Challenges and Opportunities in Quantum Optimization[*]

Amira Abbas,[1] Andris Ambainis,[2] Brandon Augustino,[3] Andreas Bärtschi,[4] Harry Buhrman,[1] Carleton Coffrin,[4] Giorgio Cortiana,[5] Vedran Dunjko,[6] Daniel J. Egger,[7] Bruce G. Elmegreen,[8] Nicola Franco,[9] Filippo Fratini,[10] Bryce Fuller,[11] Julien Gacon,[7,12] Constantin Gonciulea,[13] Sander Gribling,[14] Swati Gupta,[3] Stuart Hadfield,[15,16] Raoul Heese,[17] Gerhard Kircher,[10] Thomas Kleinert,[18] Thorsten Koch,[19,20] Georgios Korpas,[21,22] Steve Lenk,[23] Jakub Marecek,[22] Vanio Markov,[13] Guglielmo Mazzola,[24] Stefano Mensa,[25] Naeimeh Mohseni,[5] Giacomo Nannicini,[26] Corey O'Meara,[5] Elena Peña Tapia,[7] Sebastian Pokutta,[19,20] Manuel Proissl,[7] Patrick Rebentrost,[27] Emre Sahin,[25] Benjamin C. B. Symons,[25] Sabine Tornow,[28] Víctor Valls,[29] Stefan Woerner,[7] Mira L. Wolf-Bauwens,[7] Jon Yard,[30] Sheir Yarkoni,[31] Dirk Zechiel,[18] Sergiy Zhuk,[29] and Christa Zoufal[7]

[1]*QuSoft and University of Amsterdam*
[2]*University of Latvia*
[3]*Massachusetts Institute of Technology*
[4]*Los Alamos National Laboratory*
[5]*E.ON Digital Technology GmbH*
[6]*Leiden University*
[7]*IBM Quantum, IBM Research Europe – Zurich*
[8]*IBM Research, IBM T.J. Watson Research Center*
[9]*Fraunhofer IKS*
[10]*Erste Group Bank*
[11]*IBM Quantum, IBM T.J. Watson Research Center*
[12]*École Polytechnique Fédérale de Lausanne*
[13]*Wells Fargo*
[14]*Tilburg University*
[15]*Quantum Artificial Intelligence Lab, NASA Ames Research Center*
[16]*USRA Research Institute for Advanced Computer Science*
[17]*Fraunhofer ITWM*
[18]*Quantagonia*
[19]*Zuse Institute Berlin*
[20]*Technische Universität Berlin*
[21]*HSBC Lab, Innovation and Ventures, HSBC, London*
[22]*Czech Technical University in Prague*
[23]*Fraunhofer IOSB-AST*
[24]*University of Zurich*
[25]*The Hartree Centre, STFC*
[26]*University of Southern California*
[27]*Centre for Quantum Technologies, National University of Singapore*
[28]*University of the Bundeswehr Munich*
[29]*IBM Quantum, IBM Research Europe – Dublin*
[30]*Institute for Quantum Computing, Perimeter Institute for Theoretical Physics, University of Waterloo*
[31]*Volkswagen AG*
(Dated: November 19, 2024)

Recent advances in quantum computers are demonstrating the ability to solve problems at a scale beyond brute force classical simulation. As such, a widespread interest in quantum algorithms has developed in many areas, with optimization being one of the most pronounced domains. Across computer science and physics, there are a number of different approaches for major classes of optimization problems, such as combinatorial optimization, convex optimization, non-convex optimization, and stochastic extensions. This work draws on multiple approaches to study quantum optimization. Provably exact versus heuristic settings are first explained using computational complexity theory – highlighting where quantum advantage is possible in each context. Then, the core building blocks for quantum optimization algorithms are outlined to subsequently define prominent problem classes and identify key open questions that, if answered, will advance the field. The effects of scaling relevant problems on noisy quantum devices are also outlined in detail, alongside meaningful benchmarking problems. We underscore the importance of benchmarking by proposing clear metrics to conduct appropriate comparisons with classical optimization techniques. Lastly, we highlight two domains – finance and sustainability – as rich sources of optimization problems that could be used to benchmark, and eventually validate, the potential real-world impact of quantum optimization.

Contents

─────────

* Summary of the Quantum Optimization Working Group.
  Authors are listed in alphabetical order.
  Contact: Stefan Woerner (`wor@zurich.ibm.com`)

I. INTRODUCTION

Quantum computing could revolutionize numerous domains in business and science, with optimization frequently identified as a prime candidate to profit from such a revolution. Optimization problems arise almost everywhere, so any improvement over state-of-the-art classical algorithms with quantum computers could have a huge impact. Moreover, these improvements could occur across multiple dimensions, such as solution quality, solution diversity, time-to-solution, and cost-to-solution. But just how tangible are the benefits of *quantum optimization*? Techniques like Grover's search [1], quantum annealing, and adiabatic quantum optimization [2, 3] provided a promising start, but it has become clear that more tools are needed to realize a quantum advantage in optimization. In this work, we address the potential of quantum optimization from various angles, namely: complexity theory, problem classes and algorithmic design, execution on noisy hardware at scale, and fair benchmarking, while outlining illustrative examples derived from real-world applications. Each topic is self-contained and written for a diverse audience, from optimization researchers to industry professionals. Thus, readers may move directly to any section of interest.

For additional context, it is often stated that quantum computers can evaluate all possible combinations of an optimization problem simultaneously. This neglects that there remains an exponential list of possible solutions and that a quantum computer requires additional effort to find good ones. For some combinatorial optimization problems in a worst-case setting, the required effort of known classical approaches scales exponentially in the problem size [4]. Here, quantum computers may offer a quadratic speedup, e.g., via Grover's Search, but a quadratic speedup over an exponential run time is still an exponential run time. The situation can, however, significantly change if we consider a concrete problem instance instead of the general worst-case scenario. Classically, many algorithms and heuristics have been developed that – even for some large problems – can obtain an (almost) optimal solution in reasonable time. The most prominent example is the Traveling Salesperson Problem (TSP), where the goal is to find the shortest cycle to visit a given set of places. It is often used to illustrate the difficulty of combinatorial optimization, but in practice, very large problem instances can be solved close to optimality classically [5–10]. In contrast, there exist problems with less than 100 variables that are difficult to tackle classically [11, 12]. While quantum optimization algorithms will not necessarily improve the performance for *all* problems, they provide additional tools with new properties that can have significant impact and improve performance for *some* problem instances and thus, improve our overall capabilities in optimization. We probe these concepts more formally in Secs. II, III and IV, by laying the foundation for quantum optimization algorithms within a variety of well-studied problem classes.

Another important aspect to consider in quantum computing, is noise. The ultimate strategy to deal with noise is quantum error correction, which introduces a significant overhead in the number of qubits required [13]. Until fault-tolerant error-corrected quantum computers can be implemented at scale, other approaches to retrieve meaningful results from noisy devices have been considered. A particular strategy is quantum error mitigation [14–16], where one reduces noise by cleverly combining multiple noisy results. In Sec. V, we investigate the possibility of using error-mitigated noisy quantum computing to bridge the gap between classical and fully fault-tolerant quantum computing for optimization problems.

In addition to noise, in practice, one often relies on optimization heuristics, where the performance of an algorithm on a given problem instance is not known upfront. This makes comparing quantum and classical algorithms rather tricky. To establish systematic, reproducible, and comparable benchmarks, Sec. VI introduces fair metrics and outlines relevant optimization problems for which different algorithms can be applied to. This is crucial to identify what strategy works best and guide toward development of new quantum optimization algorithms. Sec. VII then dives into two particular real-world problems to illustrate the nuance of optimization in practice. Combined with reasonable benchmarks, these problems highlight where we stand and what open questions must be answered before we can impact a certain domain. Since quantum computing will not necessarily accelerate all problems, it is crucial to understand exactly where advantages might be found. This needs to be probed theoretically, and empirically — which is what we showcase in this work. Quantum optimization algorithms provide additional tools that complement existing ones, and synergies with classical algorithms can certainly lead to better performance.

II. QUANTUM ADVANTAGE & COMPLEXITY THEORY

In theoretical computer science, complexity classes group computational problems together in terms of the amount of resources required to solve them. Naturally, the harder the computational problem, the more resources it will require, where the resources of most importance consist of time and memory. For this reason, complexity theory is often the guiding compass for attempting to find classes of problems that are hard for a classical computer to solve, but perhaps less resource-demanding on a quantum computer. This is a very reasonable and rigorous way to formulate a quantum advantage, but *proving* that problems require strictly more resources classically than they would on a quantum computer is often a monumental task. In this section, we discuss notable strides made in complexity theory, but also highlight issues that arise when restricting our notion of quantum advantage to strict complexity-theoretic

separations in the context of optimization. The main takeaway for this section can be summarized as follows:

Understanding complexity theory is extremely useful for gauging possible quantum advantage in optimization, but rigorous complexity-theoretic separations are not necessary, nor sufficient when seeking a practical quantum advantage.

A noteworthy example to drive home this point is the task of factoring a number into its prime constituents. This problem is interesting from a complexity point of view as it is not believed to be efficiently solvable by a classical computer — although there is no formal proof that confirms this. A quantum computer, on the other hand, can solve this factoring task efficiently, thanks to Shor's algorithm [17], and is expected to meaningfully demonstrate this efficiency once fault-tolerant quantum hardware is available [18]. Thus, our notion of quantum advantage in practice should cast a wider net than formal proofs, but within a reasonable range guided by complexity-theoretic arguments. We explain this further by delving into the basics of complexity theory, discussing pertinent results and extending the discussion to more practical scenarios for optimization. We refer the interested reader to Refs. [19, 20] for a more comprehensive overview of quantum and classical complexity theory.

A. Exact Solutions

1. Decision and relational problems

Much of complexity theory deals with so-called *decision problems*. As the name suggests, a decision problem involves determining (i.e. deciding) whether there exists a particular solution to the problem or not. Without loss of generality, it is often convenient to formalize this idea using binary strings [21]. A computational decision problem on binary strings corresponds to what is called a language in complexity theory. A language $L \subseteq \{0,1\}^*$ is a set of binary strings of arbitrary length where the corresponding decision problem is to determine whether a given input string $x \in \{0,1\}^*$ is an element of L or not. For example, the language could consist of the set of all prime numbers encoded in binary form and the problem would involve determining whether x is prime ($x \in L$) or x is composite ($x \notin L$). For completeness, an alphabet Σ, can be thought of as the set of characters from which the language is formed. In this case, $\Sigma = \{0,1\}$. It is also fairly common to think of decision problems on binary strings as binary functions $f_n : \{0,1\}^n \to \{0,1\}$ where $f_n(x) = 1$ if and only if $x \in L$, which is referred to as a "yes" instance. The famous Traveling Salesperson Problem (TSP) can be formulated in such a way, where a list of cities, the distances between them, and a total distance d is described by a bitstring $x \in \{0,1\}^n$. Then $f_n(x) = 1$ if and only if there exists a salesperson's tour

through the cities whose length is $\leq d$. This problem becomes increasingly difficult as the number of cities grows, since, even though the input size $|x| = n$ grows polynomially, the number of possible tours grows exponentially; and the best known classical dynamic programming solution still needs exponential time [22, 23]. In fact, the notion of resource scaling in terms of the input size is how hardness of a problem is formulated in complexity theory.

At this juncture, it is worth noting that decision problems only care about the existence of a particular solution, and not the actual solution itself. Clearly, in practical optimization settings, finding an actual solution to a problem is often necessary. Continuing with TSP as an example, merely knowing that there exists a path to travel to all cities and back is not enough to benefit in practice, and determining the actual path to take is needed. This leads to the notion of *relational problems*, which generalize decision problems in this sense. More concretely, given a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ and an input x, the goal of a relational problem is to output any y such that $(x,y) \in R$. So one is concerned about finding a y that satisfies the relation (sometimes called a "witness" to the original decision problem), not just determining the existence of such a y. For the discussions that follow, it is useful to keep in mind that the same problem can be formulated in different ways. The decision problem already encompasses the time complexity of the problem. Namely, having a procedure that determines whether a TSP tour of length $\leq d$ exists, can be used to efficiently, modulo the complexity of the decision procedure, find a tour of minimal length $\leq d$ by iterating over the decision problem after removing edges [24]. Looking at the decision version of a problem may be relevant from a complexity point of view, but perhaps not so much from a practical one. Rather, one may explicitly interpret a relational problem as an optimization problem, making these closer to problems observed in practice.

2. *Deterministic, randomized and quantum computation*

Decision problems that are *efficiently* solvable by a deterministic machine are grouped together into the complexity class **P**, for polynomial time. Efficient implies the existence of an algorithm which solves the decision problem with a run time that grows no faster than a polynomial in the size of the input. Crucially, the degree and constants of the polynomial are important for practical purposes. For example, an algorithm with a run time scaling like n^3 is far more practical than one that runs in n^{100} time, even though both approaches scale polynomially and are therefore deemed computationally efficient in complexity theory. The set of optimization problems whose decision versions are then in **P** is referred to as **PO** [25]. Switching to relational problems leads to the class **FP** (for functional), which consists of all relations R for which a deterministic polynomial-time algorithm out-

puts a solution y, whenever one exists. In this regard, **FP** generalizes **P**, as it is the analog of **P** for functions with an n-bit output, and importantly, the set **FP** contains **PO**. While **FP** focuses on general relational problems that require finding a specific output, **PO** is specifically about optimization problems where the best solution under certain criteria needs to be found.

