Opportunities for Quantum Acceleration of Databases: Optimization of Queries and Transaction Schedules

Umut Çalıkyılmaz
University of Lübeck
calikyilmaz@ifis.uni-luebeck.de

Sven Groppe
University of Lübeck
groppe@ifis.uni-luebeck.de

Jinghua Groppe
University of Lübeck
groppej@ifis.uni-luebeck.de

Tobias Winker
University of Lübeck
winker@ifis.uni-luebeck.de

Stefan Prestel
Quantum Brilliance GmbH
stefan.prestel@quantum-brilliance.com

Farida Shagieva
Quantum Brilliance GmbH
farida.s@quantum-brilliance.com

Daanish Arya
Quantum Brilliance GmbH
daanish.a@quantum-brilliance.com

Florian Preis
Quantum Brilliance GmbH
florian.p@quantum-brilliance.com

Le Gruenwald
The University of Oklahoma
ggruenwald@ou.edu

ABSTRACT

The capabilities of quantum computers, such as the number of supported qubits and maximum circuit depth, have grown exponentially in recent years. Commercially relevant applications that take advantage of quantum computing are expected to be available soon. In this paper, we shed light on the possibilities of accelerating database tasks using quantum computing with examples of optimizing queries and transaction schedules and present some open challenges for future studies in the field.

PVLDB Reference Format:
Umut Çalıkyılmaz, Sven Groppe, Jinghua Groppe, Tobias Winker, Stefan Prestel, Farida Shagieva, Daanish Arya, Florian Preis, and Le Gruenwald. Opportunities for Quantum Acceleration of Databases: Optimization of Queries and Transaction Schedules. PVLDB, 16(9): 2344 - 2353, 2023. doi:10.14778/3598581.3598603

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/TobiasWinker/QC4DB_VQC_Tutorial.

1 INTRODUCTION

A decade after Richard Feynman came up with the idea of quantum computing [44], the first quantum algorithms that provide a speedup for problems with practical use were developed [53, 126]. This speedup is obtained by exploiting the quantum nature of particles [102]. In this study, we analyze the current state of the art in quantum computing that can be used to accelerate database tasks.

The performance of a database management system (DBMS) is crucial, especially for large-scale data-driven applications. In the last decade, the importance of fast data storage and retrieval has increased with the emergence of the notion of big data [89]. At the same time, the first quantum computers were developed and more studies have been focused on quantum computing [103]. Some of these are on improving database performance by applying quantum annealing for query optimization [101, 124], multi-query optimization [133] and transaction scheduling [15]. The studies in [124, 133] have presented improvements in runtime up to 10^3 times. The third study [15] has shown that the runtime of quantum annealing stays constant for increasing problem size while that of classical simulated annealing rises quickly. Additionally, it is expected that the capacity of quantum computers will increase rapidly in the future years and that on-site quantum computers, which would provide lower latency than cloud-based ones, will be available in a few years (see Section 2). With all these promising results, there is an obvious need to examine possible quantum speed-ups for solving database problems. In this paper, we aim to provide guidance for such studies by showing how various quantum approaches scale by parameters of query optimization and transaction scheduling problems, and by presenting the open challenges for developing these approaches and integrating them into database systems.

The rest of the paper is organized as follows. Section 2 focuses on the current state of quantum computing technology and its estimated future timeline. Section 3 introduces the two DBMS problems of interest in detail. In Section 4 we present various quantum approaches that can be used to accelerate the database problems and their qubit and circuit depth requirements. Section 5 proposes new directions for future research by discussing open challenges. Finally, Section 6 concludes the paper by summarizing our findings.

2 EMERGENT GOLDEN AGE OF QUANTUM COMPUTING

2.1 Types of Quantum Computers

More than 20 years ago, DiVincenzo established five criteria to realize a scalable quantum computer [38]. Nowadays, one can single out a lot of different platforms on which it can be implemented: superconducting qubits [36, 57], trapped ions [27], photons [105, 106], color centers in solids [2, 20, 25, 37, 111, 143], semiconductor quantum dots [33, 83, 110, 138, 152], Rydberg atoms [28], topologically protected systems [72, 76, 122], neutral atoms [7, 8, 50] and others [30, 31, 47]. These implementations range from recently published

Proceedings of the VLDB Endowment, Vol. 16, No. 9 ISSN 2150-8097.
doi:10.14778/3598581.3598603

proposals and articles about qubit fabrication to commercially available devices. Quantum computing models can be categorized into two classes, universal and non-universal, depending on whether the model can efficiently simulate a quantum Turing machine [34].

The most prominent type of computation is the gate-based or circuit model, where the algorithm is performed through the sequential application of quantum logic gates assembled in a quantum circuit. If a quantum device allows the implementation of a universal basis of unitary operators as quantum gates, it realizes a universal quantum computer. The gate-based model is closest to classical computation in terms of the logic of calculations. The high levels of noise in present-day quantum devices have led to a focus on operating in the noisy intermediate-scale quantum (NISQ) computing era [115]. Actively pursuing protection from noise [68, 72, 112] opens the path to error-free universal quantum computing.

Non-universal computing models aim at finding solutions only for certain problem classes. Quantum annealing (QA), for example, specializes in solving optimization problems formulated as quadratic unconstrained binary optimization (QUBO) problems. QA machines are supporting today 5000+ qubit counts [70, 116].

2.2 Quantum Computer Timeline and Roadmap

Quantum algorithms can exponentially speed up computation time for a number of tasks [53, 58, 126]. However, to compete with supercomputers to solve commercially interesting problems [18, 130] in the fields of cryptography [48, 54], database optimization or quantum chemistry [74, 128], thousands of error-corrected qubits are required. Current devices accept that logical operations are performed with errors, and have moderate qubit counts of tens to a few hundred, motivating their classification as NISQ computers [115]. 50 qubits are believed to be a break-even point when quantum computers surpass the efficiency of brute force evaluations on classical supercomputers [115]. In fact, current NISQ-era devices can outperform supercomputers for some very specific calculations [10, 85], prompting the majority of hardware companies to gradually scale up qubit counts until error-correction is achieved.

These advances are due to hybrid quantum-classical algorithms. The main insight is to divide the problem into the classical and quantum parts and offload classically hard tasks onto quantum computers. Current workflows for executing hybrid computations are based on the CPU-based client interacting with a quantum processing unit (QPU) based mainframe over a network [93]. The large footprint and special operating conditions, such as cryogenic temperatures, usually mandate cloud-based approaches. The diamond quantum computer developed by Quantum Brilliance is able to operate at room temperature [39]. This enables the designing of compact quantum accelerators, which can be easily integrated with a classical computer. With on-site quantum computing, hybrid algorithms can be implemented for data-intensive applications, such as databases, without the overhead of data transfer.

2.3 Quantum Computing Libraries

There are several publicly available quantum libraries that allow for exploring the power of quantum computing, either through simulation or interfaces to real quantum hardware [46]. Many of these libraries can be used to define quantum algorithms, all the way from abstract circuits as functions, to sending pulse-level instructions that interface with the control electronics. This usually means they are shipped with pre-defined features such as a circuit library, circuit transpilation for native gate sets of different QCs, and further methods to obtain and make use of quantum measurements. Many of these libraries support simulating noise models to let users develop realistic applications for NISQ devices. The performance of a library typically depends on its specialization [97, 121].

There are several quantum computing libraries integrating circuit building, simulation, noise profiling, and access to quantum hardware. **Qiskit** by IBM [9] is the most widely-used one. It allows access to hardware from IBM, AQT, and IonQ. **Cirq** by Google [35] allows access to Google's Sycamore [10] besides AQT, IonQ, Pasqal and Rigetti. Its "qubit picking" service allows for algorithm-aware hardware selection. **Pennylane** by Xanadu [12] focuses on differentiable quantum programming [149]. It supports external backends or hardware via plugins with several common full-stack libraries. **Qristal** by Quantum Brilliance focuses on NISQ computing. Noisy simulations reflecting the hardware limitations of the diamond quantum computer and distributed calculations over multiple quantum accelerators are embedded.

3 TWO EXAMPLES OF OPTIMIZATION PROBLEMS IN DATABASES

3.1 Query Optimization

The join order in a query execution plan is crucial for the query execution time. Hence, one of the most important tasks of query optimization is to determine the join order with the best-estimated costs. The number of possible join orders for a query with n tables is given by the formula $\frac{(2(n-1))!}{(n-1)!}$ [117], and is reduced to $\frac{(2(n-1))!}{2^{n-1}(n-1)!}$, if we ignore the order of the tables in a single join.

