, this approximation still preserved the optimality conditions of the original Hamiltonian up to small deviations and even smoothened the optimization landscape, which could stabilize the optimization progress.

4. I researched into real-world datasets and provided a dataset from US hospitals along with example problem instances to let the implementation team conduct experiments on small real-world instances.

5. For the final presentation, the slides on classical algorithms, the gate-based adjacency phase and the locality reduction of the score Hamiltonian in the Quantum Annealing section were prepared and held by me.

A.2 Partial contributions

The following contributions are *partially* attributable to me:

1. Review of the theoretical background and classical algorithms for BNSL (together with Pengfei and Ziyi).

2. Presentation slides on our progress in the quantum formulation of constraint-based classical algorithms during the weekly seminars (together with Kairui).

A.3 Others' contributions

The following contributions are *not* attributable to me:

1. Implementation of an experiment setup to reproduce the Hamiltonian in (3) (Pengfei, Ziyi).
2. Experimental evaluation of the original Hamiltonian and the Hamming approximation Hamiltonian on toy data and real-world data (Pengfei, Ziyi).
3. Derivation of new lower bounds on the penalty weights for the Hamming approximation Hamiltonian (Kairui).
4. The replacement of the y qubits that limit the number of parents by a regularization term that further reduces the number of qubits (Pengfei).
5. Other contributions of our group that I did not mention.

B Motivation of BNSL with a real-world problem

Let us introduce the motivation of Bayesian networks with a concrete example. To better understand the causal relations between diseases and the physical status of humans, hospitals have collected medical records of their patients that comprise not only the diseases they were diagnosed with, but also information related to their demographic background, lifestyle, and physiological health. Having these observations gathered in a dataset with n features, we can directly compute an empirical estimate of the joint probability distribution P that represents which feature value combinations occur with what probability. But what we are really interested in is how several features influence each other. Well-known causal relations are the influence of excessive alcohol consume on cirrhosis, the influence of smoking on cancer or the influence of unhealthy nutrition with high amounts of short-chained sugars and saturated fatty acids on diabetes mellitus. But for many other combinations of diseases, physiological conditions and habits, the causal relations are not well understood. One example of practical importance which I want to motivate more thoroughly relates to overdoses in nutritional supplements. It is not uncommon to avail to such supplements to ensure that the body receives a sufficient amount in certain vitamins, amino acids, or minerals. Even if consumers do not actively choose to consume such supplements, many processed foods and in some countries even plain products such as bread are artificially enriched in some vitamins such as vitamin D. Another is example is the concentrate that we feed livestock with is mostly enriched in vitamin B12 because animals other than cows can not produce this vitamin by themselves. In view of a growing amount of people that reduce their consume of animal products and therefore lack of any significant source of vitamin B12, the market of nutritional supplements is rapidly expanding. Another trend that pushes this market is that people who regularly work out in the gym tend to consume protein powder, creatine, or other supplements that enlarge their muscle growth, regeneration, or expand their energy sources. However, many of these nutrient supplements contain far more than the daily suggested amount of nutrients. This is beneficial for people who are completely depleted of some nutrients and need to be treated medically, but for people of normal condition, it is unclear whether these continuous exposition to overdoses in specific nutrients entail any harm to the body. While vitamins of the B family are soluble in water and is excreted by the human body if provided too much, this is not the case for other vitamins and must necessarily not apply for other substances besides nutritional supplements.

C In-depth introduction to Bayesian Networks

C.1 Conditional independences in example Bayesian Networks

To better understand the relation between parenthood in a Bayesian network and the conditional independences between the features, some Bayesian networks, where each of them implies different conditional independences, are displayed in Figure 3. In the leftmost Bayesian network, the blood pressure depends on all remaining variables. But it suffices to condition its probability distribution only on the heart rate, because the heart rate again is affected by the creatine levels and the blood lipid levels. The creatine levels and the blood lipid levels therefore merely have an indirect effect on the blood pressure. This indirect conditional dependence chain is also referred to as *causal chain*. Similarly, the distribution of the heart rate is conditioned on both creatine and blood lipids. The latter two parents are in general conditionally independent when we have not yet observed any feature of a patient. But when we measure the blood rate of a patient and receive a specific value, then creatine and blood lipids are generally not conditionally independent anymore.

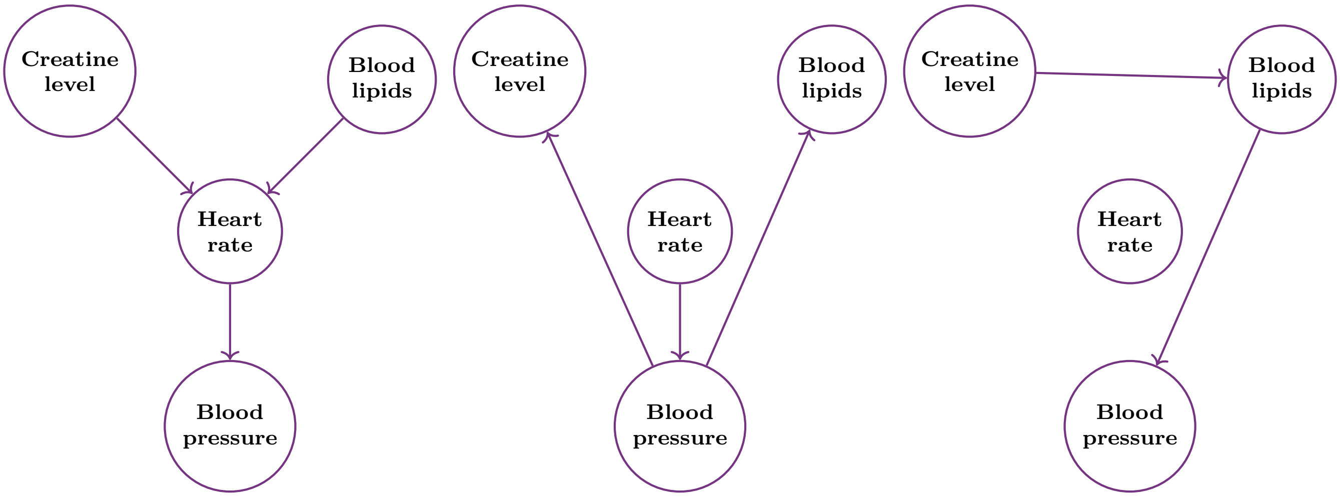
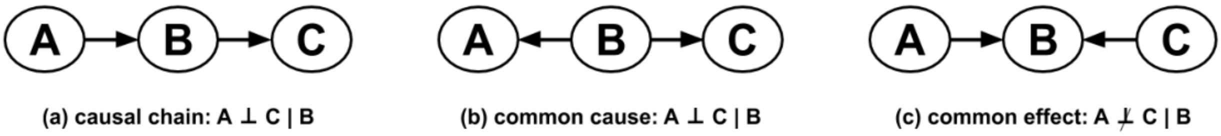


Figure 3: Bayesian networks model conditional independences between features.