Similar to **P**, **PSPACE** forms the class of decision problems that can be solved with a polynomial amount of space, but there is no restriction on the time used by the algorithm. Allowing for randomized computation brings us to the class **BPP**, which stands for bounded-error probabilistic polynomial time, where a classical probabilistic machine can solve all instances of a decision problem in polynomial time, with an error probability $\leq 1/3$. Generalizing further to a computational model that uses quantum mechanics, leads to **BQP**, the class of decision problems that quantum computers can solve in polynomial time with an error probability $\leq 1/3$ on every instance. Crucially, **BQP** contains problems that are not believed to be in **BPP** or **P**, like the decision version of factoring large integers for example, and it is widely believed that **BQP** $\neq$ **BPP** [26]. Figure 1 illustrates the relationships between these classes. It is not known whether all inclusions are strict. However, there is still no proof that **PSPACE** $\neq$ **P**. In the highly unlikely situation that **PSPACE** = **P**, it follows that **BQP** = **P** and every efficient quantum algorithm can be simulated efficiently on a classical computer!

> ***Are quantum algorithms strictly more powerful than randomized algorithms?***
> The clear expectation is that the answer is affirmative, but in light of the difficulty of proving **PSPACE** $\neq$ **P**, no proof to answer this question conclusively is known. A long line of research plausibly suggests that **P** = **BPP**. The reason for this is that good pseudo-random number generators are believed to exist and these in turn can be used to de-randomize **BPP** algorithms. There is also black-box evidence for this. Bernstein and Vazirani [26] introduced the problem of Recursive Fourier Sampling (RFS) which is in **BQP** but not in **BPP** relative to an oracle. Interestingly, Yamakawa and Zhandry [27] show that, relative to a random oracle, there exist problems in **BQP** which are not in **BPP**. Arora *et al.* [28] provide further results relative to a random oracle for shallow circuits and variational quantum algorithms, but it is not clear what practical implications, if any, these separations relative to random oracles mean.

3. Hard versus complete problems

If *useful* problems for optimization, outside of **P**, could be solved efficiently by quantum computers, such a finding would lead to huge breakthroughs in computer science, akin to that of Shor's algorithm. This highlights the importance of trying to design quantum algorithms for classically hard optimization problems. To identify such problems, the class **NP** for nondeterministic polynomial-time, becomes relevant. A decision problem is in **NP** if, given an input x, we can efficiently check that x is a "yes" instance of the problem. To do this, we are given a polynomial-size proof (sometimes called a witness) which certifies this fact. Intriguingly, there is also a relational problem extension of **NP**, namely **FNP**. As mentioned, problems in **NP** are concerned with the existence of a particular solution, such as a satisfying assignment. On the other hand, **FNP** versions of these problems are concerned with finding the actual value of a solution, if one exists, in keeping with practical requirements for optimization settings. In particular, the set of optimization problems within **NP** is termed **NPO**, and **NPO** $\subset$ **FNP**.

A problem is called **NP**-complete if all other problems in **NP** can be efficiently reduced to it, i.e., with a polynomial overhead in time. Hence, **NP**-complete problems are the most difficult problems in **NP**. Notable examples are TSP and Graph Colorability [29]. More broadly, a problem is called **NP**-hard if it is at least as hard as the hardest problems in **NP** — but not necessarily in **NP** itself. Thus, **NP**-complete problems are merely a subset of **NP**-hard problems that belong to **NP**.

When thinking about problems in **NP**, it is tempting to imagine a quantum analog of this class, with complete problems of its own. There are several possible ways to do this, cf. [33] for a discussion. We hone in on one in particular — **QMA**, for quantum Merlin-Arthur. Intuitively, **QMA** encapsulates decision problems where "yes" instances can be verified efficiently and with high probability on a quantum computer, using a quantum state as a proof. This state should occupy no more than a polynomial number of qubits in the input size, however, finding such a state may of course, be computationally hard. For "no" instances all quantum proofs will be rejected (with high probability). Problems may then similarly be classified as **QMA**-hard or **QMA**-complete (see [34]).

Other classes relevant to quantum computing that are more exotic than **NP** and **QMA** exist, albeit not necessarily in the context of optimization. Due to a recent Gaussian Boson Sampling experiment [35] the class #**P** is worth mentioning. #**P** is a counting class, defined as the set of all functions f such that for an input $x \in \{0,1\}^*$, $f(x)$ counts the number of accepting computation paths of a nondeterministic polynomial-time machine on input x. The complexity of problems in #**P** relates to challenges like computing partition functions in statistical mechanics. The class **PP** may be thought of as the decision analogue of #**P** and it is known that **QMA** $\subseteq$ **PP** [36].

> ***Can quantum computers solve NP-hard problems?***   It is generally believed that **NP**-hard problems are not in **BQP**. An intuition for this belief is given by the lower bound for search in the black-box [30, 31] setting. This proves that Grover's search algorithm [1] is optimal in the black-box setting. The intuition is that this also carries over to the Boolean satisfiability problem (SAT), which is the famous standard **NP**-complete problem. It is based on the feeling that a general instance of SAT does not contain structure that can help finding a satisfying assignment just as in the black-box setting. This would imply that Grover's algorithm is also optimal for general instances of SAT. Grover's algorithm can search over all 2^n possible assignments to an n-variable formula and obtains at most a quadratic speedup over classical unstructured search procedures. Since we do not know of either a quantum or a classical algorithm that can solve all SAT instances asymptotically faster than brute-force-search, it does not seem promising that quantum computers can efficiently solve **NP**-complete problems [21, 31]. This does, however, advocate for quantum algorithms that exploit structure within certain problem instances that arise in practice, rather than attempting to try to develop fast quantum algorithms for all instances [32].

---

4. Polynomial Time Hierarchy

A useful way to extend the complexity classes like **P** and **NP** is to give those classes access to an oracle/set A. This way one can define for example $\mathbf{P}^A$, the class of languages decided by polynomial time Turing machines that have oracle access to A. This means that the machines have access to a subroutine that computes in a single step whether a query string $x \in A$. For a rigorous treatment of oracle Turing machines see for example [37]. One can also define oracle access to a whole class of problems: $\mathbf{P}^{\mathbf{NP}}$ and $\mathbf{NP}^{\mathbf{NP}}$. The kth level of the polynomial hierarchy $\mathbf{\Sigma_k^p}$ is **NP** with $k-1$ stacks of **NP** oracles. The third level for example is $\mathbf{NP}^{\mathbf{NP}^{\mathbf{NP}}}$. The polynomial hierarchy **PH** is then defined as $\bigcup_k \mathbf{\Sigma_k^p}$. It is known that $\mathbf{BPP} \subset \mathbf{NP}^{\mathbf{NP}}$ but remains an open problem whether **BQP** is in **PH**, and interestingly, there are oracles relative to whom this is not the case [38]. Another relevant computational model is presented by Chen *et al.* [39]. The authors define the complexity class **NISQ** as the set of decision problems solvable by a **BPP** machine with has access to a noisy quantum device with poly(n) qubits. They then prove relative sep-
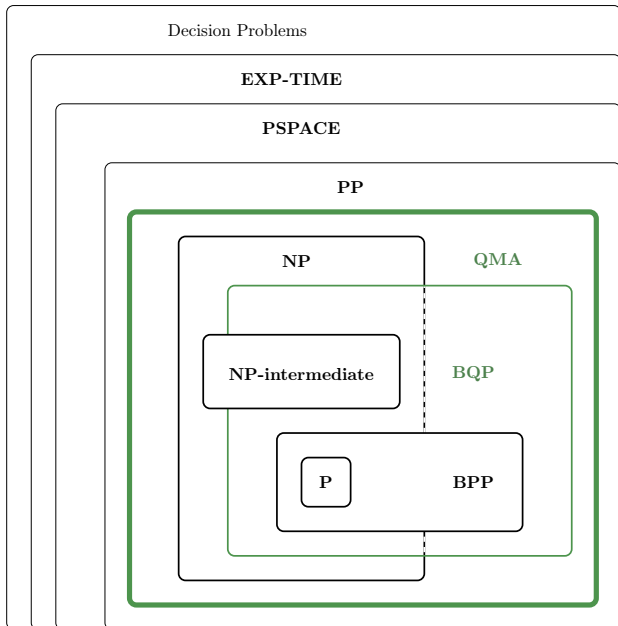
Figure 1. The inclusions of various complexity classes concerned with decision problems, as discussed in Sec. II. While these inclusions are not known to be strict, this diagram captures the gist of prominent, canonical classes in complexity theory.

arations with respect to a particular oracle O, such that $\mathbf{BPP}^{O} \subsetneq \mathbf{NISQ} \subsetneq \mathbf{BQP}^{O}$, which gives evidence that the power of **NISQ** lies strictly between **BPP** and **BQP**.

It is unlikely, that quantum computers provide exponential speedups for **NP**-hard (and thus, **NPO**-hard) problems. See [40] for stronger hypotheses along these lines. But exponential speedups for an interesting subclass of **NP** problems, not complete but also not in **P**, called **NP**-intermediate, cannot be ruled out. For example, decision versions of factoring are believed to be of this type. As depicted in Figure 1, these problems are outside of **P** and within **NP**, but are not **NP**-complete. Related classes can similarly be studied for the optimization classes **PO** and **NPO**, and relational classes **FP** and **FNP**. For instance, the relational version of TSP is $\mathbf{FP}^{\mathbf{NP}}$-complete, whereas the decision version is **NP**-complete [8]. MAX-SAT is **NPO**-complete, whereas the Vertex-Cover and Bin-Packing problems are not (even though their decision versions are **NP**-complete) [41].

5. *Worst-case complexity and fixed-parameter tractability*

It is important to emphasize the following point:

Complexity theory is traditionally focused on worst-case problem instances. In practice, one often needs to solve a "typical" instance which may be much easier than the hardest, worst-case instance.

This implies that, even when worst-case complexity-theoretic results rule out quantum advantage for **NP**-hard problems, quantum computers may still exhibit exponential speedups on instances relevant for practical problems. More concretely, they may give efficient algorithms for instances that take classical computers exponential time. We discuss this in more detail in Sec. II C.

On a similar note, while many problems of practical interest are **NP**-hard, if we are interested in measuring the computational complexity of problem which admits an algorithm $\mathcal{A}$, as a function of the input length n in conjunction with some adjustable parameter ϵ, then interestingly, parameterization could render otherwise **NP**-hard instances efficiently solvable — although the dependence could be exponential with respect to $1/\epsilon$. Such a problem is termed fixed-parameter tractable (**FPT**) and the complexity is sometimes referred to as *parameterized complexity* [42, 43]. Therefore, **FPT** amounts to the parameterized analogue of **P**. A common example is that of the Vertex-Cover problem, which is **NP**-complete when considering the input size in isolation. However, there exists an $\mathcal{O}(n)$ algorithm when the problem is restricted to constant-size covers [44]. Bremner *et al.* [45] introduced quantum parameterized complexity. In particular, the class **FPQT** is introduced as the class of parameterized decision problems tractable by a quantum computer, but intractable otherwise. It is well known that if a parameterized problem is in **FPT**, it admits kernelization, which could be seen as a pre-processing algorithm that reduces the size of an instance of the problem, often substantially [46, Theorem 1.4]. For benchmarking purposes in optimization, cf. Sec. VI, the notion of kernelization is of crucial importance. Formally, a kernelization (also known as kernelization algorithm) for a parameterized problem can be interpreted as follows: assume a language L over an alphabet Σ and parameters in $\mathbb{N}$, an algorithm given $(x, k) \in \Sigma^* \times \mathbb{N}$ outputs a string $x' \in \Sigma^*$ and an integer parameter $k' < k$, such that $(x', k') \in L$ — if the algorithm runs in polynomial time and k can be expressed as a polynomial of k', the tuple (x', k') is termed a polynomial kernel. Such polynomial kernels can be obtained for many combinatorial optimization problems considered in theoretical computer science [47]. For **FPT** problems, it thus makes sense to benchmark on the kernel.

B. Approximate Solutions

Thankfully, the story does not end at obtaining *exact* solutions efficiently, with high probability — if it did, we would probably have a difficult time motivating classical computing, let alone quantum, as a useful paradigm in (combinatorial) optimization. In reality, finding a solution that is in some sense, close to the optimal is often sufficient. The shift from finding the optimal solution, to an approximate one, leads to the study of approximability — in particular, how hard is it to approximate solutions
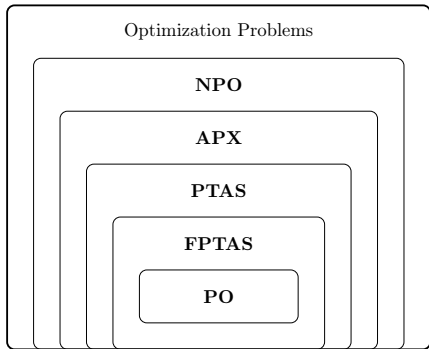
Figure 2. Complexity classes for optimization problems. Notice that, similar to #**P**, the functional problem classes are outside of the decision problems, and hence, not displayed in this hierarchy.

to various problems? This leads to interesting complexity classes for optimization, induced by approximability. Intuitively, it may seem like allowing for approximate solutions could reduce the complexity of a problem, similar to those in **FPT**. Unfortunately, it is not that straightforward when dealing with approximation and it seems that in many cases, approximating solutions to hard problems is just as hard as finding the optimal ones. We discuss this so-called hardness of approximation next, followed by classes of problems that can exhibit efficient approximation.

1.   Hardness of approximation

In the context of approximate solutions for optimization problems, a longstanding result called the **PCP**-theorem informally states that, for many problems, computing an approximate solution with arbitrary precision is as hard as computing the exact solution. To discuss this a bit more formally, the approximation ratio, which — as the name suggests — is the ratio between the (expected) approximate solution and the optimal, quantifies the quality of a solution provided by an approximation algorithm. For many **NP**-hard problems, there are known bounds on the achievable approximation ratios, and these are referred to as "inapproximability bounds". Such bounds can be derived through many techniques, including information-theoretic arguments (e.g., any algorithm that achieves an approximation factor better than c must make at least exponential queries to a function), or reductions to known **NP**-hard problems (e.g., if a problem can be approximated better than its inapproximability bound, then it is equivalent to solving another **NP**-hard problem). The former holds independent of whether **P** is equal to **NP**, but the latter inapproximability bounds are said to be conditional on $\mathbf{P} \neq \mathbf{NP}$.