Hence checking all possibilities using e.g. dynamic programming [125] is suitable only for a low number of tables. For a higher number of tables, we have to use heuristic approaches like ant colony optimization [132], machine learning [88, 150] or genetic algorithms [63], which can find a good solution without checking every possibility. Section 4 deals with quantum approaches for join order optimization including estimating the query costs [55].

3.2 Transaction Scheduling

When multiple transactions are processed concurrently in a database, ACID properties must be fulfilled to ensure the validity of data [118]. There are various policies to guarantee the isolation property by dealing with conflicts between transactions, each having some type of overhead costs [22, 56, 131]. Reordering the transactions in the queue to avoid conflicting ones running at the same time decreases these costs and raises throughput [84].

When the transactions are assumed to be atomic blocks, the transaction scheduling problem becomes scheduling t one-stage jobs (transactions) on c identical machines (cores) with additional conflict constraints. The goal is to find a schedule that minimizes the makespan (i.e., the maximum execution time of the cores). Because of the conflicts, the problem is sequencing the transactions besides assigning them to the cores. With this property, transaction scheduling resembles the famous job-shop scheduling problem [87].

In [51], job-shop schedules are encoded as permutations. Transaction schedules can be represented similarly. In this enumeration, any schedule of t transactions can be represented by a permutation of the elements of $\{1, 2, ..., t\}$. A schedule can be formed from a permutation by inserting the transactions one by one into the schedule, and by selecting the core with the minimum processing time and minimum order resulting in $t!$ possible schedules of t transactions.

4 OPPORTUNITIES FOR QUANTUM ACCELERATION OF DATABASES

In this section, we present quantum approaches that can be used to solve database optimization problems. The main limitations of NISQ devices are their low qubit counts and their vulnerability to noise [115], which affects the applicability of the quantum methods.

The required qubits for a quantum algorithm can be divided into two groups: the representation qubits used to represent possible solutions to a problem and the ancilla qubits used for additional calculations. An ancilla qubit is set to its initial state after it is utilized. These qubits do not store any information about the input or the output, but are used to keep intermediate results and are mostly needed when a function is evaluated on a quantum circuit. The circuit depth of an algorithm is the number of quantum gates needed to be applied serially. A large circuit depth causes more noise, which would result in inconsistent results on NISQ devices.

We provide a summary of the requirements of the discussed quantum methods in Table 1. The first four approaches given in the table use ancilla qubits but the number of these is not given. The reason is that the implementation of these methods requires quantum black boxes (called *Oracle*) that evaluate the objective function of a problem coherently. The design of such a black box is problem and encoding-specific. For the same reason, the circuit depths of these Oracles are not given explicitly, but they are known to have polynomial depths since both problems are in NP [77, 136].

4.1 Exact Algorithms

The exact algorithms given below can find the exact optimal solution but require a big amount of resources to do so.

Enumeration of all Possibilities: If no information is given about a function f with an enumerable domain D, then in order to find $x \in D$ with $f(x) = y$ for a given y (also called *the search problem* [53]), all $x \in D$ must be tried one by one until a solution is found on classical hardware. For an optimization problem, all $x \in D$ must be tried out to find $argmin_x f(x)$. For join order optimization of n tables and scheduling t transactions, this method needs $O\left(\frac{(2(n-1))!}{(n-1)!}\right)$ and $O(t!)$ steps respectively.

Grover's algorithm finds a solution to a search problem with N items and M solutions in $O(\sqrt{N/M})$ steps, achieving a quadratic speed-up compared to classical search [53]. The original algorithm requires the number of solutions as an input, but a variant of it (exponential searching) can solve a search problem with an unknown number of solutions in $O(\sqrt{N/M})$ steps [19].

Another variant of Grover's algorithm (called Dürr-Høyer) adapts quantum search to find the exact solution to optimization problems in $O(\sqrt{N})$ steps [41]. It has been applied to transaction scheduling in [52]. The number of qubits required to represent the items is the logarithm of the cardinality of the search space. In this approach, the objective function is evaluated for a coherent quantum state, so ancilla qubits are required, and the circuit depth is the number of Oracle calls times the circuit depth of the quantum black box.

Linear Programming: Linear programming (LP) is the name of the efforts of optimizing a linear objective function over a continuous solution space limited by linear constraints. The best-known and most-studied algorithm for this task is the simplex algorithm [100]. There are variants of LP used for different types of problems. Mixed integer linear programming (MILP) is such a variant for linear problems with discrete variables amongst continuous ones [140]. Problems that are formulated as MILP models can be solved by branch and bound methods, where the bounds are set by solving the LP relaxation of the model by the simplex algorithm [65].

The simplex algorithm can be sped up using quantum subroutines without the need for a QRAM [98] using the algorithm given in [26] to solve linear systems of equations in each iteration. In this way, the complexity of an iteration is reduced to $\tilde{O}(\frac{1}{\epsilon}\kappa d\sqrt{\alpha}(d_c\alpha + d\beta))$ where ϵ is the targeted precision, d_c and d_r are the maximum numbers of elements in the columns and rows, $d = \max\{d_c, d_r\}$, α and β are the numbers of variables and constraints, and $\tilde{O}$ means that the polylogarithmic terms are hidden. The number of qubits required to represent the matrices is $O(\log\alpha + \log\kappa)$. Also, additional qubits are required. The values of d, d_c, α and β depend on the MILP formulation, but κ depends on the specific problem instance.

In the MILP formulation for the join order problem [135] with $O(n^2)$ variables and $O(n^2)$ constraints, $d_c \in O(1)$ and $d \in O(n)$, such that the number of qubits is $O(\log n + \log\kappa)$ and the circuit depth required for the quantum simplex method is $\tilde{O}(\frac{1}{\epsilon}\kappa n^5)$.

We use the MILP formulation for job-shop scheduling [73] to analyze the requirements for transaction scheduling. The main difference between the two problems is the conflict constraints, which do not change the dominating terms. In the given formulation, the number of variables and constraints are both $O(t^2c)$, and the sparsity values d and d_c are both $O(1)$. Then the number of qubits becomes $O(\log(t^2c) + \log\kappa)$ and the circuit depth $\tilde{O}(\frac{1}{\epsilon}\kappa t^3c\sqrt{c})$.

Dynamic Programming: Dynamic programming (DP) uses the partial optimality condition of a problem to avoid enumerating all possibilities [42]. It divides a problem into sub-problems and solves them recursively. It has been shown that DP can reduce the complexity of sequencing problems from $O(N!)$ to $O(2^N)$ by searching over the combinations instead of permutations [11, 60].

An optimality condition is well-known for query optimization since DP is used to solve it for decades [125]. There is no DP approach developed specifically for the transaction scheduling problem yet, but DP is applied for other job scheduling problems [51], which could be adapted for the transaction scheduling problem.

There are studies showing that quantum subroutines might be used to improve DP [120]. One such study shows that the complexity of a DP approach for the traveling salesperson problem with N cities could be decreased to $O^*(1.728^N)$ from $O^*(2^N)$ [6]. For that, the costs of sub-paths of a specified length are classically calculated. Then, the optimal merge of those sub-paths is found using the Dürr-Høyer algorithm given in [41]. A similar approach can be developed for the problems that are examined in this paper.