To illustrate this on a simplified example, assume that the heart rate is only high if the creatine level is high and the blood lipid levels are low or vice versa. If both levels are high or low, the heart rate will be low with overwhelming probability. In our Bayesian network, whether the blood lipid levels are high or low is independent of the creatine levels if we have no prior knowledge of the patient. But as soon as we find out that the heart rate is high, we know that with overwhelming probability, either the creatine level is high and the blood lipids are low, or conversely the creatine level is low and the blood lipids are high. But we can not write the distribution of the two values as separate, their distributions depend on each other. Otherwise, this would imply that the event that blood lipids are high and the creatine level is low holds with the same probability as the event in which blood lipids are high and the creatine level is high too. But this is not the case, as the latter is far more unlikely than the former. This relation is also referred to as *common effect*.

In the rightmost network, the heart rate is isolated from the other features. Learning the heart rate will not affect the conditional dependences between the other variables. If we finally assume causal relations among the features as depicted in the middle Bayesian network, the dependences become slightly different. If we now learn the heart rate of the patient, the creatine and blood lipids levels remain independent. Although they both originate from a causal chain that emerges from the heart rate, the causal chain splits up at the blood pressure node. Formulated less abstractly, the probability distribution of the blood lipids and the distribution of the creatine levels are separate, even if they are both conditioned on the blood pressure. This third causal relation is defined as *common cause* and completes all three causal classes between three nodes, which are again summarized in Figure 4.

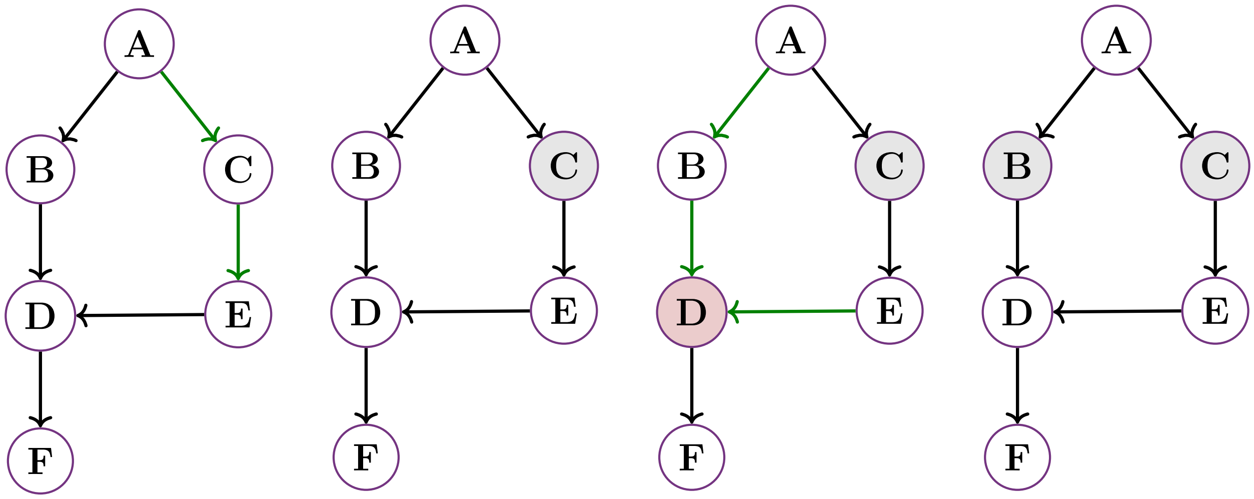


(a) causal chain: $A \perp C \mid B$ **(b) common cause: $A \perp C \mid B$** **(c) common effect: $A \not\perp C \mid B$**

Figure 4: The three causal classes for three-node Bayesian networks. (2)

C.2 Iteratively determining d-separating sets

To determine conditional independences for graphs of arbitrary size, Pearl et al. (6) introduced the notion of *d-separation*. Intuitively, for any $A, B \in V$, a set $S \in V \setminus \{A, B\}$ is said to *d-separate* A and B if all causal paths are blocked by at least one node in S. Here, a causal path can be thought of as a directed causal chain that renders B dependent of A or vice versa, but can also traverse v-structures, as I illustrated in Figure 5. There, the causal path between **A** and **E** that goes over **C** can be easily blocked by adding **C** to the d-separating set. We could actually stop from here as we already obtained a d-separating set **C**. The path over D is not a linear path and hence introduces no conditional dependence similar to the prior examples with v-structures in Figures 3 and 1. But as we further condition on **D** or **F**, this causal path becomes active in a sense and requires **B** to be added to the d-separating set to block it.



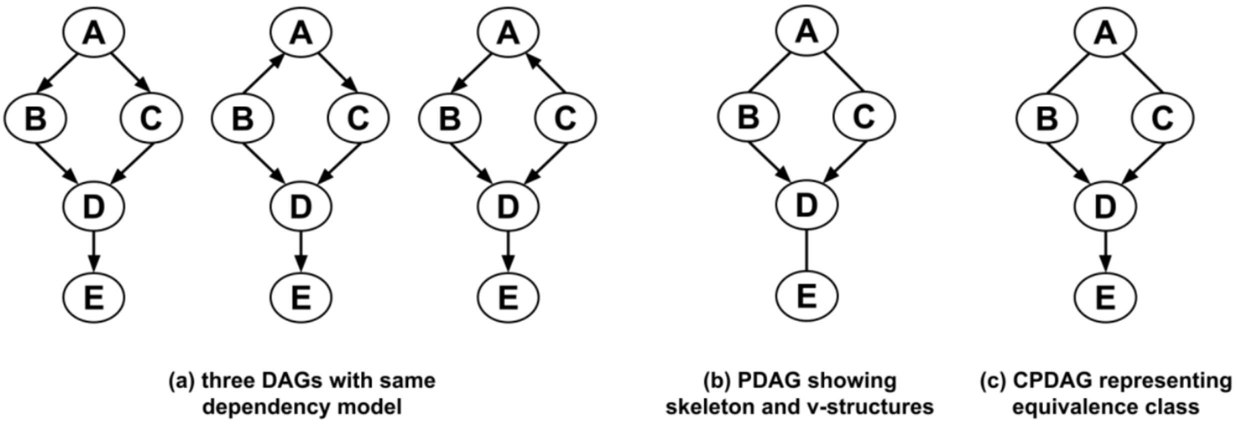
$\mathbf{A} \not\perp \mathbf{E}$ $\quad$ $\mathbf{A} \perp \mathbf{E} \mid \mathbf{C}$ $\quad$ $\mathbf{A} \not\perp \mathbf{E} \mid \mathbf{C}, \mathbf{D}$ $\quad$ $\mathbf{A} \perp \mathbf{E} \mid \mathbf{B}, \mathbf{C}$

Figure 5: Iteratively determine all d-separating sets of **A** and **E** from the left to the right. Nodes marked in red are the ones which are added to the conditioned set to violate the conditional independence, while the green arrows indicate the causal path that is not yet blocked. In this example, all possible d-separating sets are $\{\mathbf{C}\}, \{\mathbf{C}, \mathbf{B}\}$, and all $\{\mathbf{C}, \mathbf{B}\} \subseteq S \subseteq V$, because **B** and **C** already block all paths. The design of this figure was inspired from (2).