For the MAX-E3-SAT problem, one of the earliest approximation algorithms showed that a random variable

assignment satisfies each clause with probability 7/8, and thus, achieves a 7/8-approximation ratio [48]. It was later shown that there is no classical polynomial-time algorithm that can achieve a ratio of $(7/8+\epsilon)$, for any $\epsilon > 0$ unless **P**=**NP** [49]. In such cases, where inapproximability bounds are achieved by classical algorithms, improving these ratios with quantum computers is highly unlikely, as this would imply that quantum computers can solve **NP**-hard problems.

Some inapproximability bounds require additional conditions. For example, in their seminal work in 1995, Goemans and Williamson introduced an approximation algorithm for the MAXCUT problem and proved that it achieves an approximation ratio of $c \approx 0.87856$ [50]. Later, it was shown by Khot *et al.* that there cannot exist a polynomial time algorithm that achieves $(c + \epsilon)$-approximation [51] under the assumption of the Unique Games Conjecture [52]. Again, in such cases, improving these ratios with quantum computers would imply that quantum computers can solve **NP**-hard problems *or* that the Unique Games Conjecture is false. Similar results exist for other **NP**-hard problems, such as graph coloring problem and set covering [53].

There is also a quantum **PCP** conjecture proposed by Aharonov *et al.* [54–56]. At its core is the k-local Hamiltonian problem: given a Hamiltonian H with k-local terms (i.e. operations acting on up to k qubits), determine the smallest eigenvalue of H. This problem is **QMA**-complete [57], akin to SAT's **NP**-completeness. The local Hamiltonian problem comes with a promise denoted by γ. The ground state energy of H is either less than some value a, or greater than $a + b$, where $\Gamma := b$ is the absolute promise and the ratio $\gamma := b/m$, where m is the number of k-local terms, is the relative promise. The quantum **PCP** theorem then states that the k-local Hamiltonian problem is **QMA**-hard even with a constant relative promise gap $\gamma > 0$, if the interaction graph is sparse and all k-qubit interactions have bounded norm. There is some supporting evidence for the quantum **PCP** theorem, perhaps most notably the recently proved No Low-Energy Trivial State (NLTS) theorem [58]. Under NLTS, there exist families of Hamiltonians for which all low energy states require circuits of super-constant depth to prepare. If NLTS were false, this would imply that all Hamiltonians have low energy states that are efficiently simulable — effectively rendering the quantum **PCP** theorem invalid since approximating the ground state energy could be done efficiently using these low energy states. While the quantum **PCP** theorem implies NLTS, under the assumption that **QMA** is not equal to **NP**, the other direction does not hold, since there could be other types of classically simulable circuits that could prepare low energy states, other than low depth circuits. This remains a very open and interesting research area.

Indeed, the hardness of approximation seems to imply – theoretically – that quantum computers will not be very influential, even if approximate solutions are the goal. However, this is not necessarily true in general,

and less so in practice. There is hope for quantum algorithms to outperform classical methods when there is a gap between known inapproximability bounds and provable approximation factors (e.g., in metric TSP, [59, 60]). Further, quantum optimization methods might provide provable and computational speedups, even if they simply match the approximation factors of classical algorithms.

2.   Polynomial-time approximation schemes

If a problem is "approximable", it is said to belong to the class **APX** which contains the set of **NPO** problems for which there are polynomial-time approximation algorithms with approximation ratios bounded above by some constant c [61]. An interesting subclass of **NPO** problems that admit a so-called polynomial-time approximation scheme is **PTAS** [25] [62, Chapter 8]. For any problem in **PTAS**, there is a polynomial-time algorithm that is guaranteed to find a solution whose value is within a $1 + \epsilon$ factor of the optimum, where $\epsilon > 0$. For example, by restricting to the Euclidean plane, TSP is then in **PTAS** [63]. However, as with similar issues in **FPT**, there are cases where the exponent of the polynomial might depend exponentially on $1/\epsilon$. To account for this, one may consider the class **FPTAS** for fully polynomial-time approximation scheme, which demands the run time to be polynomial in the input size, as well as in $1/\epsilon$. As such, **FPTAS** $\subseteq$ **PTAS**, and the hierarchy of these classes are displayed in Figure 2. When combined with quantum computing, this area makes for an exciting and rather unexplored research direction — where quantum algorithms could serve as new approaches or improve over existing classical **PTAS** algorithms.

C.   Heuristics in Quantum Optimization

Thus far, the focus has been on worst-case instances. In practice, heuristic approaches can provide useful solutions to special, relevant and typical problem instances [62, e.g. Chapter 28]. Heuristics have also proven useful in problems exhibiting structure and average-case instances — which are oftentimes far more practical in real-world scenarios [62, 64]. The success of classical heuristic algorithms indeed motivates the study of quantum heuristic methods, where the aim of the latter is to design algorithms that leverage quantum computation to provide reasonable solutions to problems where classical approaches either cannot, or simply take too long to do so. Naturally, the word *reasonable* depends very much on the context, making heuristic approaches difficult to analyze theoretically. More concretely, heuristic algorithms are understood as algorithms without provable performance guarantees, run time guarantees, or both (as in the case of so-called metaheuristic algorithms). By this negative definition, our understanding of heuristic algorithms

is rather limited. Having said that, there are multiple attempts to provide more formal definitions. For instance Bogdanov *et al.* [65] define distributional problems, which are pairs (L, D) of a language L and a family of distributions $D = \{D_n\}$, one for each input size n. Then, a distributional problem (L, D) is in the class **HeurBPP** (for heuristic **BPP**) if there exists a polynomial-time randomized classical algorithm $\mathcal{A}$ such that for all n and $\delta > 0$, $\Pr_{x \sim D_n} \left[\Pr \left[\mathcal{A}(x, 0^{1/\delta}) = L(x) \right] \geq 2/3 \right] \geq 1 - \delta$ where the inner probability is taken over the internal randomization of $\mathcal{A}$. In other words, algorithms that solve problems in **HeurBPP** are allowed to err on a small fraction of instances drawn from D_n. Analogously, we say that a distributional problem (L, D) is in **HeurBQP** if there exists a polynomial-time quantum algorithm $\mathcal{A}$ that satisfies the same property. This "distributional" view of heuristics has been rather popular in relation to quantum advantage.

Notably, Pirnay *et al.* [66] prove a separation for a combinatorial optimization problem under very special distributions in the sense above, namely those which are Karp-reduction-images of factoring problems as pointed out by Szegedy [67]. The former proves that quantum computers can have a super-polynomial advantage over a certain class of NPO-problems and that this also holds in the approximation sense. It does so by making use of cryptographic tools and by means of notions of computational learning theory, in particular the Occam learning framework. The latter points out that similar separations hold for other NPO approximation problems, by resorting to the PCP theorem. The approach is via faithful reductions from well-established problems in **NP**, which have a large quantum advantage, namely factoring, to matching instances of **NPO** problems. Using techniques from complexity theory, it is possible to construct instances with a large approximation gap, which ensure that classical algorithms cannot approximate the correct solutions to these instances, unless factoring is classically easy. These findings, however, are unlikely to have much practical value as the proposed quantum solutions will only work on the contrived instances corresponding to factoring, and are not general optimization solvers which could solve other classes of instances efficiently as well. The results do, at least, contribute to understanding what kind of quantum advantages one can hope for in combinatorial optimization.

Independently, one could relate the distributional view of heuristics to numerous studies of so-called landscapes [68]. The term landscape here aims to generalize an intuitive mental picture where one associates the height of terrain to solution quality and traversing this terrain corresponds to moving through a space of solutions (which need not be two dimensional in general). For example, Refs. [69–71] study the landscape associated with variational quantum algorithms for specific problems, with the aim of explaining when these heuristic methods converge. An important feature in the study of landscapes are so-called *barren plateaus* [72–79],

especially in conjunction with variational quantum algorithms, which correspond to loss landscapes that are exponentially flat in the number of system qubits. In fact, this phenomenon provides a central bottleneck in the scaling of variational quantum algorithms to practically relevant problem sizes. There are multiple reasons for the appearance of barren plateaus. It was conjectured that they are induced by an ansatz that is (close to) a t-design [72, 75] or exhibits too much entanglement paired with partial measurements [76]. These findings, along with other causes of barren plateaus were united in theorems which either relate the occurrence to a mathematical object called the dynamical Lie algebra, which is induced by the structure of the generators and the form of the variational circuit [70, 71], or to the average behavior of the respectively induced light-cone [80]. Notably, these phenomena may be avoided if a good warm-starting strategy is known–in the sense of a parameterization that brings the system sufficiently close to the optimum. However, it is often unclear how such a warm-start should be chosen.

There have also been a number of works suggesting that quantum sampling could be a useful subroutine in a larger classical algorithm. For example, there are quantum circuits for Metropolis-Hastings algorithms [88] and quantum enhancement of Markov-chain Monte-Carlo (MCMC) algorithms [89, 90]. These approaches should be possible to analyze, and thus, turn into randomized approximation schemes.

D.   Takeaways from Complexity Theory in Quantum Optimization

Although it is widely believed that quantum computers will not offer exponential speedups for **NP**-hard problems, the story does not end there. As we have seen, most complexity-theoretic statements deal with worst-case settings for exact solutions. In practice, typical optimization problem instances could differ substantially from the worst-case, and thus, exponential quantum speedups in these more realistic settings are still possible in principle. Importantly, the picture may drastically change when dealing with average-case hardness. Average-case settings could exclude particularly bad instances for which most complexity-theoretic statements hold, and thus, an advantage – certainly one faster than simply the Grover approach – becomes possible and it is an interesting open problem whether quantum algorithms can solve optimization problems that are average-case hard for classical algorithms. For example, there is evidence from [91] that quantum algorithms can solve random k-SAT faster than Grover and the state-of-the-art SAT solvers. Additionally, problems with a so-called Overlap Gap Property (OGP) [92] have been proven to be average-case hard for certain families of classical algorithms (and local low-depth quantum algorithms), but it remains unclear whether these problems can be addressed by higher-depth and/or more sophisticated quantum algorithms [93–96]. If so, this would be very convincing evidence for useful superpolynomial quantum advantage, since such problems naturally arise in combinatorial optimization, statistical physics and high-dimensional statistics. Beyond OGP in the average-case setting, there is an emerging literature on the "computational-statistical gaps" of many optimization and learning problems where we could also hope for a superpolynomial quantum advantage. Moving into the approximation realm also introduces interesting open questions for quantum computing. For example, can quantum algorithms saturate inapproximability bounds that have not yet been reached by classical methods? Moreover, if approximation ratios cannot be improved, there may still be room to speedup existing approaches with quantum computing.

Another intriguing aspect of complexity theory, is that a restriction of some parameters of a problem, or some structure present, may drastically change the complexity. A perfect example of this is TSP, which – in its most general decision version form – is **NP**-hard and the optimization version is **APX**-hard. If we restrict to Euclidean TSP, where all points are in $\mathbb{R}^d$ and edge weights equal their Euclidean distances, then, remarkably, there exists an explicit **PTAS** to solve such TSP instances [63]. Shifting to heuristic algorithms for TSP, one can actually solve problems in practice with up to millions of variables. Lastly, if we were to consider the shortest path problem instead of TSP, where we just want the shortest path between two places, we would have a problem in **P**. This illustrates the nuance of complexity theory that one should keep in mind when designing quantum optimization algorithms. Akin to TSP classically, if provable quantum speedups seem unattainable, quantum heuristic algorithms may still offer meaningful results. But developing quantum heuristic methods that exploit structure in a problem, or enhance classical approaches, is not straightforward either. This serves as the basis for the next section, where we lay out various building blocks to carefully design quantum optimization algorithms.

III.   PARADIGMS IN THE DESIGN OF QUANTUM OPTIMIZATION ALGORITHMS

There are multiple approaches that lead to exact quantum optimization algorithms, such as Grover (Adaptive) Search [97–99] and the Quantum Adiabatic Algorithm (QAA). There are lesser known approaches too, like Quantum Imaginary Time Evolution (QITE) [100], but much of the published work focuses on the more well-known approaches. In this section, we introduce the core concepts deployed as subroutines in various quantum optimization algorithms discussed in Sec. IV.

Table I. An overview of the classical problem classes relevant to optimization problems. The models of analogue computation below the horizontal line are not considered in much detail in this section.

Complexity class	Example problem	Key features	Year	Ref.
FPTAS	Knapsack, Subset-Sum	Efficiently approximable		[81]
PTAS	Euclidean TSP	Approximable in poly-time		[63]
APX-Complete	MAXCUT, MAX-3-SAT, Metric-TSP	Discrete decisions	1991	[25, 81]
NPO-Complete	TSP, Binary IP	Discrete decisions	1988	[82]
#**P**	Two-stage SP [83, Theorem 3.1]	Stochastic objective function	2006	[83, 84]
PSPACE	Multi-stage SP [83, Theorem 4.1]	Stochastic objective function	1985	[83, 85]
Banach-Mazur computability	Optimal Transport [86, Theorem 5]	Real-valued decisions	2003	[86]
NP over Reals	Optimal Transport [86, Theorem 5]	Real-valued decisions	1996	[87]

A.   Grover Search

The seminal Grover algorithm for searching an unstructured database achieves a quadratic advantage over classical search in terms of the number of queries to the database (i.e. the query complexity). In exact quantum optimization, this advantage translates to an advantage for function minimization [97] and can achieve a quadratic speedup over brute force search in many problems in combinatorial optimization [101], i.e., a run time of $\mathcal{O}(\sqrt{2^n})$ instead of $\mathcal{O}(2^n)$ for n-variable problems. In the most general case, without any further assumptions on the problem, for example in terms of oracle used in the query complexity, this is the best run time and advantage that we can hope for when using a quantum computer for discrete optimization [102]. For a concrete problem class, however, brute force search is rarely the best available classical algorithm, and thus, the actual advantage achieved by Grover Search is usually sub-quadratic [103]. Nevertheless, leveraging Grover Search as a sub-routine, together with other (quantum or classical) algorithms, can sometimes help to recover an overall quadratic speedup over the best available exact classical algorithm [104].