Table 1: Quantum Approaches for Database Optimizations

Approach	Solution Type	Representation Qubits: Join Order Optimization	Representation Qubits: Trans. Scheduling	Ancilla Qubits	Circuit Depth: Join Order Optimization	Circuit Depth: Trans. Scheduling	Simulation Advantage
Enumeration of all Possibilities	Exact	$O\left(\log\left(\frac{(2(n-1))!}{(n-1)!}\right)\right)$	$O(\log(t!))$	Yes	$O\left(P(n)\sqrt{\frac{(2(n-1))!}{(n-1)!}}\right)$	$O\left(P(t)\sqrt{t!}\right)$	No
Linear Programming	Exact	$O(\log n + \log \kappa)$	$O(\log(t^2c) + \log \kappa)$	Yes	$\tilde{O}(\frac{1}{\epsilon}\kappa n^5)$	$\tilde{O}(\frac{1}{\epsilon}\kappa t^3 c\sqrt{c})$	No
Dynamic Programming	Exact	$O(n)$	$O(t)$	Yes	$O\left(P(n)2^{n/2}\right)$	$O\left(P(t)2^{t/2}\right)$	No
Nature Inspired Heuristics	Approx.	$O\left(\log\left(\frac{(2(n-1))!}{(n-1)!}\right)\right)$	$O(\log(t!))$	Yes	$O(P(n))$	$O(P(t))$	Yes
Quantum Annealing	Approx.	$O(n^2)$	$O(Rct)$	No	Not Applicable		No
Variational Algorithms	Approx.	$O(n^2)$	$O(Rct)$	No	$O(1)$	$O(1)$	No
Quantum Machine Learning	Approx.	$\log\left(\frac{(2(n-1))!}{(n-1)!}\right)$	$\log(t!)$	No	$O\left(\log\left(\frac{(2(n-1))!}{(n-1)!}\right)\right)$	$O(\log(t!))$	Yes

n: Number of Tables t: Number of Transactions c: Number of Cores R: Heuristic Timespan Value κ: Condition Number
$P(x)$: Polynomial of x $\tilde{O}$: Big O with Hidden Polylogarithmic Terms

The number of representation qubits for such an approach is linear with the problem parameters n and t because the sizes of the search spaces would be $O(2^n)$ and $O(2^t)$. To accelerate DP by quantum computing, the Dürr-Høyer algorithm is used, which requires a polynomially scalable Oracle and ancilla qubits. Also, depending on the method to be implemented, it might be necessary to load classically computed values on a QRAM or some equivalent system. Fortunately, there are circuit-based equivalents of a QRAM that can be implemented on a NISQ device [108].

4.2 Heuristics

Some heuristic algorithms, which are used to find near-optimal solutions using a small number of resources, are introduced below.

Nature Inspired Heuristics: Over the years, various heuristic algorithms that are inspired by some aspects of nature are developed. Genetic algorithm (GA) [61] mimics genetic evolution to increase the quality of the individuals in a population of solutions. Ant colony optimization (ACO) [29] simulates the foraging behavior of ants to find the shortest path on a graph through pheromone trails. Whale optimization algorithm (WOA) [96] models the spiraling and convergence of humpback whale hunting routines. Particle swarm optimization (PSO) [113] imitates the flight of a flock of birds to evolve the positions of randomly positioned particles.

Discrete variations of WOA [80] and PSO [69], which are applicable to sequencing problems, are proposed. There are implementations of GA for join order optimization [63] and job scheduling problems [32]. Also, ACO [64, 132] and PSO [81, 95] approaches are developed to solve these problems. Thus, all these algorithms can be utilized to solve database problems.

It is shown that genetic algorithms can be accelerated using quantum computing [86]. The proposed method applies the Dürr-Høyer algorithm [41] to select the most fitting elements in each generation. In this case, the exact optimality requirement is relaxed and $O(1)$ iterations are applied to find a subpopulation of individuals of maximal fitness, one of which is selected by measurement. This requires the use of QRAM. Dürr-Høyer algorithm can also be utilized to select an initial population without a need for QRAM, making the strategy much more viable on NISQ devices. Such a method would provide a better initial population, which would result in faster convergence. This approach can be applied to any heuristic that utilizes a random initial population. Thus, it is applicable for every algorithm given above, except for ACO.

In the last few decades, many studies have focused on implementing ideas of quantum computing into classical heuristics. Quantum-inspired versions of the algorithms we presented above have been developed over the years. Some of these are QGA [99, 141], QACO [82, 142], QWOA [3] and QPSO [129]. In these algorithms, some properties of the individuals of the population (or pheromone levels) acquire quantum behavior by being simulated as qubit strings or continuous quantum systems on classical computers. Even though they are just simulating quantum systems, experience suggests that they provide improvements over their conventional counterparts, showing the potential of genuine quantum algorithms.

Simulated and Quantum Annealing: Simulated annealing is a heuristic approach that is inspired by a metallurgical method of slowly cooling a solid to obtain a desirable material [71]. It is a probabilistic search algorithm, where the next point on the search space is selected by a stochastic process. First, the fitness of a randomly selected point is evaluated. If the fitness of this new point is better than the current one, the new point is selected. If it is worse, the new point is selected with some probability, which depends on the current temperature value. A higher temperature value means a higher probability to move to less fitting points. The temperature value is decreased at each step, finally reaching zero. If the temperature is decreased at a sufficiently slow rate, this process definitely converges to the global optimal and finds competitive near-optimal solutions in the time used by other heuristics for solving problems in practice [13].

Quantum annealing (QA) follows a similar logic, but it uses quantum tunneling caused by a negative potential instead of thermal fluctuations caused by temperature [45]. In this method, the intensity of the negative potential is decreased slowly to reduce the amount of fluctuation and obtain a stable quantum state at the end of the process. For the application of quantum annealing, the transverse Ising model is proposed [66]. Problems that are formulated as QUBO problems [49] can be solved using this model.

QUBO approaches for both join order optimization [101, 123, 124, 134, 137] and transaction scheduling [15, 16] were developed, which allow the usage of quantum annealing to solve them. In QUBO formulations of join order optimization and transaction scheduling, the numbers of binary variables are $O(n^2)$ and $O(Rct)$ respectively, which give us the numbers of representation qubits.

Variational Quantum Algorithms: For large problem instances, implementing quantum algorithms that require high circuit depths or many qubits is not possible on NISQ computers [115]. This situation led much research to focus on hybrid algorithms, where some part of the computation is done by classical computers. Variational quantum algorithms (VQA) [23] are hybrid algorithms where parameterized quantum circuits, known as variational quantum circuits (VQC), are run and measured repeatedly by changing the values of the parameters. The aim is to find the optimum values of the parameters and record the result obtained by using those values. Optimization is done using classical computers.

QAOA [43], VQE [109], VarQITE [92], and FVQE[4] all try to find the ground state of a given Hamiltonian. The main differences between them are the type of quantum circuits they employ and the method they use to evolve the parameters. These make each approach to be efficient for a different set of problems. It is shown that these can be used to find approximate solutions to sequencing problems [5], by formulating them as QUBO problems and the objective functions as Hamiltonians. The contributions in [101, 124] evaluate join ordering using QAOA and [101] additionally VQE, but there is no study to solve transaction scheduling using VQA. However, because transaction scheduling can be formulated as QUBO problem [15, 16], such approaches can be developed.

The numbers of representation qubits are the same as the ones for QA. The objective function is computed by a classical computer avoiding ancilla qubits. In these methods, the circuit depth is a constant, which might be tuned to obtain a better result.

There are other uses of variational algorithms. The variational quantum linear solver [21] is a hybrid algorithm using a VQC to approximately solve a linear equation for accelerating the linear programming method. VQCs can also be used for machine learning.

(Quantum) Machine Learning: Machine learning enables the prediction of the solution based on past problems by using a model. The model contains parameters to be adjusted based on the given data to improve the predictions during a learning phase. Different types of models include linear regression [91], support vector machines [104], decision trees [67], and neural networks [14].

Common approaches to quantum machine learning (QML) are hybrid algorithms with a VQC consisting of controlled gates and rotation gates. The angles of the rotation gates are adjusted by a classical optimizer like Adam optimizer [127] or by a genetic algorithm [24]. The structure of the circuit itself can be optimized [119, 151]. These algorithms are inspired by neural networks and replace a classical neural network with a quantum circuit. The advantage of VQCs is the ability to achieve the same results using fewer parameters [40] with benefits even for only simulated VQCs.

```python
# create a quantum circuit with n qubits
qc = qiskit.QuantumCircuit(n)
# Parameters for input
x = qiskit.circuit.ParameterVector('x', n)
# angle encoding
for i in range(n):
    qc.rx(x[i], i)
# add alternating rotation and entanglement
qc.compose(TwoLocal(n,['ry','rz'],'cx','linear'),3)
# connect to pytorch
model = TorchConnector(CircuitQNN(qc, ...))
# optimizer for training the model
optimizer = Adam(model.parameters())
# train the model
for episode in range(2000):
    prediction = model(Tensor(features))
    loss = ...  # calculate some loss function
    loss.backward()
    optimizer.step()
# use model to make a prediction based on features
prediction = model(Tensor(features))
```

Figure 1: Code excerpt for creating, training and using a VQC with angle encoding as a machine learning model using Qiskit and PyTorch.