C.3 Explaining the sufficiency proof of CPDAGs

In the infancy of Bayesian networks, Pearl et al. (6) already proved that the v-structures and undirected adjacencies inside a graph already fully determine the conditional dependences between the nodes. Intuitively, for each three connected nodes in the network, we can consider their induced subgraph. There have to be at least two edges in this induced subgraph, because else they would not be connected. On the contrary, they can not have more than three edges. If three edges are present, then all variables are conditionally dependent on each other as there is a direct edge between all two pairs of them. As the specific direction of each edge does not matter in this case, we can simply replace the directed edges by undirected ones. In the case where there are only two edges, we are only sensitive to whether they form a v-structure or not, because all other subgraphs belong to the same causal equivalence class.

To abstract from specific edges that do not influence these properties, Bayesian networks are usually summarized in equivalence classes. The replacement of directed edges by undirected edges for all edges that are not part of a v-structure leads to what is called a *partially directed acyclic graph* (PDAG). However, PDAGs still not coincide with causal equivalence classes, as some of the edges can only be specified in one direction to not introduce another v-structure. Consider the PDAG in Figure 6.



(a) three DAGs with same dependency model

(b) PDAG showing skeleton and v-structures

(c) CPDAG representing equivalence class

Figure 6: Transition from a particular DAG to a partially directed acyclic graph (PDAG) and finally a complete PDAG (CPDAG). (2)

Clearly, the edge between D and E must be directed from D to E. In the opposite case, it would introduce two new v-structures. But these v-structures are not present in this graph, otherwise this edge would not have been replaced by an undirected one according to the definition of PDAGs. To achieve a graph that coincides with an equivalence class, we must specify the direction of edges like the one between D and E. This way, we finally obtain a *complete PDAG* (CPDAG) that indeed coincides with an equivalence class. For this reason, what we actually desire when we seek to determine the conditional independences and solve the BNSL problem is to obtain the most likely CPDAG. And this again reduces to determining the most likely adjacencies and the v-structures inherent in the CPDAG.

D Thorough overview of classical algorithms

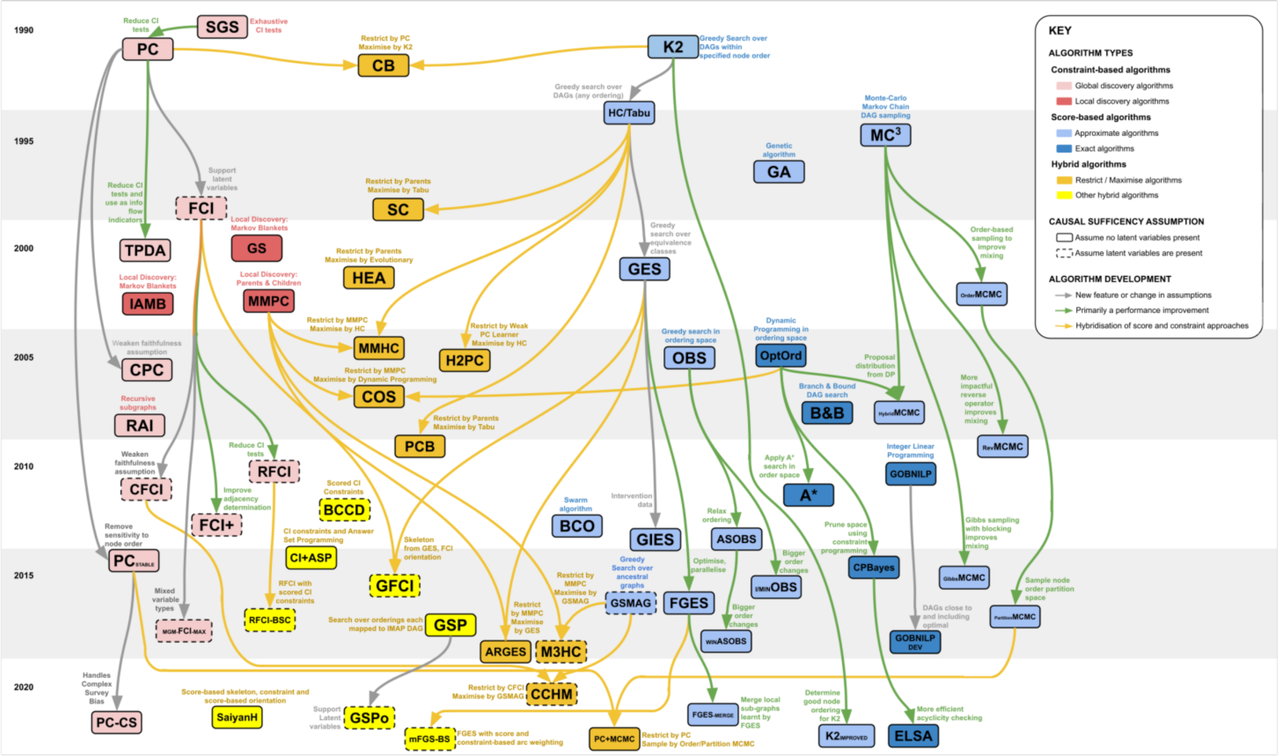

1990
1995
2000
2005
2010
2015
2020
SGS
Exhaustive CI tests
Reduce CI tests
PC
Restrict by PC Maximise by K2
CB
K2
Greedy Search over DAGs within specified node order
Greedy search over DAGs (any ordering)
HC/Tabu
Monte-Carlo Markov Chain DAG sampling
MC³
Genetic algorithm
GA
Support latent variables
FCI
Reduce CI tests and use as info flow indicators
Restrict by Parents Maximise by Tabu
SC
Local Discovery: Markov Blankets
GS
TPDA
Restrict by Parents Maximise by Evolutionary
HEA
Greedy search over equivalence classes
GES
Local Discovery: Markov Blankets
IAMB
Local Discovery: Parents & Children
MMPC
Order-based sampling to improve mixing
OrderMCMC
Restrict by MMPC Maximise by HC
MMHC
Restrict by Weak PC Learner Maximise by HC
H2PC
Greedy search in ordering space
OBS
Dynamic Programming in ordering space
OptOrd
Weaken faithfulness assumption
CPC
Restrict by MMPC Maximise by Dynamic Programming
COS
Proposal distribution from DP
Branch & Bound DAG search
B&B
HybridMCMC
More impactful reverse operator improves mixing
Recursive subgraphs
RAI
Restrict by Parents Maximise by Tabu
PCB
RevMCMC
Weaken faithfulness assumption
CFCI
Reduce CI tests
RFCI
Improve adjacency determination
FCI+
Scored CI Constraints
BCCD
Swarm algorithm
BCO
Intervention data
GIES
Relax ordering
ASOBS
Apply A* search in order space
A*
Integer Linear Programming
GOBNILP
Prune space using constraint programming
CPBayes
Gibbs sampling with blocking improves mixing
GibbsMCMC
Remove sensitivity to node order
PCSTABLE
CI constraints and Answer Set Programming
CI+ASP
Skeleton from GES, FCI orientation
GFCI
Restrict by MMPC Maximise by GSMAG
Greedy Search over ancestral graphs
GSMAG
Optimise, parallelise
FGES
Bigger order changes
I/MINOBS
Sample node order partition space
PartitionMCMC
Mixed variable types
MGM-FCI-MAX
RFCI with scored CI constraints
RFCI-BSC
Search over orderings each mapped to IMAP DAG
GSP
Restrict by MMPC Maximise by GES
ARGES
M3HC
Bigger order changes
WINASOBS
DAGs close to and including optimal
GOBNILP DEV
Handles Complex Survey Bias
PC-CS
Score-based skeleton, constraint and score-based orientation
SaiyanH
Support Latent variables
GSPo
Restrict by CFCI Maximise by GSMAG
CCHM
mFGS-BS
FGES with score and constraint-based arc weighting
PC+MCMC
Restrict by PC Sample by Order/Partition MCMC
FGES-MERGE
Merge local sub-graphs learnt by FGES
Determine good node ordering for K2
K2IMPROVED
ELSA
More efficient acyclicity checking
KEY
ALGORITHM TYPES
Constraint-based algorithms
Global discovery algorithms
Local discovery algorithms
Score-based algorithms
Approximate algorithms
Exact algorithms
Hybrid algorithms
Restrict / Maximise algorithms
Other hybrid algorithms
CAUSAL SUFFICENCY ASSUMPTION
Assume no latent variables present
Assume latent variables are present
ALGORITHM DEVELOPMENT
New feature or change in assumptions
Primarily a performance improvement
Hybridisation of score and constraint approaches