B.   Quantum Adiabatic Algorithm

The Quantum Adiabatic Algorithm (QAA) [2, 3] is another exact optimization algorithm, reminiscent of homotopy methods of numerical mathematics [105]. It assumes a problem Hamiltonian H whose ground state corresponds to the solution of a corresponding optimization problem, as well as a so-called mixing Hamiltonian H_X. The mixing Hamiltonian should have an easy-to-prepare ground state $|\psi_0\rangle$ that has a non-zero overlap with the ground state of H. A common choice is $H_X = \sum_{i=1}^n \sigma_X^{(i)}$, where $\sigma_X^{(i)}$ denotes the Pauli X matrix on qubit i, with the ground state $|+\rangle^{\otimes n}$, i.e., the n-qubit equal superposition state that has non-zero overlap with every computational basis state. Starting in $|\psi_0\rangle$, QAA slowly drives the state according to $H(t) = \lambda(t/T)H + (1 - \lambda(t/T))H_X$, and $d/dt\,|\psi_t\rangle = -iH(t)\,|\psi_t\rangle$, for some total annealing time $T > 0$ and an annealing schedule $\lambda : [0, 1] \to [0, 1]$

with $\lambda(0) = 0$ and $\lambda(1) = 1$. If the annealing is performed slowly enough, in other words, if T is large enough, with a suitable annealing schedule, the Adiabatic Theorem [106–109] guarantees that the state $|\psi_t\rangle$ always remains in the ground state of $H(t)$. Thus, $|\psi_T\rangle$ represents the ground state of H, which corresponds to the solution of the encoded problem. It has been proven that the annealing time has to scale inversely with the minimal squared spectral gap Δ of $H(t)$ over all $t \in [0, T]$, i.e., $T = \mathcal{O}(1/\Delta^2)$. During annealing, the system undergoes quantum phase transitions, and in the glassy phase, at low values of t/T, the energy gap closes exponentially with increasing system size [110–114]. Consequently, this implies exponentially long run times if one aims to precisely follow the adiabatic path. Hence, we are limited to at most polynomial speedups.

The practical implementation of the quantum annealing protocol often involves repeated finite-time runs [115] or, potentially, diabatic annealing [116]. This variant relaxes the constraint that the system must always remain in the instantaneous ground state of $H(t)$. Another variant is counter-diabatic annealing, where additional terms are included in the Hamiltonian to suppress the transition to excited states during the annealing [117, 118]. For example, the counter-diabatic terms can be built up from nested commutators [119], while the evolution can be engineered with optimal control [120]. In its practical implementation, QAA often becomes a heuristic algorithm. On a gate-based quantum computer, the time evolution to generate $|\psi_t\rangle$ can be implemented, e.g., through Trotterization [121–123]. The typical long annealing times then translate to long quantum circuits.

C.   Quantum Phase Estimation

Quantum Phase Estimation (QPE) is an algorithm to estimate the eigenphases of unitary operators [124]. It allows the estimation of the eigenvalues of certain matrices and can be used to find the ground state and corresponding energy of various Hamiltonians [125–127]. Famously, QPE is also leveraged by Shor's algorithm. Factoring is believed to be in **NP**-intermediate [41] and Shor's algorithm [128] achieves an exponential speedup over classical algorithms, as discussed in Sec. II. Like many other prob-

lems, factoring can be cast into a quadratic unconstrained binary optimization problem (QUBO) [129–131]. Thus, this gives us an exponential speedup over a subset of QUBOs that happen to correspond to instances of factoring. Similar reductions are possible from other classes of problems to factoring [67, 132]. In practice, however, we do not expect these instances to appear naturally, and it is an open question how to harness this relation in the context of optimization.

In general, QPE requires an initial state having a polynomial overlap with the ground state, i.e. an overlap greater than $1/\text{poly}(n)$. For a diagonal Hamiltonian, every computational basis state is an eigenstate, and so QPE does not increase the chance of sampling the ground state. Further, simply sampling from such an initial state provides a polynomial-time algorithm to find the ground state already. Thus, the true problem is finding such an initial state with a polynomial overlap. Nonetheless, some algorithms encode an optimization problem into a non-diagonal Hamiltonian, potentially making QPE useful in quantum optimization. The situation is similar for (Krylov) subspace expansions [133, 134], where an initial polynomial overlap is assumed to derive convergence properties towards the ground state of a Hamiltonian.

To make QPE and related techniques relevant for a large set of applications, it is important to have a quantum toolbox of matrix linear algebra operations [135–140], in analogy to the classical basic linear algebra subprograms (BLAS) [141]. Quantum signal processing [142] enables the simulation of sparse matrices, i.e., the implementation of the time evolution operator e^{-iHt} for a matrix H and a time t, with optimal complexity. Phase estimation then allows one to extract eigenvalues and eigenvectors of H. Quantum singular value transformation [143] generalizes this result and allows efficient simulation of matrix algebra such as $A + B$, AB, and singular value transformations $P(A)$, with P being a low-degree polynomial applied to the singular values of A, where A and B are matrices. The block-encoding framework unifies input models for matrices given efficient descriptions, sparse access, or quantum memory. These tools find applications in preparing Gibbs states, matrix problems, and continuous optimization.

D. Gibbs Sampling

Gibbs sampling corresponds to generating sample sequences from the joint probability distribution of multiple variables underlying a Gibbs distribution. Geman and Geman [144] introduced a Markov Chain Monte Carlo (MCMC) algorithm to facilitate Gibbs sampling. Its significance in optimization, particularly in classical MCMC algorithms, extends to solving problems like constraint satisfaction [145].

Sampling from a classical many-body Hamiltonians' H Gibbs distributions, and the related problem of approximating the partition function $\mathcal{Z}(\beta) = \text{Tr}\left(e^{-\beta H}\right)$ or the

(Gibbs) free energy of the system, $\log \mathcal{Z}$, falls within the complexity class **BPP$^{\text{NP}}$** [146, 147]. Here, β is the inverse temperature. Gibbs sampling, then, can be done with the Metropolis-Hastings algorithm [148, 149] which, despite its exponential scaling with particle number, often exhibits convergence.

In contrast, quantum Gibbs sampling [150] presents greater challenges and is believed to be at least **QMA**-hard [151, 152]. The quantum Gibbs state $\boldsymbol{\sigma}_\beta$ is (up to normalization) defined as

$$\boldsymbol{\sigma}_\beta \propto e^{-\beta H} = \sum_{i=1}^{\dim H} e^{-\beta E_i} |\psi_i\rangle \langle \psi_i|, \qquad (1)$$

where H is the Hamiltonian operator, $|\psi_i\rangle$ are the eigenstates, as before β is the inverse temperature, and E_i are the energy eigenvalues.

The problem of thermalizing a quantum state has a long tradition and was studied earlier in [153]. Quantumly, the difficulty does not only relate to the hardness of sampling but also to the difficulty to efficiently prepare Gibbs states. Quantum adaptations of the Metropolis-Hastings algorithm that involve quantum phase estimation (QPE) are challenging due to the algorithmic complexity[154, 155]. Furthermore, Gibbs sampling algorithms for Gibbs states of local Hamiltonians at all temperatures have been devised in [156]. It has also been seen that high-temperature Gibbs states are unentangled and efficiently preparable [157].

A variational method for quantum simulators to generate finite temperature Gibbs states through the preparation of thermofield double states was introduced in [158]. A quantum algorithm to prepare the quantum Gibbs state, as well as the partition function within ϵ-error, was proposed in [150]. A different VQA approach for the preparation of Gibbs states suitable for NISQ devices was presented in [159] where the authors presented an algorithm in their paper by applying a truncated Taylor series to compute the free energy, and then selecting this truncated value as the loss function. Another VQA was proposed in [160] where the variational parameters are chosen by minimizing the free energy. Warren *et al.* [161] introduced an adaptive VQA approach to prepare Gibbs states by introducing an objective function that is easier to measure than the free-energy and using dynamically generated, problem-tailored ansätze. Finally, a variational method for approximate Gibbs preparation via imaginary time simulation has been presented in [162]. The difficulty in these variational approaches either lies in the requirement to continuously compute the von Neumann entropy or to correctly propagate imaginary time dynamics via a variational ansatz. If sampling from the Gibbs state is sufficient, approaches like "Quantum Minimally Entangled Typical Thermal States" [100] can be used, which are based on repeated imaginary time evolution of a weakly entangled initial state. Wild *et al.* [163] introduced a set of quantum algorithms that

provide unbiased samples by creating a state that encodes the quantum Gibbs distribution in its entirety and demonstrate that this method can outperform classical Markov chain algorithms in various scenarios, including the Ising model and sampling from weighted independent sets on two distinct graphs. Additionally, recent advancements in quantum Gibbs sampling, like the Dissipative Quantum Gibbs Sampling algorithm [147], offer a simpler and potentially more efficient alternative, as they utilize local update rules and are more feasible for near-term quantum hardware.

Quantum Gibbs sampling plays a crucial role in various optimization algorithms, particularly in convex optimization as discussed in Sec. IV B 1. Quantum speedups are possible because, under certain conditions, Gibbs states can be prepared more efficiently quantumly than classically. The block-encoding framework [143] demonstrates this, where an $n \times n$ Gibbs state is constructed with complexity $\mathcal{O}(\sqrt{n})$ in the dimension parameter, albeit other parameters also appear in the running time. This method and its applications in algorithms like the Matrix Multiplicative Weights Update for solving general semidefinite optimization problems are further elaborated in [164–167]. Notably, these algorithms leverage Gibbs states for computing trace inner products.

Furthermore, the complexity analysis of quantum partition functions, as explored in [168], reinforces the importance of efficient Gibbs sampling methods in quantum computing. Encoding Gibbs distributions is at the heart of simulated annealing approaches for Gibbs partition functions [169–172], which can be used to approximately count certain combinatorial objects such as k-colorings or matchings of a graph, with genuine quantum speedups in theory. Gibbs sampling methods find application in specialized solutions to semidefinite relaxations of problems like MAXCUT [173, 174], see Sec. IV B 1, and are integral to quantum algorithms for zero-sum games [175–177]. Finally, [178] introduced the notion of a Gibbs objective function which can be useful for quantum optimization problems. The objective can be evaluated by the means of Gibbs sampling, although it can be challenging, as discussed previously. We further mention this approach in Sec. IV A 1.

E.   Approximation Algorithms

Grover Search, (Trotterized) QAA, QITE, (and QPE) are all expected to require Fault-Tolerant Quantum Computing (FTQC) due to the resulting circuit sizes. However, they also serve as theoretical motivation for approximation algorithms and heuristics, which may already lead to quantum advantages in optimization with noisy quantum computers, as discussed later in this section.

In addition to the exact algorithms discussed in Sec. II, there exist also approximation algorithms that guarantee a certain approximation ratio. Quantum Computing could be used to accelerate known classical approxima-

tion algorithms via Quantum subroutines. For instance, Semidefinite Programming (SDP) relaxations, as used in the famous Goemans-Williamson algorithm for MAX-CUT [179, 180] or more general variants for QUBO [181], may profit from Quantum SDP solvers, cf. Sec. IV B 1. Eventually, for special cases, these may even achieve an exponential speedup for solving some SDPs compared to classical algorithms. However, unless the problem is given in a compact functional form and assuming we are not interested in reading out the full solution but only the optimal objective value, the end-to-end advantage will be reduced to polynomial due to input/output bottlenecks. It also raises the question whether classical algorithms may be able to leverage this special setting to achieve similar improvements, as has been shown, e.g., in the context of recommendation systems [182] or solving low-rank linear systems [183]. Further, there are many practical problems to achieve such an advantage for solving SDPs, as will be discussed in more depth in Sec. IV B 1, and very likely this will require FTQC.

F.   Variational Methods

Within the near-term intermediate-scale quantum (NISQ) devices, much work is focused on so-called Variational Quantum Algorithms (VQAs) [184–188], i.e., a family of randomized search algorithms that solve certain optimization problems classically while evaluating involved expectations quantumly. Specifically, a VQA is an algorithmic scheme where one provides some problem data $\mathcal{D}$, a parameterized ansatz $|\Psi(\vartheta)\rangle$ and a corresponding cost function $C(\vartheta)$ as input. Then, the parameterized quantum state is measured in some basis $\{O_k\}_{k\in\mathbb{N}} \in \mathrm{Herm}(\mathbb{C}^n)$ to evaluate the cost function. The next step involves optimizing over the cost function in order to obtain updated parameters, which are then fed back onto the circuit. See Fig. 3 for a schematic.

Although finding the optimal parameters in a VQA is in general **NP**-hard [189], it can still result in a powerful heuristic to find good solutions, even potentially offering speedups [190]. This situation is similar to the training of classical artificial neural networks, whose training to the optimum is also **NP**-hard [191, 192] (even with only $k = 2$ layers, a ReLU neural network cannot be trained in time bounded by a polynomial in the dimension of the weights [193]). VQAs differ mainly in the choice of the ansatz, the cost function, and the optimizer, as discussed, for example, in Sec. IV A.