An important step of a quantum algorithm is the encoding of classical data into a quantum state as it affects what data can be encoded, which operations can be executed, and how many qubits are required. Three common encodings used in QML are basis encoding (storing each bit of classical data in a qubit), angle encoding (storing one real value in the amplitudes of a qubit), and amplitude encoding (storing real values in the amplitudes of the combined quantum states of multiple qubits) [144]. Quantum associative memory (QuAM) is a form of basis encoding that encodes a set of bit strings instead of a single bit string as the superposition of the basis encodings of all strings in the set. Basis and angle encodings have a linear qubit complexity of $O(n)$, while it is $O(log(n))$ for amplitude encoding, which allows the encoding of exponentially more values in the same number of qubits. But to achieve this dense encoding, the circuit depth of amplitude encoding scales linearly with the number of values or exponentially with the number of qubits, while basis and angle encodings require a single gate per qubit.

By finding an appropriate encoding and decoding, QML can be used to solve join ordering and transaction scheduling problems and predict cardinalities. Such an approach for join ordering has already been developed [147, 148].

We developed an approach [148] for join order optimization using a VQC which encodes each table into a qubit with angle encoding and interprets the probability of each quantum state as the predicted reward for a join order. We choose the join order with the highest predicted reward. Figure 1 presents a code excerpt for realizing our approach[1] in Python (v. 3.8.12) with Qiskit (v. 0.19.1) and PyTorch (v. 1.11.0). In our experimental evaluation, we use queries joining 4 relations from the ErgastF1 benchmark [1], the SGD optimizer and the reward function $\frac{b}{c}$ with c the execution time

[1] https://github.com/TobiasWinker/QC4DB_VQC_Tutorial

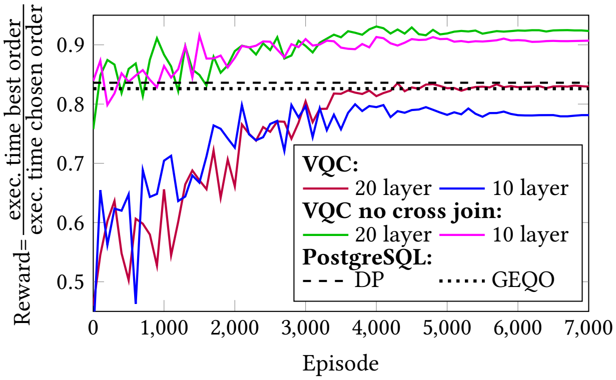

Figure 2: Comparison of the reward evolution during training for optimizing join orders with VQCs, DP and GEQO

of the chosen and b of the best join order. We additionally propose a variant that chooses only join orders without cross joins. In our experimental results depicted in Figure 2, a simulated 20-layer VQC achieved better join orders than PostgreSQL GEQO but worse than PostgreSQL DP. With the additional prevention of cross joins, the VQC reaches an average reward 10.5% higher than PostgreSQL DP.

5 FUTURE RESEARCH DIRECTIONS

Open challenges in quantum database applications are e.g.:

Developing Hybrid Algorithms: Some of the approaches mentioned in Section 4 are designed considering the limitations of the NISQ devices and are intrinsically hybrid. Other approaches are inherently suitable for combining with paradigms like branch and bound, divide and conquer or DP [120]. In such approaches, classical algorithms call quantum algorithms (working on smaller problem instances) to solve larger problems. Developing hybrid algorithms for database problems is an important focus for future research.

Designing Quantum Circuits: For implementation, circuit designs must be created from high-level designs of quantum algorithms. Some algorithms, such as Grover's search, assume the existence of a quantum "black box" evaluating a function using a quantum circuit. To run such algorithms on hardware, quantum circuits implementing the black box must be designed explicitly.

Compiling Quantum Circuits: A quantum circuit must be compiled to allow execution on quantum computers. The first reason is that a quantum computer is able to apply a limited set of gates [102]. Therefore, the arbitrary gates in a quantum circuit must be converted to combinations of applicable gates. Many such compiled circuits exist, and finding the optimal one is NP-complete [17]. When compiling, an efficient circuit must be chosen considering the hardware limitations. The second reason for compilation is the limited connectivity of qubits. Most algorithms assume two-qubit gates can be applied to any pair of qubits, which is not the case for real-world quantum hardware. Connectivity constraints might affect the complexity significantly [62, 139]. For each quantum method, an efficient way to compile circuits should be developed to reduce the latency in quantum database applications.

Developing Noise-Resilient Algorithms: Noise reduction might also be taken into account while designing algorithms, not only while compiling them. Quantum algorithms that depend on VQCs, such as VQA or QML, rely on explicit quantum circuits that can be designed to minimize the circuit depth and swap operations for a given problem, thus potentially reducing quantum hardware noise. To this end, the problem structure and symmetries [90, 94] or the capabilities of the hardware [78] may serve as guidelines. Similar studies may increase the performance of quantum databases.

Running Experiments on DBMSs: The performances of various quantum and classical methods for databases should be evaluated for different problem instances and parameter settings, by integrating them into some DBMS. This allows us to assess quantum speed-ups and choose the most suitable method for a given problem. These studies can also assess different computing models, such as cloud-based and on-site quantum computing.

Selecting the Best Approach: We do not expect a single algorithm to dominate all others for all problem instances. Instead, the focus should be on deciding the best approach for a given instance. Ideally, on a DBMS, this decision should be made at runtime. To be able to do this, first, the statistics for various methods and for various problem settings must be collected. A hybrid quantum algorithm, which utilizes CPU and QPU, might e.g. in some cases be outperformed by classical algorithms that use CPU [84, 125], GPU [59] or FPGA [145, 146]. The ability to dynamically choose the best method and hardware might increase the throughput significantly.

Quantum Acceleration of Other Database Problems: One might investigate quantum accelerating database problems where classical machine learning has been applied [75, 79, 107, 114] like learned indices, workload prediction, natural language interfaces to data, automating exploratory data analysis and data cleaning.

6 SUMMARY AND CONCLUSIONS

Various quantum algorithms can be used to solve database problems on NISQ hardware (see Table 1). The exact algorithms work by coherently evaluating some lengthy functions requiring additional qubits and long circuit depths. In the near future, these can only be employed for small problem instances or used as a subroutine for a hybrid algorithm. Most of the heuristic algorithms do not require ancilla qubits and all of them have small circuit depths. Those can be used to find near-optimal solutions to larger problem instances. For database applications, more than one approach can be implemented, and the DBMS can switch between them for different instances. This would give an additional speed-up to a quantum DBMS.

Some of the quantum approaches, such as quantum versions of nature-inspired heuristics and quantum machine learning, have an advantage over classical algorithms even when they are simulated on classical computers. This result shows the potential of quantum computing. Using the right approaches, even the most primitive quantum computers can improve the performance of many applications, including database management systems.

ACKNOWLEDGMENTS

This work is funded by the German Federal Ministry of Education and Research within the funding program quantum technologies - from basic research to market – contract number 13N16090.

REFERENCES

[1] 2023. ErgastF1 Benchmark. http://ergast.com/mrd/. Accessed: May 15, 2023.

[2] M. H. Abobeih, J. Randall, C. E. Bradley, H. P. Bartling, M. A. Bakker, M. J. Degen, M. Markham, D. J. Twitchen, and T. H. Taminiau. 2019. Atomic-scale imaging of a 27-nuclear-spin cluster using a quantum sensor. *Nature* 576, 7787 (2019), 411–415. https://doi.org/10.1038/s41586-019-1834-7

[3] R.K. Agrawal, Baljeet Kaur, and Surbhi Sharma. 2020. Quantum based Whale Optimization Algorithm for wrapper feature selection. *Applied Soft Computing* 89 (April 2020), 106092. https://doi.org/10.1016/j.asoc.2020.106092

[4] David Amaro, Carlo Modica, Matthias Rosenkranz, Mattia Fiorentini, Marcello Benedetti, and Michael Lubasch. 2022. Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology* 7, 1 (2022), 015021.

[5] David Amaro, Matthias Rosenkranz, Nathan Fitzpatrick, Koji Hirano, and Mattia Fiorentini. 2022. A case study of variational quantum algorithms for a job shop scheduling problem. *EPJ Quantum Technology* 9, 1 (2022), 5.

[6] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. 2019. Quantum Speedups for Exponential-Time Dynamic Programming Algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 1783–1793. https://doi.org/10.1137/1.9781611975482.107

[7] Luigi Amico, Dana Anderson, Malcolm Boshier, Jean-Philippe Brantut, Leong-Chuan Kwek, Anna Minguzzi, and Wolf von Klitzing. 2022. Colloquium: Atomtronic circuits: From many-body physics to quantum technologies. *Reviews of Modern Physics* 94, 4 (2022), 041001.