Figure 7: Evolution of classical algorithms for Bayesian Network Structure Learning. (2)

An overview of the landscape of classical algorithms is depicted in Figure 7. The classical algorithms mainly fall into two groups, the *constraint-based*, and the *score-based* algorithms. While the constraint-based algorithms directly avail to the entire theoretical framework I just introduced before to determine the most likely CPDAG with conditional independence tests, score-based methods formulate the BNSL as an optimization problem and avail to the massive framework of optimization algorithms to solve this problem. Their interplay also sets the stage to *hybrid* methods, which use the constraint-based methods to restrict the solution space for the optimization conducted in a score-based algorithm.

While some of the developed algorithms – the one encircled in dashed lines – already ignore the causal sufficiency assumption and consider unknown latent variables, we will focus on the original case and abide by the causal sufficiency assumption. Constraint-based methods also distinguish themselves in whether they try to obtain the entire Bayesian network or just the local *Markov blanket* of one variable, which comprises the parents, children, and siblings of one variable and thenceforth all variables that matter to the conditional independence. The latter ones are depicted in Figure 7 in deep red, while the remaining majority is filled in bright red. Again, we stick with the original problem formulation, that sought to obtain the entire CPDAG.

The score-based methods also divide themselves into approximate and exact optimization algorithms, where the latter usually avail to dynamic programming or branch and bound to remove redundancies in the optimization search. Since approximate algorithms better suit the uncertainty and capabilities innate to quantum computers, we also neglect exact optimization algorithms here.

D.1 Complexity analysis of the SGS algorithm

Because each CI test itself requires $O(2^{2+|S|})$ arithmetic operations, the overall complexity of the SGS algorithm, whose pseudocode is depicted in 8, is mainly determined by the adjacency phase, which amounts to

$$\underbrace{\binom{n}{2}}_{A,B\in V}\cdot\underbrace{\sum_{q=0}^{n-2}\binom{n-2}{q}}_{S\subseteq V\setminus\{A,B\}}\cdot\underbrace{2^{2+q}}_{\text{CI test}}=\binom{n}{2}\cdot 4\cdot\sum_{q=0}^{n-2}\binom{n-2}{q}\cdot 2^{q}\cdot 1^{n-2-q}=4\cdot\binom{n}{2}\cdot 3^{n-2}.$$

```
SGS Algorithm: The origin and prototype of constrained-based methods.
Adjacency phase: For all $A, B \in V$, test if $A \not\perp B \mid S$ for all $S \subseteq V \setminus \{A, B\}$.
V-structure phase: For any $\{\{A, B\}, \{B, C\}\} \subseteq E, \{A, C\} \notin E$:
    Test if $A \not\perp C \mid S$ for all $\{S, B\} \subseteq V \setminus \{A, B, C\}$.
Orientation propagation phase: Transform PDAG to CPDAG by specifying direction of edges where
    possible.
```

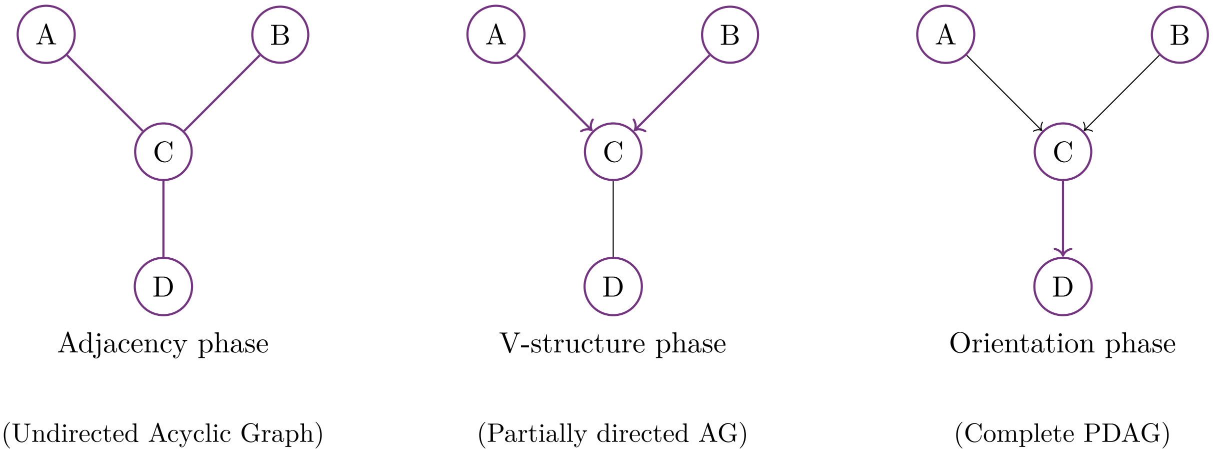
Figure 8: SGS algorithm pseudo code.

Figure 9 visualize the step-wise process in which the SGS algorithms and its descendents determine the final CPDAG.

E Details on Quantum Algorithms

E.1 Gate-based adjacency phase

E.1.1 Computing $N_{a,b,s}$ for variable S



Adjacency phase

V-structure phase

Orientation phase

(Undirected Acyclic Graph)

(Partially directed AG)

(Complete PDAG)

Figure 9: Illustration how the three phases of the SGS algorithm iteratively narrow down the graph structure to culminate in a CPDAG.

```
for all $s_i \in V \setminus \{A, B\}$ $(O(n))$ do
    if $q_i = 0$ then
        for all $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \{0,1\}^{n-1}$ $(O(2^n))$ do
            $|q_{x_1,\dots x_{i-1},0,x_{i+1},\dots,x_n}\rangle += |q_{x_1,\dots x_{i-1},1,x_{i+1},\dots,x_n}\rangle$
        end for
    end if
end for
```

Figure 10: Uniform algorithm to compute $N_{a,b,s}$ for variable input S. The specific order in which we check the values and add up along the dimensions does not influence the final property that we have the total sum available in the tensor element where all indices $i \notin S \cup \{A, B\}$ are set to 0.