Although there are analyses of asymptotic convergence relating variational methods to the QAA [194], bounds on iteration complexity of variational methods [195, 196], and in some cases, bounds the objective function value at the limit point [197, 198], VQAs are still mostly seen as heuristics.
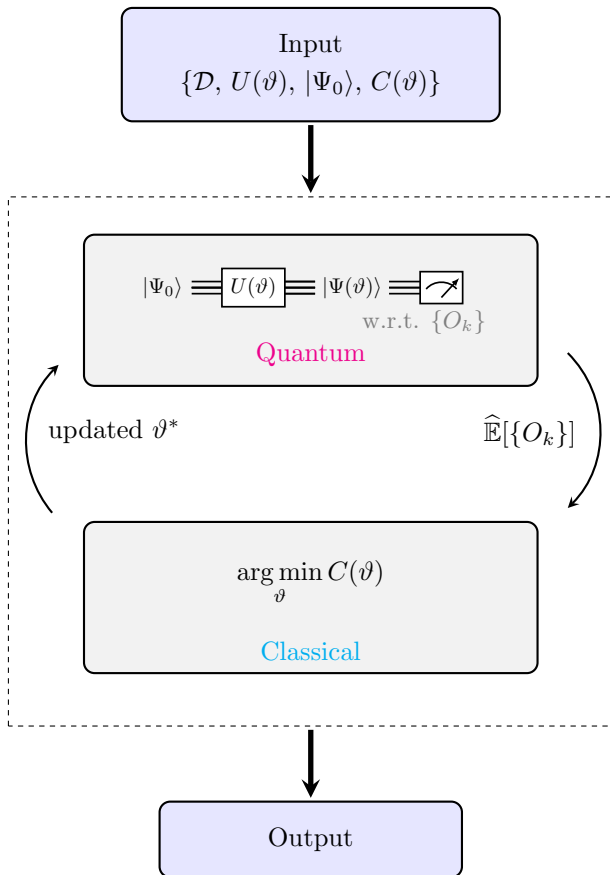
Figure 3. Variational methods iterate between the evaluation of empirical estimates of the expectations of a set of observables $\{O_k\}$ associated with a parameterized quantum circuit, and classical computation of updated parameters for the parameterized quantum circuit.

IV.   PROBLEM CLASSES & ALGORITHMS

There are many different classes of optimization problems that are defined through the types of variables involved, the objective function(s) used, presence of constraints, stochasticity, and so on. Furthermore, problems can be modeled in different ways, which can lead to representations of the same problem in different problem classes. Although these formulations might be formally equivalent, the chosen formulation can play a crucial role in how a problem is solved, as different formulations put more or less emphasis on certain properties of a problem. In this section, we introduce the most important classes of optimization problems and discuss existing algorithms to solve them. We then discuss key open questions in each domain that could help advance the field of optimization.

A.   Discrete Optimization

Discrete optimization is a branch of mathematical optimization that focuses on problems where, as the name suggests, variables take on discrete values. This often implies a combinatorial explosion of possible solutions with respect to size of the input parameters. We focus on discrete optimization problems with and without constraints. While problems of one class can be converted into problems of the other class, approaches to solve them can differ significantly and leveraging available structure might be beneficial in many cases. Moreover, the equivalence often only holds if a globally optimal solution is found. The remainder of this section discusses exact algorithms, approximations, and heuristics for the considered problem classes within discrete optimization.

1.   Unconstrained Discrete Optimization

In the realm of unconstrained discrete optimization, Quadratic Unconstrained Binary Optimization (QUBO) serves as a foundational example and is given as follows:

$$\min_{x \in \{0,1\}^n} x^T Q x, \tag{2}$$

where $Q \in \mathbb{R}^{n \times n}$ is the cost matrix and $n \in \mathbb{N}$ denotes the number of variables. Other types of unconstrained discrete optimization problems involve, for instance, high-order polynomial objective functions [199], also referred to as Polynomial Unconstrained Binary Optimization (PUBO), or black box objective functions [200] as well as (bounded) integer variables. While one could also define linear unconstrained discrete optimization, this class is of little interest as it can be solved by looking at every variable independently. Strictly speaking, QUBO also allows one to model linear objective functions when Q is a diagonal matrix, because binary variables satisfy $x^2 = x$.

Some problems, such as MAXCUT, are naturally formulated as a QUBO, and most unconstrained discrete optimization problems can be mapped to QUBOs [201–204], for example, by replacing integer variables by a suitable encoding [205, 206] or by converting higher-order polynomial terms to additional binary variables and quadratic terms [207, 208]. In addition, certain constraints can be modeled by translating them into penalty terms and adding them to the objective function [201, 203]. However, these conversions usually assume that the problems are solved exactly, i.e., that the global optimum is found. In the case of approximate solutions, this has to be carefully analyzed. For instance, equality constraints might only be satisfied approximately when being represented via a penalty term, which, depending on the application, might be acceptable or not. Furthermore, binary encodings of integers usually do not reflect the intrinsic order between values, i.e., they lack some important structure that a solver may leverage to find good solutions.

Within this section, we focus primarily on QUBO. QUBOs are a natural problem class to consider for quantum optimization as they can easily be converted into a ground state problem, often explored in quantum computing, usually in the context of quantum chemistry or quantum physics [184]. The relationship between solving a QUBO and finding the ground state of a Hamiltonian is fundamental to many quantum optimization algorithms. The most common translation involves two steps: first, substituting binary variables $x \in \{0, 1\}$ with spin variables $z = 1 - 2x$, where $z \in \{-1, +1\}$, and secondly, replacing these spin variables with Pauli Z matrices. The result is a diagonal Hamiltonian $H \in \mathbb{R}^{2^n \times 2^n}$ that encodes the objective values of Eq. (2) on its diagonal. The ground state of this Hamiltonian represents the QUBO's optimal solution [201].

QUBOs with spin variables $z \in \{-1, +1\}$ naturally arise in physics and are referred to as *Ising spin glasses* [209]. Here, the present quadratic terms are usually given via a lattice or graph, and their weight corresponds to an interaction strength between neighboring sites. By allowing a polynomial resource overhead, many graph covering and coloring problems can be mapped to an Ising spin glass on a regular lattice [201]. However, to save qubit resources, many benchmarks use simpler Ising Hamiltonians on hardware-efficient lattices without a specific problem embedding [115].

In general, QUBO problems are **NP**-hard [209]. In principle, one could consider, for instance, Grover search or quantum annealing, as suggested in Sec. III, to solve QUBOs; however, these algorithms with provable performance guarantees are expected to require FTQC. QUBO problems are also **APX**-hard, i.e. there cannot exist a PTAS for QUBO problems unless **P**=**NP** [210]. But, there exist PTASes for certain problems that can be formulated as QUBO problems, like the Euclidean Traveling Salesperson Problem [211]. Little is known about PTASes on quantum computers; there are many possibilities to try using quantum algorithms to accelerate known classical PTASes. An example might be to apply Grover Search or Quantum Dynamic Programming, cf. Sec. IV D, as sub-routines to achieve (sub-) quadratic speedups compared to purely classical PTASes. Like Grover Search, Quantum Dynamic Programming is also expected to require FTQC. Since exact algorithms and acceleration of PTASes most likely require FTQC, there is arguably more hope for a near-term quantum advantage in optimization sourced from quantum approximation algorithms and heuristics.

Originally proposed for quantum chemistry applications, the Variational Quantum Eigensolver (VQE) [184] denotes a class of heuristic VQAs to approximate ground states of Hamiltonians. This is achieved by choosing a parametrized quantum state $|\Psi(\vartheta)\rangle = U(\vartheta)|\Psi_0\rangle$, where $U(\vartheta)$ denotes a parametrized quantum circuit and $|\Psi_0\rangle$ denotes an initial state, and then using a quantum computer to evaluate $\langle\Psi(\vartheta)|H|\Psi(\vartheta)\rangle$ for given parameter values ϑ, and employing a classical optimizer to minimize the expected value of this quantity. The variational principle guarantees that the expectation value is lower bounded by the ground state energy of H, with equality if and only if the ground state is reached.

The Quantum Approximate Optimization Algorithm (QAOA) [185] denotes a special case of VQE that can be applied to find good solutions to QUBOs. QAOA is motivated by QAA and has been shown to be a computationally universal algorithm [212]. More precisely, QAOA uses a particular problem-dependent ansatz given by pairs of unitaries in each layer. These unitaries have the form

$$U_X(\beta) = e^{-i\beta H_X}, \tag{3}$$

$$U_P(\gamma) = e^{-i\gamma H}, \tag{4}$$

and they are termed as mixing and problem unitaries, respectively. The unitaries are repeated alternatingly with new parameters for each layer and applied to the initial state $|+\rangle^{\otimes n}$, as shown in Fig. 4. Here, H denotes the problem Hamiltonian, and H_X denotes the mixing Hamiltonian, see Sec. III B for more details. Both unitaries are alternated and repeated p times, with individual parameters β_j, γ_j for $j = 1, \ldots, p$.
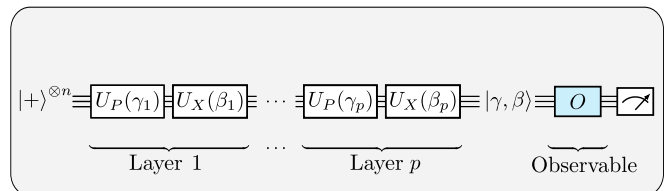

Figure 4. Structure of QAOA ansatz: every layer consists of a mixing and a problem unitary with corresponding parameters γ_j and β_j for layer $j = 1, \ldots, p$.

QAOA is an approximation algorithm with worst-case performance bounds for certain problems and algorithmic settings. For example, there exist worst-case performance guarantees for QAOA with $p = 1, 2, 3$ layers for the MAXCUT problem on 3-regular graphs. Specifically, for $p = 1$ Farhi *et al.* [185] derived a lower bound on the approximation ratio given by 0.692, while Wurtz and Love [213] found that a lower bound given by 0.7559 for $p = 2$ and (under certain assumptions) 0.7924 for $p = 3$. Furthermore, the performance of QAOA for (large-girth) d-regular graphs has been analyzed numerically and it has been shown to outperform many known SDP-based relaxations from $p \geq 11$ [214, 215]. However, for general QUBO, QAOA is a heuristic without performance guarantees.

A crucial task for VQAs is to determine the optimal parameters, cf. Sec. III F. The problem-dependent approach QAOA to construct an ansatz implies a relatively small number of parameters, which is desirable for the corresponding classical optimization loop. Considerable research has been published on training strategies for QAOA as well as on properties of the optimal param-

eters. There are different strategies to solve this problem as we will discuss now.

The conventional approach to parameter optimization in a VQA is to leverage the quantum computer to evaluate the loss function and subsequently use a classical optimizer to improve the parameters. Good parameter initialization is crucial for this task to succeed and there is increasing evidence that this requires some physics-inspired initialization, rather than generic random initialization [216–218]. Proposed strategies range from annealing-inspired parameters [217], to machine learning based approaches [219]. Similarly, one can try to evaluate the loss function for the optimization problem classically and only sample solutions from the quantum computer. The rationale here is that evaluating (or approximating) the (two-)local observables involved in QAOA for QUBO using say, tensor networks or Clifford perturbation theory, is computationally easier than sampling from the complete quantum state [220, 221]. Alternatively, for some classes of QUBOs, QAOA has a property called *concentration of parameters* or *transfer of parameters* [220, 222–224]. For these problems, the optimal parameters become instance independent. In other words, one can train the parameters for one instance, and reuse it for other instances. Depending on the setting, the instances can be of the same size, but also of larger size. This may even allow for training of a smaller instance classically and reusing the parameters for larger problem instances where classical simulation is not possible anymore. Moreover, this could enable good solutions to a problem very quickly by just using pre-optimized parameters. Sometimes, the opposite direction is also possible, i.e., optimal parameters for finite instances are derived from studying the limit of infinite size instances [214, 225]. Parameter optimization for QAOA in general, and transfer/concentration of parameters in particular, are crucial to progress towards a successful application of QAOA for practical settings.

Several variants of QAOA have been proposed over time. Many of them propose different mixers and initial states to achieve certain goals. The Quantum Alternating Operator Ansatz, which shares the acronym QAOA, denotes a family of QAOA variants that encode constraints into the mixer such that they are preserved and the algorithm is restricted to feasible states. This will be discussed in more depth in Sec. IV A 2. Other variants adjust QAOA in order to warm-start it from solutions obtained by classical algorithms. The idea is to use a classical algorithm to compute an approximate solution, and then use it to warm-start QAOA to further improve upon the obtained solution [197, 198]. These approaches inherit some of the performance guarantees of the classical algorithms used for warm-starting and have been shown to significantly outperform standard QAOA on certain instances. With a modification of the ansatz of QAOA based on the warm-starts, one can still show the convergence to MAXCUT under the adiabatic limit (i.e., when the circuit depth $p \to \infty$). However, showing

provable guarantees on warm-started QAOA for finite circuit depth with these custom mixers remains an open question. Other variants suggest to reuse qubits, leveraging mid-circuit measurements and dynamic circuits, to increase the number of variables for a limited number of qubits [226]. It seems also natural to restart the search, when one arrives in so-called barren plateau [196], where the (sub)gradient is not informative enough. Mastropietro *et al.* [196] have shown that the more barren regions there are and the higher their proportion in the parameter space, the higher the speedup obtained by restarting.

Another alternative approach to QAOA is Recursive QAOA [227], which uses the quantum computer to produce a sequence of reduced problems. Here, at each step QAOA is used to estimate correlations between variables, and the problem Hamiltonian is reduced by fixing the strongest one. This process is iterated until the reduced problem becomes small enough to for (provably exact or heuristic) classical solvers, which then yields an approximate solution to the input problem. Recent works [228–232] have generalized recursive quantum approaches to problems beyond MAXCUT and to circuit ansätze beyond QAOA.

In the same paper that introduces Recursive QAOA [227], the authors also show that, in the large problem limit, constant-depth QAOA cannot outperform the classical Goemans-Williamson approximation for certain instances of MAXCUT on d-regular graphs, i.e., the depth needs to grow with the problem size. However, these results are not applicable to alternative approaches, e.g., other QAOA algorithm variants or Recursive QAOA. There are further results about the limitations of QAOA and requirements on circuit depth with respect to problem size and structure. For instance, Farhi *et al.* [233] show that QAOA "needs to see the whole graph" for a particular QAOA approach to the Maximum Independent Set (MIS) problem. More precisely, the authors show that for $p < \log(n)/\log(d/\ln(2))/2$, where n denotes the number of nodes in the graph and d its degree, for large enough d, the approximation ratio of QAOA is upper bounded by 0.854. However, for larger p, no such limitation has been found. Since the logarithm grows slowly in n, this is not too strong of a restriction. Similar obstructions to low-depth QAOA have also been shown to apply to k-local generalizations of MAXCUT [234].