[8] L. Amico, M. Boshier, G. Birkl, A. Minguzzi, C. Miniatura, L.-C. Kwek, D. Aghamalyan, V. Ahufinger, D. Anderson, N. Andrei, A. S. Arnold, M. Baker, T. A. Bell, T. Bland, J. P. Brantut, D. Cassettari, W. J. Chetcuti, F. Chevy, R. Citro, S. De Palo, R. Dumke, M. Edwards, R. Folman, J. Fortagh, S. A. Gardiner, B. M. Garraway, G. Gauthier, A. Günther, T. Haug, C. Hufnagel, M. Keil, P. Ireland, M. Lebrat, W. Li, L. Longchambon, J. Mompart, O. Morsch, P. Naldesi, T. W. Neely, M. Olshanii, E. Orignac, S. Pandey, A. Pérez-Obiol, H. Perrin, L. Piroli, J. Polo, A. L. Pritchard, N. P. Proukakis, C. Rylands, H. Rubinsztein-Dunlop, F. Scazza, S. Stringari, F. Tosto, A. Trombettoni, N. Victorin, W. Von Klitzing, D. Wilkowski, K. Xhani, and A. Yakimenko. 2021. Roadmap on Atomtronics: State of the art and perspective. *AVS Quantum Sci.* 3, 3 (sep 2021), 039201. https://doi.org/10.1116/5.0026178 arXiv:2008.04439

[9] MD SAJID ANIS and et al. 2021. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505 See full list of authors on Github: https://github.com/Qiskit/qiskit/blob/master/AUTHORS.

[10] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (oct 2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[11] Richard Bellman. 1962. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)* 9, 1 (1962), 61–63.

[12] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M. Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, and et al. 2020. Pennylane: Automatic differentiation of hybrid quantum-classical computations. https://arxiv.org/abs/1811.04968v3

[13] Dimitris Bertsimas and John Tsitsiklis. 1993. Simulated annealing. *Statistical science* 8, 1 (1993), 10–15.

[14] Chris M Bishop. 1994. Neural networks and their applications. *Review of scientific instruments* 65, 6 (1994), 1803–1832.

[15] Tim Bittner and Sven Groppe. 2020. Avoiding blocking by scheduling transactions using quantum annealing. In *Proceedings of the 24th Symposium on International Database Engineering & Applications*. ACM, 1–10. https://doi.org/10.1145/3410566.3410593

[16] Tim Bittner and Sven Groppe. 2020. Hardware Accelerating the Optimization of Transaction Schedules via Quantum Annealing by Avoiding Blocking. *Open Journal of Cloud Computing (OJCC)* 7, 1 (2020), 1–21. http://nbn-resolving.de/urn:nbn:de:101:1-2020112218332015343957

[17] Adi Botea, Akihiro Kishimoto, and Radu Marinescu. 2018. On the complexity of quantum circuit compilation. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 9. 138–142.

[18] Francesco Bova, Avi Goldfarb, and Roger G. Melko. 2021. Commercial applications of quantum computing. *EPJ Quantum Technol.* 8, 1 (2021), 1–13. https://doi.org/10.1140/epjqt/s40507-021-00091-1

[19] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. 1998. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics* 46, 4-5 (1998), 493–505.

[20] C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau. 2019. A Ten-Qubit Solid-State Spin Register with Quantum Memory up to One Minute. *Phys. Rev. X* 9, 3 (sep 2019), 031045. https://doi.org/10.1103/PhysRevX.9.031045

[21] Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles. 2019. Variational Quantum Linear Solver. (2019). https://doi.org/10.48550/ARXIV.1909.05820

[22] Michael J Carey and Waleed A Muhanna. 1986. The performance of multiversion concurrency control algorithms. *ACM Transactions on Computer Systems (TOCS)* 4, 4 (1986), 338–378.

[23] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.

[24] Samuel Yen-Chi Chen, Chih-Min Huang, Chia-Wei Hsing, Hsi-Sheng Goan, and Ying-Jer Kao. 2022. Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology* 3, 1 (2 2022), 015025. https://doi.org/10.1088/2632-2153/ac4559

[25] YunHeng Chen, Sophie Stearn, Scott Vella, Andrew Horsley, and Marcus W Doherty. 2020. Optimisation of diamond quantum processors. *New Journal of Physics* 22, 9 (2020), 093068.

[26] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. 2017. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM J. Comput.* 46, 6 (2017), 1920–1950. https://doi.org/10.1137/16m1087072

[27] J. I. Cirac and P. Zoller. 1995. Quantum Computations with Cold Trapped Ions. *Phys. Rev. Lett.* 74, 20 (may 1995), 4091–4094. https://doi.org/10.1103/PhysRevLett.74.4091

[28] Sam R Cohen and Jeff D Thompson. 2021. Quantum computing with circular Rydberg atoms. *PRX Quantum* 2, 3 (2021), 030322.

[29] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, Francisco J. Varela, and Paul Emile Bourgine. 1991. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Paris, France). 134–142.

[30] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. 1997. Ensemble quantum computing by NMR spectroscopy. *Proc. Natl. Acad. Sci.* 94, 5 (mar 1997), 1634–1639. https://doi.org/10.1073/pnas.94.5.1634

[31] A. J. Dahm, J. M. Goodkind, I. Karakurt, and S. Pilla. 2002. Using electrons on liquid helium for quantum computing. *J. Low Temp. Phys.* 126, 1-2 (2002), 709–718. https://doi.org/10.1023/A:1013764511968 arXiv:0111029 [quant-ph]

[32] Ren Qing dao-er ji and Yuping Wang. 2012. A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research* 39, 10 (Oct. 2012), 2291–2299. https://doi.org/10.1016/j.cor.2011.12.005

[33] Nathalie P de Leon, Kohei M Itoh, Dohun Kim, Karan K Mehta, Tracy E Northup, Hanhee Paik, BS Palmer, Nitin Samarth, Sorawis Sangtawesin, and David W Steuerman. 2021. Materials challenges and opportunities for quantum computing hardware. *Science* 372, 6539 (2021), eabb2823.

[34] D. Deutsch. 1985. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. London. A. Math. Phys. Sci.* 400, 1818 (jul 1985), 97–117. https://doi.org/10.1098/rspa.1985.0070

[35] Cirq Developers. 2022. Cirq. https://doi.org/10.5281/zenodo.6599601 See full list of authors on Github: https://github.com/quantumlib/Cirq/graphs/contributors.

[36] M. H. Devoret, A. Wallraff, and J. M. Martinis. 2004. Superconducting Qubits: A Short Review. (nov 2004). arXiv:0411174 [cond-mat] http://arxiv.org/abs/cond-mat/0411174

[37] Berk Diler, Samuel J. Whiteley, Christopher P. Anderson, Gary Wolfowicz, Marie E. Wesson, Edward S. Bielejec, F. Joseph Heremans, and David D. Awschalom. 2020. Coherent control and high-fidelity readout of chromium ions in commercial silicon carbide. *npj Quantum Inf.* 6, 1 (dec 2020), 11. https://doi.org/10.1038/s41534-020-0247-7

[38] David P. DiVincenzo. 2000. The Physical Implementation of Quantum Computation. *Fortschritte der Phys.* 48, 9-11 (sep 2000), 771–783. https://doi.org/10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E

[39] Marcus Doherty. 2021. Quantum accelerators: a new trajectory of quantum computers. *Digitale Welt* 5, 2 (March 2021), 74–79. https://doi.org/10.1007/s42354-021-0342-8

[40] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. 2020. Expressive power of parametrized quantum circuits. *Physical Review Research* 2, 3 (jul 2020), 033125. https://doi.org/10.1103/physrevresearch.2.033125

[41] Christoph Dürr and Peter Høyer. 1996. A Quantum Algorithm for Finding the Minimum. https://doi.org/10.48550/ARXIV.QUANT-PH/9607014