E.1.2 Integer arithmetic in the CI unitary with numerical stability

Because each data qubit $|N_{i_1 \dots i_n}\rangle$ could theoretically contain the entire sum of all absolute frequencies N, it would take $\log N$ qubits for each $N_{i_1 \dots i_n}$ to store our data. In general, we have $N \gg 2^n$ because the dataset should ideally contain enough records for all possible value combinations to conduct stable CI tests. To avoid that the required number of qubits depend on this additional factor, one could choose a constant range k and approximate the dataset size-dependent range $[0, \dots, N]$ by scaling each record $N_{i_1 \dots i_n}$ by $\frac{2^{k-1}-1}{N}$ and rounding the result value to the closest integer in $[0, \dots 2^{k-1} - 1]$. This preprocessing stage can be performed on classical computers. Because we need to account for rounding errors that increase the overall sum of $N_{i_1 \dots i_n}$ to a value slightly higher than $2^{k-1} - 1$, we then needed not $k - 1$, but k qubits for each data qubit $|N_{i_1 \dots i_n}\rangle$. This provides us with a controllable constant to steer the approximation accuracy of datasets. Generally, it is reasonable to choose this k in $O(n)$.

Because we can compute each summand in the overall χ^2 term sequentially, we only need five additional qubits to store the intermediate values $N_{a,s}, N_{b,s}$ and N_s for $a, b \in \{0, 1\}$. Although regarding each specific value for a and b separately reduces this number to three, it requires to recompute the values $N_{a,s}, N_{b,s}$ and N_s for different a, b, where two of them will remain the same in each step. By computing them directly for all a, b and regarding the s values as the outer iteration and a, b as the inner iteration, we hence avoid recomputations that eventually render the negligibly smaller number of qubits inefficient. All of these five values were originally bounded by N and are hence guaranteed to be representable by k qubits.

Another hurdle are the divisions that are necessary to compute the χ^2 term. Performing these divisions too early and continuing computations on these rounding errors can aggravate the overall accuracy. To ensure as much numerical stability as possible, it is not quite reasonable to directly compute $Expected = \frac{N_{a,s} \cdot N_{b,s}}{N_s}$. Instead, equivalence transformations reveal that

$$
\begin{aligned}
\chi^2 &:= 2 \cdot \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} \sum_{b=0}^{1} \frac{(Expected - Observed)^2}{Expected} \\
&= 2 \cdot \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} \sum_{b=0}^{1} \frac{\left(\frac{N_{a,s} \cdot N_{b,s}}{N_s} - N_{a,b,s}\right)^2}{\frac{N_{a,s} \cdot N_{b,s}}{N_s}} \\
&= 2 \cdot \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} \sum_{b=0}^{1} \frac{N_{a,s} \cdot N_{b,s}}{N_s} - 2N_{a,b,s} + \frac{N_s N_{a,b,s}^2}{N_{a,s} \cdot N_{b,s}} \qquad | \sum_{b=0}^{1} \frac{N_{a,s} \cdot N_{b,s}}{N_s} = N_{a,s}, \sum_{b=0}^{1} N_{a,b,s} = N_{a,s} \\
&= 2 \cdot \left(\sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} -N_{a,s} + \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} \sum_{b=0}^{1} \frac{N_s N_{a,b,s}^2}{N_{a,s} \cdot N_{b,s}} \right) \qquad | \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} -N_{a,s} = -N \\
&= -2N + 2 \cdot \sum_{s \in \{0,1\}^{|S|}} \sum_{a=0}^{1} \sum_{b=0}^{1} \frac{N_s N_{a,b,s}^2}{N_{a,s} \cdot N_{b,s}}.
\end{aligned}
$$

This means that we can simply adjust the rejection threshold by the constant summand $2N$ and the constant factor 2 – independent of input S – and only have the computation of $\frac{N_s N_{a,b,s}^2}{N_{a,s} \cdot N_{b,s}}$ left, which avoids both treatment of negative numbers with the two's complement and unnecessary propagation of early division rounding errors. To ensure the most stable result, we first make the term as large as possible and compute the product $N_s N_{a,b,s}^2$, which requires $3k$ qubits. The division is then performed with rounding errors. To further avoid complications with undefined terms when $N_{a,b,s} = 0$, the original absolute frequencies could all be increased by $\left\lceil \frac{1}{2} \cdot \frac{N}{2^{k-1}-1} \right\rceil$ before converted to $[0, 2^{k-1} - 1]$.

E.1.3 Complexity analysis

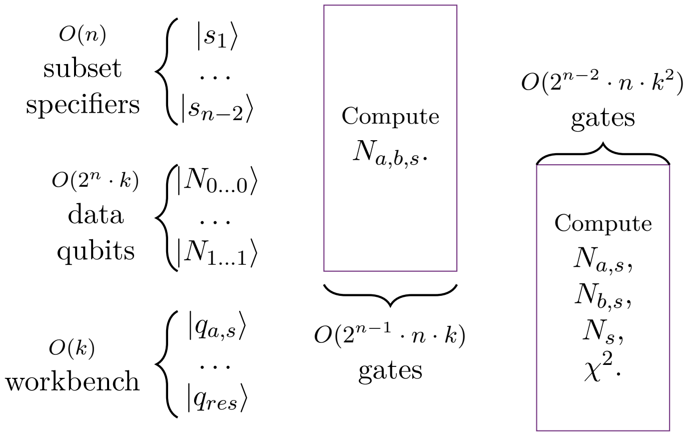


Figure 11: Schematic overview of the unitary circuit along with the required number of qubits and gates in each phase.

E.2 Score-based Quantum Annealing

In the following, we will clearly define each term of the original Hamiltonian $H(\mathbf{d}, \mathbf{y}, \mathbf{r}) = H_{\text{score}}(\mathbf{d}) + H_{\max}(\mathbf{d}, \mathbf{y}) + H_{\text{trans}}(\mathbf{r}) + H_{\text{consist}}(\mathbf{d}, \mathbf{r})$ that was proposed in (3).