QAOA also inspired the development of classical algorithms, such as the the Mean Field Approximate Optimization Algorithm [235]. Here, the QAOA circuit is approximated through mean-field approximations, which is demonstrated to perform well for certain problem instances. This may also be of interest as a classical alternative to train or initialize the QAOA parameters before sampling solutions from a quantum computer.

Since the ansatz for QAOA is derived from the problem, the structure of the problem directly affects the required qubit connectivity of the quantum circuit. Given the often sparse-connectivity of quantum devices, this re-

quires a transpilation step to map the problem/circuit to the hardware. The transpilation and execution of quantum circuits on noisy hardware will be discussed in depth in Sec. V. However, addressing this issue at the problem and modeling level might lead to significant improvements through hardware-optimized problem formulations. A more detailed review of QAOA can be found in [236].

Another family of near-term compatible approximation algorithms for MAXCUT is given by Quantum Random Access Optimization (QRAO, also called Quantum Relaxation Algorithms) [237, 238]. Here, the variable-to-qubit ratio is chosen to be larger than one and results in a non-diagonal Hamiltonian. This allows the encoding of larger optimization problems into a given number of qubits. However, depending on the chosen encoding and resulting ratio, there might be restrictions on the allowed quadratic terms in the QUBO, i.e., edges in a graph. Assuming a feasible graph, and that we can prepare a good enough approximation of the ground state of the corresponding non-diagonal Hamiltonian, which, in general, cannot be guaranteed, it has been shown that QRAO achieves approximation ratios for MAXCUT of at least 0.555, 0.625, and 0.722, for variable-to-qubit ratios of 3, 2, and 1.5 respectively. Furthermore, [238] introduce another QRAO with ratio 2 that does not impose any constraints on the graph, but might not always achieve a non-trivial performance guarantee (i.e., $> 1/2$). Since providing the sufficiently accurate approximation of the ground state of the Hamiltonian can in general not be guaranteed, QRAO is strictly speaking not an approximation, but a heuristic algorithm. There also exist other encoding schemes that represent more variables than qubits and provide some variational heuristics for optimization algorithms [239]. Note, that in principle, one could encode up to 2^n binary variables into n qubits. However, that would usually imply that writing down the problem (with a cost matrix in $\mathbb{R}^{2^n \times 2^n}$) would be similarly expensive as simulating the involved n-qubit circuits classically. Thus, only moderate, i.e., polynomial, encodings make sense unless the problem is given in a compact function form and access to the full solution is not needed, only the objective value. That is a similar situation, as for instance, in semidefinite programming, cf. Sec. IV B 1.

While neither QAOA nor QRAO achieve the approximation ratios of classical approximations for MAXCUT, i.e., 0.878567, they are nonetheless very interesting heuristics. As discussed in Sec. II, there exist problems with approximation algorithms where the known lower bounds do not yet achieve the inapproximability bounds i.e., where there is room for improvement for classical and also quantum approximations. This represents a very interesting direction of research and potential quantum advantage over known classical approximations.

Another time evolution-based heuristic is quantum imaginary time evolution (QITE) [100, 240]. Like QAA, QITE assumes an initial state with a non-zero overlap

with the ground state of H. Then, it applies the evolution

$$|\psi_\tau\rangle = \frac{e^{-H\tau}}{Z_\tau} |\psi_0\rangle, \qquad (5)$$

where $\tau = it$ and $Z_\tau = \sqrt{\langle\psi_0|\, e^{-2H\tau}\, |\psi_0\rangle}$ denotes a normalization factor related to the partition function. QITE exponentially suppresses amplitudes of eigenstates with larger eigenvalues and amplifies the amplitudes of the ground state. The time T to exceed a fixed overlap with the ground state, assuming at least an exponentially small initial overlap with the ground state, scales inversely to the spectral gap Δ of H and linear in the problem size, i.e., $T = \mathcal{O}(n/\Delta)$ [241]. Unlike QAA, the Hamiltonian is not time-dependent, and thus, the gap does not depend on the time and not necessarily on the problem size. In addition, a small gap might also imply that there are multiple very good solutions, which – for shorter times – would make it an interesting heuristic. However, QITE is a non-unitary evolution and can be challenging to implement on a quantum computer [100]. If it were possible, it would allow **QMA**-hard problems such as generic ground state search or Gibbs state preparation to be solved [55]. Thus, only heuristic approximations of QITE exist to the best of our knowledge. These include projecting the non-unitary evolution to unitary operations [100] or embedding it in a unitary of augmented dimension [242]. Another family of algorithms are VQAs that map the time evolution to an ansatz by means of variational principles [240, 243] or directly projecting a Taylor-expansion of the time-evolution operator onto the ansatz [244, 245]. Variational approaches based on variational principles are also closely related to alternative optimizers, such as Quantum Natural Gradients [246] as well as stochastic approximations [247]. The advantage of these heuristics is that they do not require the ansatz to be derived from the problem, which makes it easier to implement them on real hardware and allows one to apply them also to unconstrained black box binary optimization [200]. However, at the same time, that is a potential disadvantage since it is unclear what circuit structure might work and there is less theoretical foundation on why they may lead to a potential quantum advantage. The design of suitable circuits, possibly in an adaptive way, is a key open question for these type of algorithms.

In addition to variational approximations to QITE, there are also variational approximations to real-time evolution [243, 248]. Those can be used to approximate QAA and form another family of heuristics. However, like for QITE, the design of the ansatz is a crucial open question. In addition, all of these variational time evolution-based algorithms introduce relatively high costs in terms of the number of circuits to be evaluated which may lead to a significant bottleneck to scale them [249] — the cost is increased even further if bounds on the preparation accuracy are computed [250].

While VQAs are often defined as minimizing an expectation value, in quantum optimization we might not care about an expectation value but rather about sampling good solutions, which gives us some flexibility in the choice of cost function. Alternative cost functions can help to increase the robustness against noise or relax the requirements on an ansatz. A frequently used example is the Conditional Value at Risk (CVaR) [251, 252], where we do not average over the objective values corresponding to all samples obtained from measuring a quantum state. Instead, we sort them and only take the average over the best α-fraction of samples, for a pre-defined $\alpha \in [0, 1]$, where $\alpha = 1$ corresponds to the full expectation value and $\alpha = 0$ corresponds to the single best observed sample. It has been shown by Barron *et al.* [252] that choosing α based on the present noise when evaluating the corresponding circuit on a noisy quantum device, CVaR can provably bound noise-free expectation values. Another proposal is the Gibbs objective function [178]. The rationale is the same as for the CVaR, it puts more emphasize on good samples than on bad samples, i.e., it biases the expectation value.

Within this section, multiple possible quantum optimization algorithms have been discussed. All of them are limited by the number of available qubits on existing respectively near-term hardware. However, if an empirical quantum advantage had been demonstrated for a smaller problem, it might be possible to extend it, e.g., using decomposition or multi-level schemes to leverage the quantum computer also for larger problems [253].

Unconstrained discrete optimization is a rich research domain that has already led to plenty of results. Nevertheless, there are still many open questions to be answered regarding the potential for quantum advantage for this problem class. Important questions can be found at every level, from quantum-specific problem formulations, to better encodings of integer variables, and more efficient encodings of problems in general [206]. Furthermore, a key step in classical solvers is pre-processing which can simplify problems tremendously. However, little is known about the potential of quantum-specific pre-processing. In addition, classical algorithms, such as branch-and-bound, often come with a posteriori bounds on the optimality gap. This is crucial because in many cases this results in almost optimal solutions and also helps to determine how much compute resources to invest in finding good solutions to a given problem. In addition to a priori or a posteriori performance bounds, it is crucial to understand how they carry over when the algorithm is executed on noisy quantum hardware and how problems can be optimally modeled and mapped to the hardware to minimize the impact of noise. Because these questions will often be difficult to answer, systematic benchmarking becomes even more important, cf. Sec. VI. To conclude, with the availability of quantum computers with more than one hundred qubits, it is crucial to develop and test quantum optimization heuristics on these devices for non-trivial problem sizes. This is at a

scale where exact simulation is not possible anymore, and even if approximate simulation might be possible, it does not help to learn how to deal with the noise, and thus, how algorithms are scaling in practice. Developing this intuition is key to progress towards a practical quantum advantage in optimization. While discrete unconstrained optimization already represents a very broad domain of optimization, many practically relevant problems naturally come with constraints, such as budget, capacity, or structural constraints. Thus, in the next section, we discuss different approaches to incorporate constraints into quantum algorithms for discrete optimization.

2. Constrained Discrete Optimization

Constrained Discrete Optimization is defined by adding constraints of the type $g(x) = 0$ or $g(x) \leq 0$ to an Unconstrained Discrete Optimization problem. In the following, for simplicity, we consider QUBO and add linear inequality and equality constraints, although some of the algorithms could be applied to more complex types of constraints as well. The conversions for higher-order polynomial objective functions to quadratic functions and from integer to binary variables still apply as discussed in Sec. IV A 1. Thus, the considered problems are given by

$$
\begin{aligned}
\min_{x \in \{0,1\}^n} \quad & x^T Q x \\
\text{s.t.} \quad & A x = b, \\
& C x \leq d
\end{aligned}
\tag{6}
$$

for matrices $A \in \mathbb{R}^{m_e \times n}$, $C \in \mathbb{R}^{m_i \times n}$, and vectors $b \in \mathbb{R}^{m_e}$, $d \in \mathbb{R}^{m_i}$, where m_e, m_i denote the number of equality and inequality constraints, respectively. Note, that unlike for QUBOs, linear objective functions (i.e., a diagonal Q) are interesting due to the presence of constraints.

Under mild assumptions on A and C, this can always be converted to QUBO by adding slack variables to convert all inequality constraints to equality constraints and then converting equality constraints to penalty terms by squaring the right-hand-side and adding it to the objective function with a large weight [203]. Some constraints can also be encoded into penalty terms more efficiently (see, e.g., [203]). While this is always possible, it may come at a substantial cost in terms of additional binary variables and a dramatic increase in non-zero terms in the cost matrix in Eq. (6); i.e., it often maps an otherwise sparse matrix Q to a dense one. Further, adding penalty terms scaled by a large constant can tremendously amplify the range of values of an objective function, which can imply numerical instabilities or requirements to control certain parameters in the software or hardware stack up to an infeasible accuracy. These are some of the reasons why explicitly keeping the structure can lead to more efficient algorithms. In the following, we discuss strate-

gies to incorporate constraints directly.

Like for unconstrained discrete optimization, we can leverage Grover Search to achieve a quadratic speedup over brute force search [101]. This requires not only an oracle for the objective function, but also one for each constraint such that we can mark all feasible solutions. Since brute force search is rarely the most efficient classical algorithm the achieved quantum advantage is usually sub-quadratic or sometimes not present at all. However, Grover Search might still be useful as a subroutine in other algorithms.

Classically, constrained discrete optimization problems are often solved using branch-and-bound algorithms. In [254], a quantum branch and bound algorithm is introduced that accelerates classical branch-and-bound algorithms. It is shown that it solves most instances of the Sherrington-Kirkpatrick model to optimality with a high probability in time $\mathcal{O}(2^{0.226n})$. The algorithm adapts Grover Search to be applicable to search branch-and-bound trees. While this has the potential to accelerate classical schemes, it has the disadvantage that no a posteriori bounds on the optimality gap are generated, which often allow an early termination of such search schemes. Further, it requires FTQC, which adds additional overhead.

There are multiple proposals to incorporate constraints in VQAs. The most straight-forward approach is to design an ansatz that – in the noise-free case – is restricted to the feasible space. For instance, suppose a constraint that fixes the Hamming weight of feasible bit strings, i.e., the number of ones. Then, we can prepare as initial state a uniform superposition of all these feasible bit strings, i.e., a Dicke state [255]. Or, we can use a parametric iSWAP or a Givens rotation gate, like in a particle preserving ansatz in quantum chemistry [256], to construct an ansatz that only generates feasible bit strings. This is straight-forward for some type of constraints, but more difficult in general. Further, it takes the constraints into account, but not yet the objective function.

The Quantum Alternating Operator Ansatz (QAOA') [257, 258] generalizes the Quantum Approximate Optimization Algorithm to much wider classes of problems and encodings, in particular to problems with hard constraints. Here, more general initial states are considered, such as ones from the subspace of feasible states for constrained problems. The QAOA operators of Eq. (3) are generalized to arbitrary families of phase of parameterized operators, again applied in p alternating layers. In particular, mixers can be constructed as ordered products of non-commuting local operators such that problem hard constraints are guaranteed to remain satisfied. This has the advantage to keep the structure of the objective function and often leads to significantly cheaper implementations of cost operators than when adding constraints as penalty terms. Alternatively, one can also directly replace the mixing Hamiltonian H_X in Eq. (3), e.g., for Hamming weight constraints by XY-model [259, 260] or Grover-based [190, 261] Hamiltonians. However, all of this leads to more complex mixers, and implementing them can become challenging. Hardware-efficient ansatz variants [229, 262–264] are one alternative approach toward alleviating this difficulty.

Another approach is to leverage quantum Zeno dynamics to repeatedly project the state back to the feasible subspace [265]. This requires auxiliary qubits as well as projective mid-circuit measurements. While this allows one to include multiple constraints of different types, the resulting circuits become deeper and are likely to require FTQC.

Instead of handling constraints explicitly in the quantum circuit, unconstrained black box binary optimization [200] allows hiding constraints in the black box objective function. Instead of requiring quadratic penalty terms for every constraint, a general metric for infeasibility can be defined, which allows one to add a more balanced penalty term to the objective without the disadvantages mentioned earlier in this section. However, as mentioned in Sec. IV A 1, it is unclear how to construct promising parameterized circuits.