[42] Sean R Eddy. 2004. What is dynamic programming? *Nature Biotechnology* 22, 7 (July 2004), 909–910. https://doi.org/10.1038/nbt0704-909
[43] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm. (nov 2014). arXiv:1411.4028 http://arxiv.org/abs/1411.4028
[44] Richard P Feynman. 1982. Simulating Physics with Computers. *International Journal of Theoretical Physics* 21, 6/7 (1982), 467–488.
[45] Aleta Berk Finnila, MA Gomez, C Sebenik, Catherine Stenson, and Jimmie D Doll. 1994. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical physics letters* 219, 5-6 (1994), 343–348.
[46] Quantum Open Source Foundation. 2022. List of open quantum projects. https://qosf.org/project_list/
[47] A. Gaita-Ariño, F. Luis, S. Hill, and E. Coronado. 2019. Molecular spins for quantum computation. *Nat. Chem.* 11, 4 (apr 2019), 301–309. https://doi.org/10.1038/s41557-019-0232-y
[48] Craig Gidney and Martin Ekerå. 2021. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* 5 (2021), 1–31. https://doi.org/10.22331/Q-2021-04-15-433 arXiv:1905.09749
[49] Fred Glover, Gary Kochenberger, and Yu Du. 2018. A tutorial on formulating and using QUBO models. *arXiv preprint arXiv:1811.11538* (2018).
[50] T. M. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemeyer, M. Kwon, M. Ebert, J. Cherek, M. T. Lichtman, M. Gillette, J. Gilbert, D. Bowman, T. Ballance, C. Campbell, E. D. Dahl, O. Crawford, N. S. Blunt, B. Rogers, T. Noel, and M. Saffman. 2022. Multi-qubit entanglement and algorithms on a neutral-atom quantum computer. *Nature* 604, 7906 (apr 2022), 457–462. https://doi.org/10.1038/s41586-022-04603-6 arXiv:arXiv:2112.14589v3
[51] Joaquim A.S. Gromicho, Jelke J. van Hoorn, Francisco Saldanha da Gama, and Gerrit T. Timmer. 2012. Solving the job-shop scheduling problem optimally by dynamic programming. *Computers & Operations Research* 39, 12 (Dec. 2012), 2968–2977. https://doi.org/10.1016/j.cor.2012.02.024
[52] Sven Groppe and Jinghua Groppe. 2021. Optimizing Transaction Schedules on Universal Quantum Computers via Code Generation for Grover's Search Algorithm. In *IDEAS 2021: 25th International Database Engineering and Applications Symposium, Montreal, QC, Canada*. 149–156. https://doi.org/10.1145/3472163.3472164
[53] Lov K. Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*. ACM Press, 212–219. https://doi.org/10.1145/237814.237866
[54] Jinyoung Ha, Jonghyun Lee, and Jun Heo. 2022. Resource analysis of quantum computing with noisy qubits for Shor's factoring algorithms. *Quantum Inf. Process.* 21, 2 (2022), 1–19. https://doi.org/10.1007/s11128-021-03398-1
[55] Yuxing Han, Ziniu Wu, Peizhi Wu, Rong Zhu, Jingyi Yang, Liang Wei Tan, Kai Zeng, Gao Cong, Yanzhao Qin, Andreas Pfadler, Zhengping Qian, Jingren Zhou, Jiangneng Li, and Bin Cui. 2021. Cardinality Estimation in DBMS: A Comprehensive Benchmark Evaluation. https://doi.org/10.48550/ARXIV.2109.05877
[56] Theo Härder. 1984. Observations on optimistic concurrency control schemes. *Information Systems* 9, 2 (1984), 111–120.
[57] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. 2010. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* 82, 2 (jul 2010), 024511. https://doi.org/10.1103/PhysRevB.82.024511 arXiv:1004.1628
[58] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* 103, 15 (oct 2009), 150502. https://doi.org/10.1103/PhysRevLett.103.150502
[59] Max Heimel and Volker Markl. 2012. A First Step Towards GPU-assisted Query Optimization. *ADMS@ VLDB* 2012 (2012), 33–44.
[60] Michael Held and Richard M Karp. 1962. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics* 10, 1 (1962), 196–210.
[61] John H. Holland. 1992. Genetic algorithms. *Scientific american* 267, 1 (1992), 66–73.
[62] Adam Holmes, Sonika Johri, Gian Giacomo Guerreschi, James S Clarke, and Anne Y Matsuura. 2020. Impact of qubit connectivity on quantum algorithm performance. *Quantum Science and Technology* 5, 2 (2020), 025009.
[63] Jorng-Tzong Horng, Cheng-Yan Kao, and Baw-Jhiune Liu. 1994. A genetic algorithm for database query optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, 350–355. https://doi.org/10.1109/icec.1994.349926
[64] Kuo-Ling Huang and Ching-Jong Liao. 2008. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research* 35, 4 (April 2008), 1030–1046. https://doi.org/10.1016/j.cor.2006.07.003
[65] Ellis L Johnson, George L Nemhauser, and Martin WP Savelsbergh. 2000. Progress in linear programming-based algorithms for integer programming: An exposition. *Informs journal on computing* 12, 1 (2000), 2–23.
[66] Tadashi Kadowaki and Hidetoshi Nishimori. 1998. Quantum annealing in the transverse Ising model. *Physical Review E* 58, 5 (1998), 5355.
[67] Bogumił Kamiński, Michał Jakubczyk, and Przemysław Szufel. 2018. A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research* 26, 1 (01 Mar 2018), 135–159. https://doi.org/10.1007/s10100-017-0479-6
[68] H. G. Katzgraber and R. S. Andrist. 2013. Stability of topologically-protected quantum computing proposals as seen through spin glasses. *J. Phys. Conf. Ser.* 473, 1 (dec 2013), 012019. https://doi.org/10.1088/1742-6596/473/1/012019 arXiv:1306.0540
[69] James Kennedy and Russell C Eberhart. 1997. A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, Vol. 5. IEEE, 4104–4108.
[70] Andrew D King, Sei Suzuki, Jack Raymond, Alex Zucca, Trevor Lanting, Fabio Altomare, Andrew J Berkley, Sara Ejtemaee, Emile Hoskinson, Shuiyuan Huang, et al. 2022. Coherent quantum annealing in a programmable 2,000 qubit Ising chain. *Nature Physics* 18, 11 (2022), 1324–1328.
[71] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.
[72] A.Yu. Kitaev. 2003. Fault-tolerant quantum computation by anyons. *Ann. Phys. (N. Y).* 303, 1 (jan 2003), 2–30. https://doi.org/10.1016/S0003-4916(02)00018-0
[73] Wen-Yang Ku and J Christopher Beck. 2016. Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research* 73 (2016), 165–173.
[74] Michael Kühn, Sebastian Zanker, Peter Deglmann, Michael Marthaler, and Horst Weiß. 2019. Accuracy and Resource Estimations for Quantum Chemistry on a Near-Term Quantum Computer. *J. Chem. Theory Comput.* 15, 9 (2019), 4764–4780. https://doi.org/10.1021/acs.jctc.9b00236 arXiv:1812.06814
[75] Arun Kumar, Matthias Boehm, and Jun Yang. 2017. Data Management in Machine Learning: Challenges, Techniques, and Systems. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1717–1722. https://doi.org/10.1145/3035918.3054775
[76] Ville Lahtinen and Jiannis Pachos. 2017. A Short Introduction to Topological Quantum Computation. *SciPost Phys.* 3, 3 (sep 2017), 021. https://doi.org/10.21468/SciPostPhys.3.3.021 arXiv:1705.04103
[77] Jan Karel Lenstra, AHG Rinnooy Kan, and Peter Brucker. 1977. Complexity of machine scheduling problems. In *Annals of discrete mathematics*. Vol. 1. Elsevier, 343–362.
[78] Lorenzo Leone, Salvatore FE Oliviero, Lukasz Cincio, and M Cerezo. 2022. On the practical usefulness of the Hardware Efficient Ansatz. *arXiv preprint arXiv:2211.01477* (2022).
[79] Guoliang Li, Xuanhe Zhou, and Lei Cao. 2021. AI meets database: AI4DB and DB4AI. In *Proceedings of the 2021 International Conference on Management of Data*. 2859–2866.
[80] Ya Li, Yichao He, Xuejing Liu, Xiaohu Guo, and Zewen Li. 2020. A novel discrete whale optimization algorithm for solving knapsack problems. *Applied Intelligence* 50, 10 (2020), 3350–3366.
[81] Tsung-Lieh Lin, Shi-Jinn Horng, Tzong-Wann Kao, Yuan-Hsin Chen, Ray-Shine Run, Rong-Jian Chen, Jui-Lin Lai, and I-Hong Kuo. 2010. An efficient job-shop scheduling algorithm based on particle swarm optimization. *Expert Systems with Applications* 37, 3 (2010), 2629–2636.
[82] Min Liu, Feng Zhang, Yunlong Ma, Hemanshu Roy Pota, and Weiming Shen. 2016. Evacuation path optimization based on quantum ant colony algorithm. *Advanced Engineering Informatics* 30, 3 (2016), 259–267.
[83] Daniel Loss and David P. DiVincenzo. 1998. Quantum computation with quantum dots. *Phys. Rev. A* 57, 1 (jan 1998), 120–126. https://doi.org/10.1103/PhysRevA.57.120 arXiv:9701055 [cond-mat]
[84] Gang Luo, Jeffrey F Naughton, Curt J Ellmann, and Michael W Watzke. 2010. Transaction reordering. *Data & Knowledge Engineering* 69, 1 (2010), 29–49.
[85] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi. Askarani, Fabien Rortais, Trevor Vincent, Jacob F. F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, Adriana E. Lita, Thomas Gerrits, Sae Woo Nam, Varun D. Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. 2022. Quantum computational advantage with a programmable photonic processor. *Nature* 606, 7912 (jun 2022), 75–81. https://doi.org/10.1038/s41586-022-04725-x
[86] A. Malossini, E. Blanzieri, and T. Calarco. 2008. Quantum genetic optimization. *IEEE Transactions on Evolutionary Computation* 12, 2 (2008), 231–241.
[87] Alan S Manne. 1960. On the job-shop scheduling problem. *Operations research* 8, 2 (1960), 219–223.
[88] Ryan Marcus and Olga Papaemmanouil. 2018. Deep Reinforcement Learning for Join Order Enumeration. In *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. ACM, 1–4. https://doi.org/10.1145/3211954.3211957