E.2.1 Score Hamiltonian

Besides assuming its decomposability, we regard the specific objective function used in H_{score} as a functional black box. Recall a decomposable score function is a sum of local scores each attributable

CI Test: Classical, non-uniform.

```
Test: For any $S \subseteq V \setminus \{A, B\}, q := |S|$ . $(O(2^{n-2}))$
for all $s = (s_1, \dots, s_q) \in \{0,1\}^q$ $(O(\binom{n-2}{q}))$ do
    Compute $N_{a,b,s}$ for $a, b \in \{0,1\}$. $(O(2^{n-q}))$
    Compute $N_{a,s}, N_{b,s}, N_s$ for $a, b \in \{0,1\}$. $(O(1))$
    Evaluate the CI term. $(O(1))$
end for
Accept/Refute based on p-value.
```

CI Test: Quantum, **uniform** unitary.

```
Input: $q_1, \dots, q_{n-2},\ s_i \in S \Leftrightarrow q_i = 1$.
for all $s = (s_1, \dots, s_q) \in \{0,1\}^q$ do
    Compute $N_{a,b,s}$ for $a, b \in \{0,1\}$. $(O(2^n \cdot n^2))$
    Compute $N_{a,s}, N_{b,s}, N_s$ for $a, b \in \{0,1\}$. $(O(2^n \cdot n^3))$
    Evaluate the CI term. $(O(2^n \cdot n^2))$
end for
Accept/Refute based on p-value.
```

Figure 12: Pseudocode and complexity comparison of the classical non-uniform conditional independence (CI) test with our quantum gate-based unitary. While the classical CI test has a complexity of $O(n^2 \cdot 3^n) \approx O(n^2 \cdot 2^{1.585n})$, our unitary comprises only $O(2^n \cdot n^3)$ gates. The repeated evaluation of this unitary with Grover search scales this factor by another $\sqrt{2^n}$, culminating in a complexity of $O(2^{1.5n} \cdot n^3)$ for the quantum algorithm. The factor n^2 in the classical complexity originates from the transfer of arithmetic operation complexity to gate complexity.

to one specific node. The score Hamiltonian hence takes the form

$$H_{\text{score}}(\mathbf{d}) = \sum_{i=1}^{n} H_{\text{score}}^{(i)}(\mathbf{d}) = \sum_{i=1}^{n} H_{\text{score}}^{(i)}(\Gamma_i),$$

where Γ_i represents the parent set of node i. The specific objective function can either be one information-theoretic score as the AIC introduced in the last section or a Bayesian score, which the authors eventually chose.

E.2.2 Transitive order

To avoid cycles in the graph, we first have to ensure that the variables $\mathbf{r}$ identify a transitive order on the notes. Although the following notation might look overwhelming, it basically verifies the definition of transitivity for each triple of nodes. That is, the overall term consists out of $H_{\text{trans}}(\mathbf{r}) = \sum_{1 \leq i \neq j \neq k \leq n} H_{\text{trans}}^{(ijk)}(r_{ij}, r_{ik}, r_{jk})$. For each triple, we first check whether $i < j$ and $j < k$ implies $i < k$. This corresponds to the term $r_{ij} r_{jk}(1 - r_{ik})$, which equates to 1 is this constraint is violated. Conversely, we check whether $k < j$ and $j < i$ implies $k < i$, which leads to the symmetric second term $(1 - r_{ij})(1 - r_{jk}) r_{ik}$. Violations of this constraint are scaled by the penalty weight $\delta_{\text{trans}}^{(ijk)}$, whose necessary magnitude we will discuss later. Since the cubic terms $r_{ij} r_{jk} r_{ik}$ cancel with each other, the overall term remains 2-local.

$$\begin{aligned}
H_{\text{trans}}^{(ijk)}(r_{ij}, r_{ik}, r_{jk}) &= \delta_{\text{trans}}^{(ijk)} \left[r_{ij} r_{jk}(1 - r_{ik}) + (1 - r_{ij})(1 - r_{jk}) r_{ik} \right] \\
&= \delta_{\text{trans}}^{(ijk)} \left(r_{ik} + r_{ij} r_{jk} - r_{ij} r_{ik} - r_{jk} r_{ik} \right) \\
&= \begin{cases} \delta_{\text{trans}}^{(ijk)}, & \text{if } (x_i \leq x_j \leq x_k \leq x_i) \vee (x_i \geq x_j \geq x_k \geq x_i), \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

E.2.3 Consistent order

So far, the topological order that $\mathbf{r}$ is supposed to encode does not have to coincide with the actual adjacencies implied by $\mathbf{d}$. The following consistency term takes care of that. It suffices to compare the direction of edges locally with the topological order, because an inconsistency in the order necessarily violates such an edge direction somewhere along the erroneous order. We merely need to check that an edge from i to j implies $i < j$ in the topological order for each pair of nodes $i, j \in V$. The overall term hence amounts to

$$H_{\text{consist}}(\mathbf{d}, \mathbf{r}) = \sum_{1 \leq i < j \leq n} H_{\text{consist}}^{(ij)}(d_{ij}, d_{ji}, r_{ij}) = \sum_{1 \leq i < j \leq n} \delta_{\text{consist}}^{(ij)}(d_{ji} r_{ij} + d_{ij}(1 - r_{ij})),$$

where $\delta_{\text{consist}}^{(ij)}$ again identifies a penalty weight to scale consistency violations.

E.2.4 Maximum parents

Finally, the authors restrict their search to DAGs where each node has at most m parents. As discussed in the prior section on classical algorithms, this notably reduces the overall computational cost. On the contrary, the authors require additional slack variables $\mathbf{y}$ to encode this constraint. Each slack variable set $0 \leq \mathbf{y}_i \leq m$ encodes a non-negative integer and hence requires $n\lceil \log(m+1) \rceil$ additional variables. To represent the integer that $\mathbf{y}_i$ encodes, we define $y_i := \sum_{l=1}^{\lfloor \log_2(m+1) \rfloor} 2^{l-1} y_{il}$. To further simplify the notation in the following, define as $d_i := \sum_{\substack{1 \leq j \leq n \\ j \neq i}} d_{ji}$ the number of parents of node i.

If this number of parents is smaller than m, then our freedom on y_i allows us to close the gap between these two integers. However, if $d_i > m$, then y_i cannot revert this overshoot because it can only represent non-negative values.

For each node i in the graph, we therefore formulate the term

$$H_{\max}^{(i)}(d_i, y_i) = \delta_{\max}^{(i)}(m - d_i - y_i)^2,$$

such that for the optimal choice of y_i, $\min_{y_i} H_{\max}^{(i)}(d_i, y_i) = \begin{cases} 0, & d_i \leq m, \\ \delta_{\max}(d_i - m)^2, & d_i > m. \end{cases}$

E.2.5 Penalty weights

To ensure that the optimal solution indeed encodes the most likely DAG, the authors had to ensure the penalty weights scale constraint violations large enough. Graphs that do not abide by the constraints of DAGs can achieve an unfair advantage on minimizing the objective function, and hence needs adequate punishment. To that end, it is sufficient to ensure that each traversal step that introduces new constraint violations is punished more by the three penalty terms $H_{\max}(\mathbf{d}, \mathbf{y}), H_{\text{trans}}(\mathbf{r})$, and $H_{\text{consist}}(\mathbf{d}, \mathbf{r})$ than rewarded by the score term $H_{\text{score}}(\mathbf{d})$.