Alternatively, Iterative or Recursive QAOA variants, cf. Sec. IV A 1, may be adapted to enforce constraints as suggested in [230]. Here, as for the unconstrained case, problems are solved iteratively and usually one variable is removed at a time. Adjusting the selection rules for how variables are fixed can allow one to enforce constraints in certain cases.

Adding constraints as penalty terms to the objective function, i.e., casting constrained binary optimization to QUBO, is often the main driver for the density of the cost matrix, and thus, the challenges to implement the algorithms on real hardware. If at least some constraints, e.g., cardinality, packing, or covering constraints, could be natively incorporated into an algorithm, this could lead to sparser problems, and thus, could lead to significant simplifications for implementations on real quantum hardware. This is sometimes called QUBO-Plus [266], i.e., QUBO plus special types of constraints. For classical solvers, these constraints can even simplify the problem, since the knowledge about the optimal solution can be leveraged within the algorithm. This is also the idea behind most of the presented quantum algorithms within this section. However, often the complexity is just pushed to a different part of the algorithm, for instance, from the QAOA cost operator to its mixer.

The approaches for constrained discrete optimization mentioned in this section essentially leverage three strategies to handle constraints: they use Grover Search (and variants), they add penalty terms to the objective function, and they try to restrict or project states to the feasible space. Since constraints are appearing almost everywhere in practice, further progress on how to efficiently handle them is crucial on the path towards quantum advantage in optimization.

B.   Continuous Optimization

1.   Convex Optimization

Convex optimization problems can often be solved efficiently, both in theory and in practice, and find numerous applications in business, science and engineering. Quantum algorithms for convex optimization have been proposed for several relevant classes of problems.

One of the most general formulations of a convex optimization problem is the task of minimizing a convex function $f : \mathbb{R}^n \to \mathbb{R}$ over all $x \in \mathbb{R}^n$ that have at most a certain size, say $\|x\|_2 \leq R$ for a given bound R, when one is provided black-box access to evaluate the function and possibly its derivatives. For example, given access to an oracle that on input $x \in \mathbb{R}^n$ outputs $(f(x), \nabla f(x))$, our task is to find an approximate minimizer of f over the convex set $\{x \in \mathbb{R}^n : \|x\|_2 \leq R\}$. Gradient descent is an example of a well known algorithm that works in this framework. A natural question is whether quantum computers offer a speedup, be it in query complexity or run time. The answer to this question is not always positive; see, e.g., [267, 268], which consider black-box access to the function and its (sub)gradients. In this setting, classical and quantum algorithms have the same lower bound (no n-dependence, but a polynomial scaling in the inverse of the desired precision ϵ) for the query complexity in general. The other regime, where we do allow a polynomial n-dependence, but only a polylogarithmic dependence on the desired precision, is equally well studied. Under mild assumptions, one can solve general convex optimization problems in polynomial time using e.g., the ellipsoid method [269]. In this very general setting one aims, for example, to solve a problem of the form $\min c^T x$ s.t. $x \in K$ for some bounded convex set K to which we have various types of oracle access (membership, separation, optimization). Here mild quantum speedups (and no-go results) are known [270, 271].

Often much more can be said when we consider convex optimization problems with a certain structure. The most notable such classes are arguably linear programming (LP) and semidefinite programming (SDP). Here the objective is to minimize a linear function subject to linear inequalities on non-negative vectors (LP) or on positive semidefinite matrices (SDP). An SDP can be expressed in the form

$$\begin{aligned} \max_{X \succeq 0} \quad & \operatorname{tr}(CX) \\ \text{s.t.} \quad & \operatorname{tr}(A_i X) \leq b_i \, \forall i \in [m] \end{aligned} \tag{7}$$

where the problem is defined by symmetric matrices $C, A_1, \ldots, A_m$, each of size $n \times n$, and a vector $b \in \mathbb{R}^m$. Note that LPs correspond to a special case of SDP in which each of the input matrices is diagonal. Quantum algorithms for these problems usually trade off an improved dependence on the dimension n and the number of inequalities m with a worse dependence on the preci-sion ϵ to which one solves the problem. We can categorize quantum algorithms for these problems based on the solution methodology.

First-order methods. The first quantum algorithms for SDPs were first order methods, based on the Multiplicative Weights Update (MWU) framework [272]. In this framework one uses candidate solutions X that are proportional to $\rho = \exp(\sum_{i=1}^m y_i A_i) / \operatorname{tr}(\exp(\sum_{i=1}^m y_i A_i))$ for some (sparse) vector $y \in \mathbb{R}^m$. That is, the candidate solutions are proportional to Gibbs states, which naturally correspond to trace-normalized positive semidefinite matrices and can be efficiently prepared on a quantum computer under some conditions. To implement the MWU framework, one additionally needs to compute trace inner products $\operatorname{tr}(A\rho)$ between such Gibbs states ρ and a matrix A (which is either one of the A_i's or C), and essentially solve a search problem based on these values. All these steps can be performed on a quantum computer, with various levels of efficiency; the initial run time bounds derived in the pioneering papers [164, 165] were subsequently improved [166, 167, 273], and further to [167].

The run time of quantum MWU methods depends on several natural parameters; the dimension n, the number of constraints m, the desired (additive) error ϵ in the objective function; and on two instance-specific parameters, typically denoted by R and r, that are related to the diameter of the feasible region for the primal $(\operatorname{tr}(X) \leq R)$ and dual problems $(\|y\|_1 \leq r)$, and bound the so-called "width" of the oracle. The initial works had a run time scaling as $\widetilde{\mathcal{O}}\left(\sqrt{mn}s^2 \operatorname{poly}\left(\frac{Rr}{\epsilon}\right)\right)$, where s is the row sparsity of the input matrices [164, 165] and $\widetilde{\mathcal{O}}$ ignores logarithmic factors, later works improved this to $\widetilde{\mathcal{O}}\left((\sqrt{m} + \sqrt{n})s^2 \operatorname{poly}\left(\frac{Rr}{\epsilon}\right)\right)$. Throughout, the polynomial was of a high degree; the state of the art in a certain input model is $\widetilde{\mathcal{O}}\left(\left(\sqrt{m} + \sqrt{n}\frac{Rr}{\epsilon}\right)s\left(\frac{Rr}{\epsilon}\right)^4\right)$, see [167]. In all these bounds, the parameters r, R, and ε appear together as one "scale-invariant" parameter $\gamma = Rr/\epsilon$ [165]. Often, this parameter γ scales poorly in terms of n and m [165], especially when parameters r, R scale linearly or superlinearly with n. (For example, in the SDP relaxation of MAXCUT [179], cf. (9), $\operatorname{tr}(X)$ scales linearly with n.) Notable exceptions are applications to shadow tomography, quantum state discrimination, and E-optimal design (i.e., optimal design in which one maximizes the smallest eigenvalue of the information matrix) [166, 167]. As an aside, we mention a second famous appearance of the MWU framework in the quantum (complexity) literature: it is a key tool in the proof of the equality **QIP** = **PSPACE** [274].

For comparison, the classical complexity of a general-purpose SDP-solver based on the MWU framework is $\widetilde{\mathcal{O}}\left(mns\left(\frac{Rr}{\epsilon}\right)^4 + ns\left(\frac{Rr}{\epsilon}\right)^7\right)$, see [165], but the input models of the classical and quantum algorithms are not necessarily equivalent so these results are difficult to compare. Namely, quantum algorithms often rely on the use of Quantum Random Access Memory (QRAM), which is

a RAM addressed by qubits. QRAM accessed by a unitary, swaps the state of the addressed memory cell with the state of a target qubit. Although concrete proposals exist [275, 276], a scalable fault-tolerant implementation is yet to be found [277–280], making this at best a long-term prospect. The QRAM model of computation is common in the quantum optimization literature, and it is also studied in some of the papers mentioned in other sections: we mention it here because the frameworks discussed in this section are particularly reliant on it. Also, in the classical literature the MWU framework was specialized (and significantly accelerated) for the SDP relaxation of structured combinatorial problems such as a form of QUBO, balanced separator, and sparsest cut, see [19, 272].

So far, we have discussed the MWU framework in the context of SDPs. One can equally well apply this machinery to the simpler class of linear programs, often obtaining much better run times, see, e.g., the works [175–177] that are based on a zero-sum game approach [281, 282]. In particular, [176] successfully implemented a dynamic data structure in the quantum setting. It is an interesting open question to extend such a result to SDPs. One could also study further first-order methods [283, e.g.], beyond MWU, to SDPs. While these have been prototyped [283], and there may be some speed-up in per-iteration complexity, their iteration complexity remains the same as in the classical case.

Second order methods. Interior Point Methods (IPMs) are popular second order methods for solving structured convex optimization problems both from a theoretical and practical perspective. At their core, IPMs start from a point lying in the interior of a convex feasible set $\mathcal{X} \subset \mathbb{R}^n$, and use Newton's method to solve a sequence of barrier problems of the form

$$\min_{x \in \mathbb{R}^n} \ \eta \cdot \langle c, x \rangle + f(x), \tag{8}$$

where $\langle \cdot, \cdot \rangle$ is an inner product on $\mathbb{R}^n$, $\eta \geq 0$ and $f : \text{int}\,(\mathcal{X}) \mapsto \mathbb{R}$ is a barrier function encoding the constraints defining $\mathcal{X}$. That is, $f(x) \to \infty$ as x approaches the boundary of $\mathcal{X}$, and thus the only "constraint" present in Eq. (8) is that f is only defined on the interior $\text{int}\,(\mathcal{X})$ of $\mathcal{X}$. In each iteration, the value of η is increased, and the set of minimizers $\{x(\eta) : \eta \geq 0\}$ of Eq. (8) define the so-called *central path*, an analytic curve extending throughout the interior of the feasible region to the set of optimal solutions that is uniquely determined by the starting point. By iteratively applying Newton's method, IPMs approximately follow the central path to the optimal solution via a sequence of local linearizations (i.e., Newton steps). Computing the Newton step requires one to solve a linear system of equations, and constitutes the dominant operation at each iterate of an IPM. It is therefore natural to try to use quantum linear system solvers [140, 284] in an effort to accelerate the computation of the Newton step [285]. Work on this line of research was initiated by [286], with several follow-up

works trying to address the shortcomings in the initial proposal [287–289]. Because the direction computed in the Newton step is necessary to process the subsequent iteration, the quantum linear system algorithm is followed by the application of a quantum state tomography algorithm. Specifically, two issues have emerged: the use of state tomography introduces inexactness in the search direction and a $1/\epsilon$ dependence in the run time, where ϵ is the precision to which tomography is performed [290]; and the use of the quantum linear systems algorithm introduces a run time dependence on the condition number of the Newton linear system, which, unfortunately, goes to ∞ as we approach optimality. The first issue can be dealt with by developing an IPM framework that allows for inexact search directions [287], and iterative refinement techniques can alleviate (but not necessarily eliminate) the remaining issues [288, 289]. Overall, assuming access to QRAM, quantum IPMs achieve favorable dependence on the size of the problem (m and n) for linear and semidefinite optimization compared to classical algorithms, but the remaining dependence on some numerical parameters (e.g., a condition number bound for the Newton systems) makes it unclear if an overall speedup is achieved.

Very recently, a quantum speedup for interior point methods was achieved without introducing a dependence on a condition number [291]. Under some mild assumptions and access to QRAM, this IPM achieves a quantum speedup for linear programs in which the number of linear inequalities m is much larger than the number of variables n, i.e. LPs of the form $\max c^T x$ s.t. $Ax \geq b$ where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ with $m \gg n$. In particular, the best classical solver achieves a scaling of $\widetilde{\mathcal{O}}(mn + n^{2.5})$ [292], whilst the quantum analogue scales as $\sqrt{m} \cdot \text{poly}(n, \log(1/\epsilon))$. The key new subroutine here is a faster quantum algorithm to compute spectral approximations of matrices of the form $B^T B$, provided query access to a tall matrix $B \in \mathbb{R}^{m \times n}$. A spectral approximation of such matrices suffices to speedup the costly Newton step. This subroutine can be seen as a generalization of a quantum graph sparsifier (by letting B be the edge-vertex adjacency matrix) [293], which in turn has been used in optimization algorithms in both the continuous [294] and discrete settings (e.g. cut approximations) [293].

Quantum SDP-solvers for approximating MAXCUT. Given the challenges in evidencing end-to-end speedup for general SDPs posed by the scale invariant error parameter Rr/ϵ, the MWU framework has been specialized to the semidefinite approximation of MAXCUT:

$$\begin{aligned}\max_{X \succeq 0} \quad & \text{tr}(CX) \\ \text{s.t.} \quad & X_{ii} = 1, \, \forall i \in [n].\end{aligned} \tag{9}$$

Brandao *et al.* [181] demonstrated that, after normalizing the diagonal constraints by n and relaxing slightly

the constraints, feasibility reduces the task of solving the MAXCUT SDP to a possibly simpler task: preparing a Gibbs state that is (i) approximately indistinguishable from the maximally mixed state when measured in the computational basis, and (ii) whose trace inner product with C is close to an estimate of the optimal objective value, determined via binary search. This enables the design of specialized oracles for testing feasibility within the MWU scheme, and leads to an algorithm that, provided access to QRAM, outperforms previous algorithms in the problem dimension and sparsity of C. However, the algorithm exhibits a poor $\mathcal{O}(\epsilon^{-28})$ dependence on precision, prohibiting an overall speedup.