[89] Diana Martinez-Mosquera, Rosa Navarrete, and Sergio Lujan-Mora. 2020. Modeling and management big data in databases—A systematic literature review. *Sustainability* 12, 2 (2020), 634.

[90] Atsushi Matsuo, Yudai Suzuki, and Shigeru Yamashita. 2020. Problem-specific parameterized quantum circuits of the VQE algorithm for optimization problems. *arXiv preprint arXiv:2006.05643* (2020).

[91] Dastan Maulud and Adnan M Abdulazeez. 2020. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends* 1, 4 (2020), 140–147.

[92] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. 2019. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information* 5, 1 (2019), 1–6.

[93] Alexander McCaskey, Eugene Dumitrescu, Dmitry Liakh, and Travis Humble. 2018. Hybrid Programming for Near-Term Quantum Computing Systems. In *2018 IEEE Int. Conf. Rebooting Comput.* IEEE, 1–12. https://doi.org/10.1109/ICRC.2018.8638598

[94] Johannes Jakob Meyer, Marian Mularski, Elies Gil-Fuster, Antonio Anna Mele, Francesco Arzani, Alissa Wilms, and Jens Eisert. 2023. Exploiting symmetry in variational quantum machine learning. *PRX Quantum* 4, 1 (2023), 010328.

[95] Xiao Mingyao and Li Xiongfei. 2015. Embedded Database Query Optimization Algorithm Based on Particle Swarm Optimization. In *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*. 429–432. https://doi.org/10.1109/ICMTMA.2015.109

[96] Seyedali Mirjalili and Andrew Lewis. 2016. The Whale Optimization Algorithm. *Advances in Engineering Software* 95 (May 2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[97] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. 2019. Full-stack, real-system Quantum Computer Studies. In *Proceedings of the 46th International Symposium on Computer Architecture*. 527–540. https://doi.org/10.1145/3307650.3322273

[98] Giacomo Nannicini. 2021. Fast Quantum Subroutines for the Simplex Method. In *Integer Programming and Combinatorial Optimization: 22nd International Conference, IPCO 2021, Atlanta, GA, USA, May 19–21, 2021, Proceedings 22*. Springer, 311–325.

[99] Ajit Narayanan and Mark Moore. 1996. Quantum-inspired genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation*. IEEE, 61–66.

[100] J.C. Nash. 2000. The (Dantzig) simplex method for linear programming. *Computing in Science & Engineering* 2, 1 (2000), 29–31. https://doi.org/10.1109/5992.814654

[101] Nitin Nayak, Jan Rehfeld, Tobias Winker, Benjamin Warnke, Umut Çalıkyılmaz, and Sven Groppe. 2023. Constructing Optimal Bushy Join Trees by Solving QUBO Problems on Quantum Hardware and Simulators. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments (BiDEDE), Seattle, WA, USA*. https://doi.org/10.1145/3579142.3594298

[102] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.

[103] Peter Nimbe, Benjamin Asubam Weyori, and Adebayo Felix Adekoya. 2021. Models in quantum computing: a systematic review. *Quantum Information Processing* 20, 2 (2021), 80.

[104] William S. Noble. 2006. What is a support vector machine? *Nature Biotechnology* 24, 12 (01 Dec 2006), 1565–1567. https://doi.org/10.1038/nbt1206-1565

[105] Jeremy L. O'Brien. 2007. Optical Quantum Computing. *Science (80-.).* 318, 5856 (dec 2007), 1567–1570. https://doi.org/10.1126/science.1142892

[106] Jeremy L. O'Brien, Akira Furusawa, and Jelena Vučković. 2009. Photonic quantum technologies. *Nat. Photonics* 3, 12 (dec 2009), 687–695. https://doi.org/10.1038/nphoton.2009.229

[107] Fatma Őzcan, Abdul Quamar, Jaydeep Sen, Chuan Lei, and Vasilis Efthymiou. 2020. State of the Art and Open Challenges in Natural Language Interfaces to Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2629–2636. https://doi.org/10.1145/3318464.3383128

[108] Daniel K Park, Francesco Petruccione, and June-Koo Kevin Rhee. 2019. Circuit-based quantum random access memory for classical data. *Scientific reports* 9, 1 (2019), 1–8.

[109] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* 5, 1 (sep 2014), 4213. https://doi.org/10.1038/ncomms5213

[110] L. Petit, H. G. J. Eenink, M. Russ, W. I. L. Lawrie, N. W. Hendrickx, S. G. J. Philips, J. S. Clarke, L. M. K. Vandersypen, and M. Veldhorst. 2020. Universal quantum logic in hot silicon qubits. *Nature* 580, 7803 (apr 2020), 355–359. https://doi.org/10.1038/s41586-020-2170-7

[111] Sébastien Pezzagna and Jan Meijer. 2021. Quantum computer based on color centers in diamond. *Appl. Phys. Rev.* 8, 1 (mar 2021), 011308. https://doi.org/10.1063/5.0007444

[112] Dmitry I. Pikulin, Bernard van Heck, Torsten Karzig, Esteban A. Martinez, Bas Nijholt, Tom Laeven, Georg W. Winkler, John D. Watson, Sebastian Heedt, Mine Temurhan, Vicky Svidenko, Roman M. Lutchyn, Mason Thomas, Gijs de Lange, Lucas Casparis, and Chetan Nayak. 2021. Protocol to identify a topological superconducting phase in a three-terminal device. (mar 2021), 1–28. arXiv:2103.12217 http://arxiv.org/abs/2103.12217

[113] Riccardo Poli, James Kennedy, and Tim Blackwell. 2007. Particle swarm optimization. *Swarm intelligence* 1, 1 (2007), 33–57.

[114] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. 2017. Data Management Challenges in Production Machine Learning. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1723–1726. https://doi.org/10.1145/3035918.3054782

[115] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (aug 2018), 79. https://doi.org/10.22331/q-2018-08-06-79

[116] Mucoadhesive Properties and Human Oral Mucosa. 2017. Technical Report : Technical Report :. *Pulse Lab Jakarta* 3rd Resear, May (2017), 22–25. https://beta.unglobalpulse.org/wp-content/uploads/2017/05/3rd-Research-Dive.pdf{#}page=29

[117] Saeed K Rahimi and Frank S Haug. 2010. *Distributed database management systems*. Wiley-Blackwell.

[118] Raghu Ramakrishnan, Johannes Gehrke, and Johannes Gehrke. 2003. *Database management systems*. Vol. 3. McGraw-Hill New York.