To obtain a worst-case estimate of the advantage that adding an edge from j to i could effect on the minimization of the objective function, we have to consider that all other incoming edges from other nodes $k \neq i, j$ are optimized to achieve the best advantage in the score. For any edge from j to i, we hence denote this worst-case advantage in the minimization of the objective function by

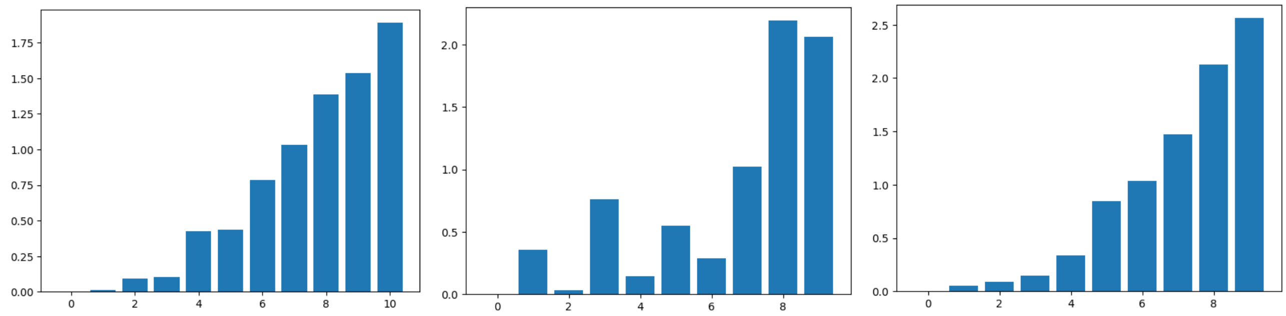
$$\Delta_{ji} = - \min_{\{d_{ki} | k \neq i, j\}} \{ H_{score}^{(i)}|_{d_{ji}=1} - H_{score}^{(i)}|_{d_{ji}=0} \}.$$

Having obtained the advantage of adding an edge from j to i for every j, i, we need to account for the intuition that the optimization search will choose to add the edge that yield the *best* advantage. For that reason, we finally define $\Delta_i := \max_{\substack{1 \leq j \leq n \\ j \neq i}} \Delta_{ji}$ as the best advantage that adding an edge from any other node to i could possibly entail and $\Delta := \max_i \Delta_i$ as the best advantage adding *any* edge to the graph could possibly achieve during the entire optimization procedure.

Then, we can directly define the lower bound for the transitive penalty weights as $\delta_{\text{trans}}^{i,j,k} > \Delta$ and the maximum parent penalty weight as $\delta_{\max}^{(i)} := \Delta_i$.

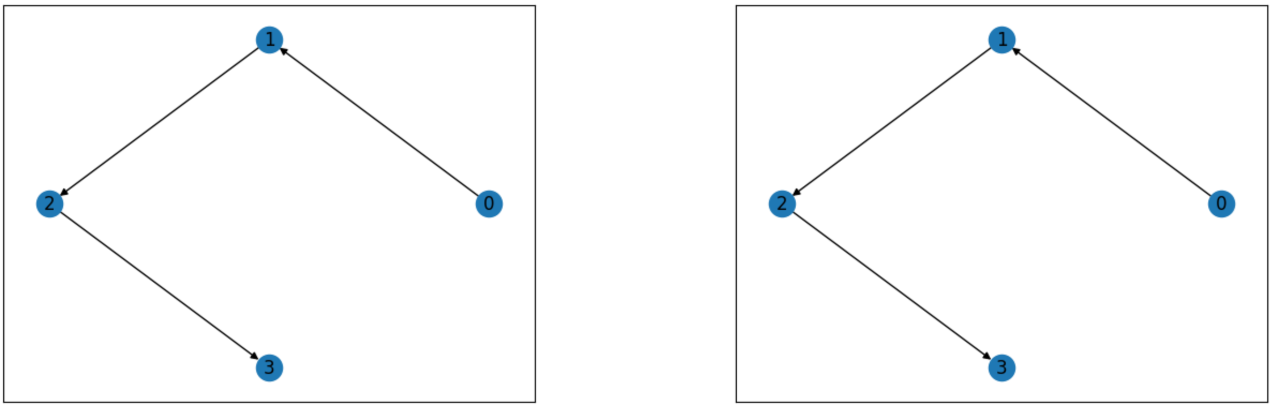
Although this is surely not an optimal lower bound and lower penalties might still satisfy the optimality condition, this formulation has the nice property that it regards the score terms $H_{score}^{(i)}$ as functional black boxes. This will later allow us to directly apply their lower bound on our alternative formulation of the score Hamiltonian. Moreover, we have theoretically not accounted for the possibility that the lowest energy of a penalized state is smaller than the energy of the first excited unpenalized state, which could further enlarge the required computation time to meet the conditions of the adiabatic theorem (3).

F Further Experiment results



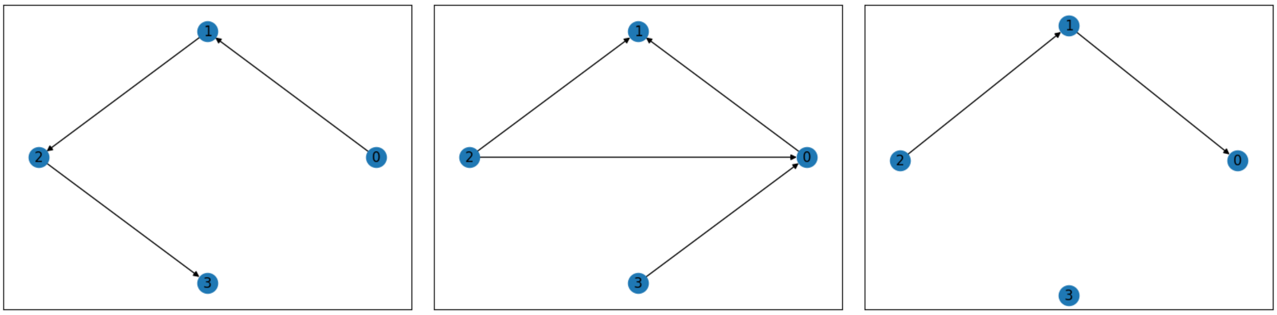
(a) 1-local $\hat{H}_{\text{score}}$ (b) Original H_{score} (c) 2-local $\hat{H}_{\text{score}}$

Figure 13: The optimal objective function score among the set of parent sets J grouped in increasing order by their DAG distance to the true parent set J^*. The minimum energies for the original term are quite irregular across several distances (Fig. 13b), which is due to the irrelevance of certain edges to their causal equivalence classes.



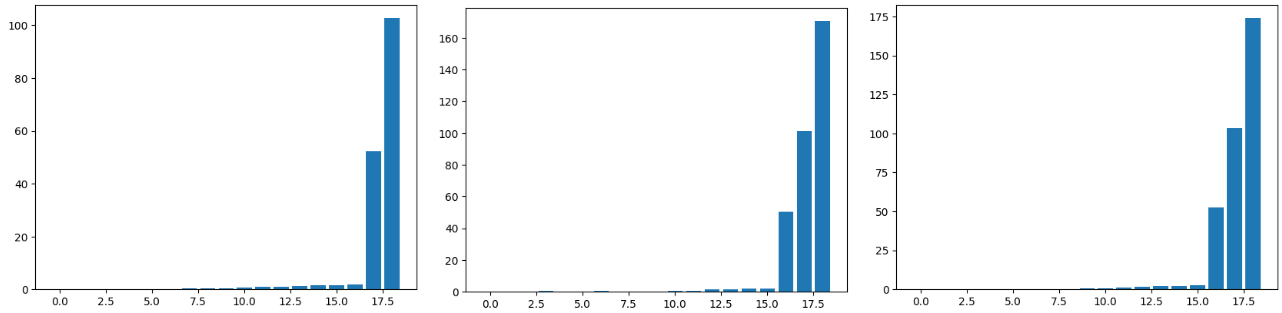
(a) Best result (b) Second-best result

Figure 14: The ground states returned by our QA algorithm without λ-regularization on a real-world problem instance. According to medical research, no correlations exist between diabetes mellitus, cirrhosis, and immunosuppression. This result also achieves the lowest energy in our Hamiltonian and is measured with noticeable probability, while the result with the second-lowest energy in 14b is obtained even with higher probability.