[119] Arthur G. Rattew, Shaohan Hu, Marco Pistoia, Richard Chen, and Steve Wood. 2020. A Domain-agnostic, Noise-resistant, Hardware-efficient Evolutionary Variational Quantum Eigensolver. arXiv:1910.09694 [quant-ph]

[120] Pooya Ronagh. 2019. The Problem of Dynamic Programming on a Quantum Computer. https://doi.org/10.48550/ARXIV.1906.02229

[121] Marie Salm, Johanna Barzen, Frank Leymann, Benjamin Weder, and Karoline Wild. 2021. Automating the Comparison of Quantum Compilers for Quantum Circuits. In *Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2021)*. Springer International Publishing, 64–80. https://doi.org/10.1007/978-3-030-87568-8_4

[122] Giordano Scappucci, Christoph Kloeffel, Floris A. Zwanenburg, Daniel Loss, Maksym Myronov, Jian-Jun Zhang, Silvano De Franceschi, Georgios Katsaros, and Menno Veldhorst. 2021. The germanium quantum information route. *Nat. Rev. Mater.* 6, 10 (oct 2021), 926–943. https://doi.org/10.1038/s41578-020-00262-z

[123] Manuel Schönberger. 2022. Applicability of Quantum Computing on Database Query Optimization. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD), Philadelphia, PA, USA (SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 2512–2514. https://doi.org/10.1145/3514221.3520257

[124] Manuel Schönberger, Stefanie Scherzinger, Wolfgang Mauerer, Karen Wintersperger, Hila Safi, Wolfgang Mauerer, Maja Franz, Lucas Wolf, M Periyasamy, Ch Ufrecht, et al. 2023. Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware. In *Proceedings of ACM SIGMOD/PODS International Conference on Management of Data*. Gesellschaft für Informatik.

[125] P. Griffiths Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. 1979. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD international conference on Management of data - SIGMOD '79*. ACM Press, 23–34. https://doi.org/10.1145/582095.582099

[126] Peter W Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 124–134.

[127] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. 2020. Quantum Natural Gradient. *Quantum* 4 (5 2020), 269. https://doi.org/10.22331/q-2020-05-25-269

[128] Martin Suchara, John Kubiatowicz, Arvin Faruque, Frederic T. Chong, Ching Yi Lai, and Gerardo Paz. 2013. QuRE: The quantum resource estimator toolbox. *2013 IEEE 31st Int. Conf. Comput. Des. ICCD 2013* (2013), 419–426. https://doi.org/10.1109/ICCD.2013.6657074

[129] Jun Sun, Bin Feng, and Wenbo Xu. 2004. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, Vol. 1. IEEE, 325–331.

[130] Murray Thom. 2021. Solving problems with quantum annealing. *Digit. Welt* 5, 2 (apr 2021), 70–73. https://doi.org/10.1007/s42354-021-0341-9

[131] Alexander Thomasian and In Kyung Ryu. 1991. Performance analysis of two-phase locking. *IEEE transactions on software engineering* 17, 5 (1991), 386.

[132] Preeti Tiwari and Swati V. Chande. 2018. Optimal Ant and Join Cardinality for Distributed Query Optimization Using Ant Colony Optimization Algorithm. In *Advances in Intelligent Systems and Computing*. Springer Singapore, 385–392. https://doi.org/10.1007/978-981-13-2285-3_45

[133] Immanuel Trummer and Christoph Koch. 2015. Multiple query optimization on the D-Wave 2X adiabatic quantum computer. *arXiv preprint arXiv:1510.06437* (2015).

[134] Immanuel Trummer and Christoph Koch. 2016. Multiple query optimization on the D-Wave 2X adiabatic quantum computer. *Proceedings of the VLDB Endowment* 9, 9 (May 2016), 648–659. https://doi.org/10.14778/2947618.2947621

[135] Immanuel Trummer and Christoph Koch. 2017. Solving the join ordering problem via mixed integer linear programming. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1025–1040.

[136] Yushi UNO and Toshihide IBARAKI. 1991. Complexity of the optimum join order problem in relational databases. *IEICE TRANSACTIONS on Information and Systems* 74, 7 (1991), 2067–2075.

[137] Valter Uotila. 2022. Synergy between Quantum Computers and Databases. In *Proceedings of the VLDB 2022 PhD Workshop co-located with the 48th International Conference on Very Large Databases (VLDB 2022), Sydney, Australia (CEUR Workshop Proceedings)*, Zhifeng Bao and Timos K. Sellis (Eds.), Vol. 3186. CEUR-WS.org. http://ceur-ws.org/Vol-3186/paper_1.pdf

[138] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak. 2017. Silicon CMOS architecture for a spin-based quantum computer. *Nat. Commun.* 8, 1 (dec 2017), 1766. https://doi.org/10.1038/s41467-017-01905-6 arXiv:1609.09700

[139] Daniel Vert, Renaud Sirdey, and Stephane Louise. 2019. On the limitations of the chimera graph topology in using analog quantum computers. In *Proceedings of the 16th ACM international conference on computing frontiers*. 226–229.

[140] Juan Pablo Vielma. 2015. Mixed integer linear programming formulation techniques. *Siam Review* 57, 1 (2015), 3–57.

[141] Huaixiao Wang, Jianyong Liu, Jun Zhi, and Chengqun Fu. 2013. The Improvement of Quantum Genetic Algorithm and Its Application on Function Optimization. *Mathematical Problems in Engineering* 2013 (2013), 1–10. https://doi.org/10.1155/2013/730749

[142] Ling Wang, Qun Niu, and Minrui Fei. 2007. A Novel Quantum Ant Colony Optimization Algorithm. In *Bio-Inspired Computational Intelligence and Applications*. Springer Berlin Heidelberg, 277–286. https://doi.org/10.1007/978-3-540-74769-7_31

[143] J. R. Weber, W. F. Koehl, J. B. Varley, A. Janotti, B. B. Buckley, C. G. Van de Walle, and D. D. Awschalom. 2011. Defects in SiC for quantum computing. *J. Appl. Phys.* 109, 10 (may 2011), 102417. https://doi.org/10.1063/1.3578264

[144] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. 2021. Expanding Data Encoding Patterns For Quantum Algorithms. In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 95–101. https://doi.org/10.1109/ICSA-C52384.2021.00025

[145] Stefan Werner, Dennis Heinrich, Sven Groppe, Christopher Blochwitz, and Thilo Pionteck. 2016. Runtime Adaptive Hybrid Query Engine based on FPGAs. *OJDB* 3, 1 (2016), 21–41. http://nbn-resolving.de/urn:nbn:de:101:1-201705194645

[146] Stefan Werner, Dennis Heinrich, Thilo Pionteck, and Sven Groppe. 2017. Semi-static operator graphs for accelerated query execution on FPGAs. *Microprocessors and Microsystems - Embedded Hardware Design* 53 (2017), 178–189. https://doi.org/10.1016/j.micpro.2017.07.010

[147] Tobias Winker, Sven Groppe, Valter Uotila, Zhengtong Yan, Jiaheng Lu, Maja Franz, and Wolfgang Mauerer. 2023. Quantum Machine Learning: Foundation, New Techniques, and Opportunities for Database Research. In *Proceedings of ACM SIGMOD/PODS International Conference on Management of Data (SIGMOD)*. https://doi.org/10.1145/3555041.3589404

[148] Tobias Winker, Umut Çalıkyılmaz, Le Gruenwald, and Sven Groppe. 2023. Quantum Machine Learning for Join Order Optimization using Variational Quantum Circuits. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments (BiDEDE), Seattle, WA, USA*. https://doi.org/10.1145/3579142.3594299

[149] Xanadu. 2022. Pennylane. https://pennylane.ai/index.html

[150] Xiang Yu, Guoliang Li, Chengliang Chai, and Nan Tang. 2020. Reinforcement Learning with Tree-LSTM for Join Order Selection. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1297–1308. https://doi.org/10.1109/icde48307.2020.00116

[151] Linghua Zhu, Ho Lun Tang, George S. Barron, F. A. Calderon-Vargas, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. 2022. An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. arXiv:2005.10258 [quant-ph]

[152] A. M. J. Zwerver, T. Krähenmann, T. F. Watson, L. Lampert, H. C. George, R. Pillarisetty, S. A. Bojarski, P. Amin, S. V. Amitonov, J. M. Boter, R. Caudillo, D. Correas-Serrano, J P Dehollain, G Droulers, E M Henry, R Kotlyar, M Lodari, F Lüthi, D J Michalak, B K Mueller, S Neyens, J Roberts, N Samkharadze, G Zheng, O K Zietz, G Scappucci, M Veldhorst, L M K Vandersypen, and J S Clarke. 2022. Qubits made by advanced semiconductor manufacturing. *Nat. Electron.* 5, 3 (mar 2022), 184–190. https://doi.org/10.1038/s41928-022-00727-9