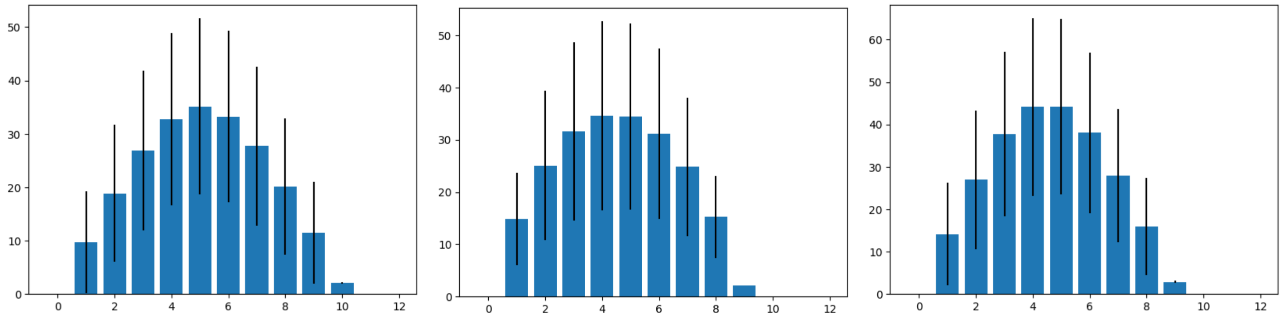
(a) Ground truth (b) QA result with $\lambda = 0$ (c) QA result with $\lambda = 1$

Figure 15: The ground states returned by our QA algorithm, when we employed λ-regularization instead of the $H_{\max}$ term.



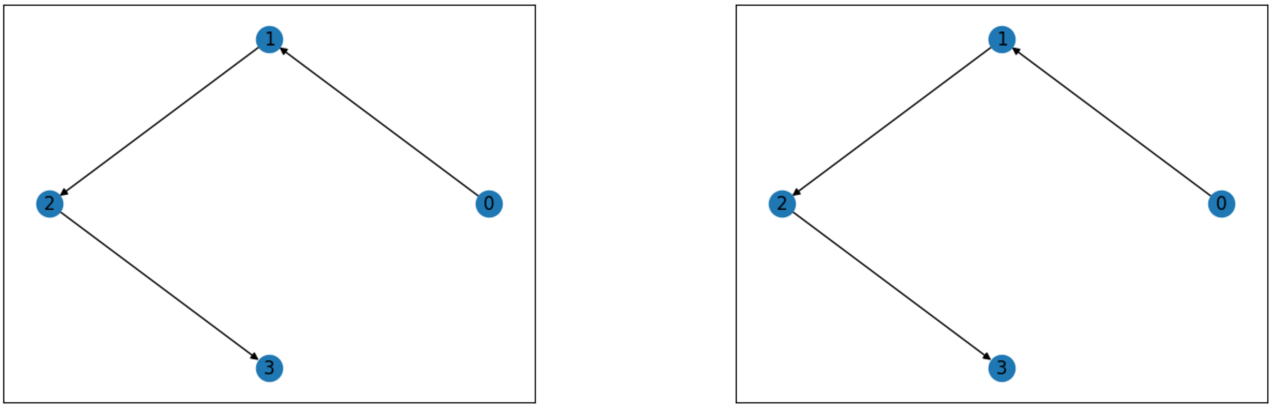
(a) 1-local $\hat{H}_{\text{score}}$ (b) Original H_{score} (c) 2-local $\hat{H}_{\text{score}}$

Figure 16: The optimal objective function score among the set of parent sets J grouped in increasing order by their spin distance to the true parent set J^*.



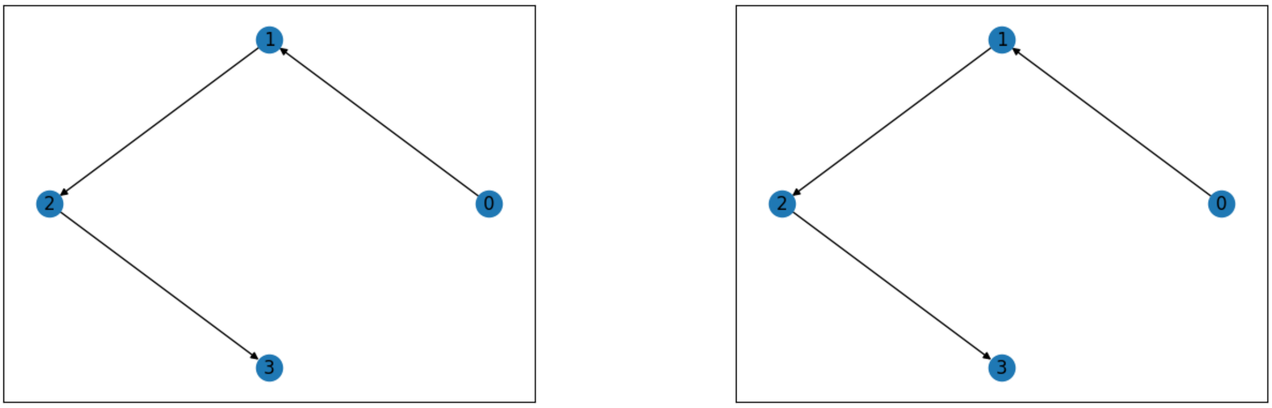
(a) 1-local $\hat{H}_{\text{score}}$ (b) Original H_{score} (c) 2-local $\hat{H}_{\text{score}}$

Figure 17: Average scores for parent sets J grouped in increasing order by their DAG distance to the true parent set J^*. The black bars indicate the standard deviation of the scores in each set.



(a) Best result (b) Second-best result

Figure 18: The ground states returned by our QA algorithm with λ-regularization on a real-world problem instance. To convert the Body Mass Index (BMI) to a binary variable, we chose a conservative threshold at 28. Similarly, the age binary variable was set to 1 when the age exceeded 60.



(a) $\lambda = 0$

(b) $\lambda = 50$

Figure 19: The optimal objective function score among the set of parent sets J grouped in increasing order by their DAG distance to the true parent set J^* when employing different regularization penalty weights $\lambda = 0, 50$. As a result, the excited states are largely scaled for some of the adjacencies they contain might be superfluous to fit the data.