The impractical error scaling was addressed by Augustino *et al.* [174], where iterative refinement techniques are used to exponentially improve the dependence on the inverse precision, for both the quantum and classical algorithms proposed by Brandao *et al.* [181]. Moreover, it is shown that when C is stored in QRAM, one can remove the dependence on the sparsity parameter by using Gibbs sampling techniques based on Quantum Singular Value Transformation [143]. This yields a quantum algorithm that runs in time $\widetilde{\mathcal{O}}\left(n^{1.5}\right)$, plus the time it takes to read C, obtaining a polynomial speedup over the best known classical algorithm. Due to the lack of a classical lower bound for the problem, however, it is possible that better (and faster) classical algorithms for this problem could be developed. Note that the speedup in Augustino *et al.* [174] is reliant on QRAM: the authors analyze their algorithm in the standard gate model and show that the resulting algorithm is outperformed by classical approaches.

Quantum algorithms for matrix scaling and matrix balancing. Aside from the SDP relaxation of MAXCUT, another convex optimization problem that has enjoyed a provable quantum speedup for first- and second order methods is the matrix scaling problem. One version of the matrix scaling problem can be stated as follows: given an entrywise non-negative matrix, find positive weights for the rows and columns such that the reweighted matrix becomes doubly stochastic. This problem has many applications, both in theory (e.g., approximating the permanent [295]) and in practice (used by default as a preconditioner in LAPACK [296]). Recent works [294, 297] have shown how to speedup both first- and second-order methods for a natural convex formulation of the problem, using a variety of techniques (basic amplitude amplification, but also graph sparsification). Importantly, these quantum algorithms do not introduce a dependence on a condition number.

Quantum approaches with no direct classical equivalent. Some classical continuous optimization algorithms can be phrased in terms of the solution of a certain dynamical system, i.e., ODEs. These ODEs can then be cast as the Schrödinger equation, and solved by means of a quantum algorithm, thereby solving the optimization problem. The first work in this line of research is Quantum Hamiltonian Descent (QHD) [298], which can

be viewed as a quantization of the Bregman-Lagrangian framework for continuous-time accelerated gradient descent introduced by Wibisono *et al.* [299]. By properly designing a family of Hamiltonians that, as time progresses, lead to the optimal solution of the dynamical system, quantum Hamiltonian descent can be shown to converge to the optimal solution even on certain non-convex problems (although it may not do so efficiently, as these problems are provably hard). With regard to output, QHD returns a classical description of the solution without the need for state tomography. The final state is a probability distribution concentrated near the global minimizer of the objective function, and thus a bitstring corresponding to a solution is obtained upon sampling from the QHD final state.

Along this line, Augustino *et al.* [300] describe a fully quantum algorithm for solving linear optimization problems by quantum-mechanical simulation of the central path, which the authors call the Quantum Central Path Method (QCPM). While IPMs and QIPMs approximately track the central path using successive linearizations of the perturbed KKT conditions (i.e., Newton steps), the QCPM consists of a single simulation that works directly with the nonlinear complementarity equations. This is achieved by designing a Hamiltonian which encodes the central path in its ground state. Like QHD, no intermediate measurement is required, and a classical description of the solution is obtained upon sampling from the final state. This approach is faster than any classical or quantum IPM in a certain sparsity/condition number regime, while also avoiding the use of QRAM, block-encodings, quantum linear system algorithms, and state tomography.

2. Non-Convex Optimization

In non-convex optimization, one minimizes a function:

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{10}$$

in dimension $n \in \mathbb{N}$. Many variants are undecidable [301], as long as one allows for periodic functions f, or sufficiently non-smooth functions. (As suggested in Tab. I, some are inapproximable in even in the Banach-Mazur model [86].) One can hence hope to study k-th order critical points of non-convex functions, in general. In smooth non-convex optimization problems, this means points where necessary conditions of optimality involving first k partial derivatives are satisfied. In non-smooth non-convex optimization problems, this often involves necessary conditions of optimality involving the Clarke subdifferential [302, 303].

Echoing the situation in convex optimization [267], there are clear limits as to the speedup in non-convex optimization [304]. In particular, when first k partial derivatives are available via an oracle or when one has access to stochastic gradients (SG), and no further assump-

tions are made, classical lower bounds apply to quantum algorithms too. One hence wishes to find suitable special cases, such as by considering the numbers of local minima, local behavior of the functions around those, and behavior of the trajectories connecting nearby local minima ("the barriers").

Consider the problem of finding an ϵ-stationary point $x \in \mathbb{R}^n$ of a non-convex function f, which means that

$$\|\nabla f(x)\| \leq \epsilon. \tag{11}$$

For $x \in \mathbb{R}^n$, consider the set of k-order derivatives $\nabla^{(0,\cdots,k)} f(x) := \{f(x), \nabla f(x), \cdots \nabla^k f(x)\}$ with universal Lipschitz constants, and consider also the stochastic gradient $g(x, \xi)$, with ξ a random seed and satisfying $\mathbb{E}_\xi[g(x, \xi)] = \nabla f(x)$ and $\mathbb{E}_\xi\|g(x,\xi) - \nabla f(x)\| \leq 1$. Assuming quantum access $O_f^{(k)} |x\rangle |y\rangle = |x\rangle |y \oplus \nabla^{(0,\cdots,k)} f(x)\rangle$, which outputs a binary representation of $\nabla^{(0,\cdots,k)} f(x)$, Zhang and Li [304] prove the lower bound $\Omega(\epsilon^{-(k+1)/k})$. Assuming quantum access $O_g |x\rangle |\xi\rangle |y\rangle = |x\rangle |\xi\rangle |y \oplus g(x,\xi)\rangle$, produces a lower bound of $\Omega(\epsilon^{-4})$. Both query lower bounds are the same as the classical case. On the other hand, Sidford and Zhang [305] show query upper bounds for finding critical points with quantum algorithms for stochastic gradient descent (SGD). With access to a probability-weighted superposition over the bounded-variance stochastic gradient, the algorithm requires $O(\sqrt{n}\epsilon^{-3})$ queries, which is only better than the classical algorithm $O(\epsilon^{-4})$ when the dimension is $< \epsilon^{-2}$. This can be improved with access to mean-squared smooth SGs, from $O(\epsilon^{-3})$ to $O(\sqrt{n}\epsilon^{-5/2})$. The run time for all of these algorithms both classically and quantumly will in most cases be polynomial in n.

In high-dimensional situations, the availability of the quantum linear systems algorithm and of quantum linear-algebra methods motivates the direct quantum implementation of gradient descent over $O(\text{polylog}(n))$ qubits [285, 306] to provide descriptions of stationary points of non-convex functions. With a non-convex differentiable function $f(x) : \mathbb{R}^n \to \mathbb{R}$ and an initial point $x^0 \in \mathbb{R}^n$, consider the gradient descent update $x \to x - \eta \nabla f(x)$ with $\eta > 0$. For $T \in \mathbb{N}$, the task is to output a quantum state proportional to x^T, where the update is applied T times starting from x^0. To mimic a classical gradient descent step for fixed $t \in [T]$, define the $\log(n)$-qubit states $|x^t\rangle := \frac{1}{\|x^t\|_2} \sum_{j=1}^n x_j^t |j\rangle$ and $|\nabla^t\rangle := \frac{1}{\|\nabla f(x^t)\|_2} \sum_{j=1}^n \nabla f(x^t)_j |j\rangle$. Let us be given a circuit to prepare $\alpha^t |0\rangle_1 |x^t\rangle + |\phi^t\rangle$, where $|\phi^t\rangle$ is some non-normalized orthogonal state and $|\alpha^t| \leq 1$, and a circuit U_∇ such that $U_\nabla |x^t\rangle = |\nabla^t\rangle$. Using an auxiliary qubit and Hadamard gates, the state $\alpha^{t+1} |00\rangle_{12} |x^{t+1}\rangle + |\phi^{t+1}\rangle$ can be prepared, where

$$\alpha^{t+1} := \alpha^t \frac{\|x^t - \eta \nabla f(x^t)\|_2}{\sqrt{2}\sqrt{\|x^t\|_2^2 + \|\nabla f(x^t)\|_2^2}} \tag{12}$$

and $|\phi^{t+1}\rangle$ is an orthogonal non-normalized state. Hence, the $|00\rangle_{12}$ part of the state proportionally encodes the

desired gradient step and we may continue with the next step. In this setting, Rebentrost *et al.* [285] consider certain classes of multivariate polynomials, and includes Newton steps. While the algorithm may admit a $\text{polylog}(n)$ complexity, amplifying the $|\bar{0}\rangle$ state at the end of the gradient procedure can cost 2^T, since at every step $|\alpha^{t+1}| \leq |\alpha^t|/\sqrt{2}$. An efficient circuit U_∇ may be constructed only for special cases [285, 307], or could be approximated via pre-training a parameterized quantum circuit. For affine gradients corresponding to quadratic optimization problems, Kerenidis and Prakash [306] show an efficient implementation using phase estimation.

The idea of simulated annealing, finding global minima assisted by thermal fluctuations, applies to continuous non-convex optimization as well. Quantum computers obtain polynomial speedups in executing the classical simulated annealing process via quantum walks [308]. Quantum fluctuations provide an additional resource allowing quantum tunneling through barriers between different minima [309]. Recently, Liu *et al.* [310] investigated quantum tunneling for SGD, where the SGD process is approximated by a continuous-time stochastic differential equation. Mastropietro *et al.* [196] investigated restarting the search, when one arrives in so-called barren plateau, which allows for speed-up when there are many or large barren regions. These quantum walks can provide a speedup over classical SGD when the barriers between different local minima are high but thin and the minima are flat.

There are several promising directions and open questions. What are lower bounds for various settings of quantum polynomial optimization? Beyond lower bounds, more in-depth focus on possible run time advantages of quantum algorithms is required. One direction is about the quantum speedup could one obtain by utilizing moment/sum-of-squares approaches to polynomial optimization and quantum algorithms for SDPs, in analogy to mixed-integer programming discussed below. Extending to larger classes of polynomials and beyond, what classes of non-convex functions would allow for amplitude-encoded embeddings, efficient quantum circuits for gradient computations, and converging gradient procedures.

C.   Mixed-Integer Programming

Combining discrete decision variables and continuous decision variables, Mixed-Integer Programs (MIP) are non-convex and at least as hard as either of the discrete optimization (cf. Sec. IV A above) and continuous optimization (cf. Sec. IV B above) problems they subsume. We differentiate between Mixed Integer Linear Programs (MILP), usually used synonymously to MIP, and its generalization Mixed Integer Quadratic Programs (MIQP). In practice, some instances with unbounded variables are notoriously hard [311]. Having said that, in the case of mixed-integer linear programming, espe-

cially when restricted to binary variables, classical solvers based on branch-and-bound-and-cut have made an astonishing progress over the past three decades, often solving instances of up to one million binary variables to proven optimality, cf. Fig. 6, and sometimes solving structured problems with tens of millions of binary variables [312] to proven optimality.

Following Eq. (2), let us consider the following illustrative MIQP:

$$
\begin{aligned}
\min_{x \in \{0,1\}^{n_b}, y \in \mathbb{R}^{n_c}} \quad & x^T Q x + y^T R y \\
\text{s.t.} \quad & Ax = a, \\
& By = b, \\
& Cx \le c, \\
& Dy \le d
\end{aligned}
\tag{13}
$$

where $n_b, n_c \in \mathbb{N}$ denote the numbers of binary and continuous variables, respectively, $Q \in \mathbb{R}^{n_b \times n_b}$ and $R \in \mathbb{R}^{n_c \times n_c}$ are cost matrices, and matrices $A \in \mathbb{R}^{m_e \times n_b}$, $B \in \mathbb{R}^{m_e \times n_c}$, $C \in \mathbb{R}^{m_i \times n_b}$, $D \in \mathbb{R}^{m_i \times n_c}$, and vectors $a \in \mathbb{R}^{m_e}$, $b \in \mathbb{R}^{m_e}$, $c \in \mathbb{R}^{m_i}$, $d \in \mathbb{R}^{m_i}$, define $2m_e$ equality constraints and $2m_i$ inequality constraints.

There are several potential algorithms to approach such a problem with a quantum computer, whose short overview is provided in Tab. II. In more detail:

Branch-and-Bound-and-Cut: the main workhorse in classical optimization. One possibility for a hybrid implementation is that discrete decisions are handled by branching and generation of cuts implemented classically, while the continuous, possibly convex relaxations could be solved on the quantum computer. Considering the progress in quantum algorithms for convex optimization (cf. Sec. IV B 1), this seems plausible, although distant. Perhaps even more ambitiously, one could determine the branching decisions on the quantum computer as well, which would yield an additional quadratic speedup [254, 313–315] over the speedup of quantum algorithms for convex optimization, while increasing the requirements on the numbers of qubits and fault tolerance substantially. We note that the earlier papers [254, 313] consider the speedup in the case of locating *all* optima (measured against the classical cost of finding all optima), whereas Refs. [314, 315] consider the speedup of obtaining one optimum (out of possibly multiple ones), measured against the classical cost of finding one optimum. Recent research by Dalzell *et al.* [316] suggests that super-quadratic speedups may be possible in some cases.

Decomposition/Splitting/ADMM: this is a large class of algorithms that is based on decomposing the problem. Refs. [317, 318] suggested solving the QUBO sub-problem using a quantum computer, while solving the continuous sub-problem classically. This approach is based on the hope that a practical quantum speedup for some type of QUBO can be proven. Under some restrictive conditions [318], such an iterative approach can

be shown to be globally convergent. A similar approach has been tested by Chang *et al.* [319], in the context of Benders decomposition.

Reformulation to an unconstrained optimization problem: Braine *et al.* [320] experimented with a decomposition minimizing values of slack variables used to convert inequality constraints to equality constraints, and a Lagrangian relaxation of the resulting equality-constrained continuous-valued problem. Another, approach that optimizes over the Lagrangian function classically over the circuit parameters and dual variables has been introduced by Le and Kekatos [321]. Such Lagrangian methods are well understood [322, 323], but global-convergence results are limited by very restrictive conditions, in general, or require some form of backtracking. One could also solve the problem approximately by utilizing a test done by Aspman *et al.* [324] for the active set remaining fixed subsequently, and only then utilize the Lagrangian relaxation. This could allow for easier global convergence proofs without backtracking. It should also be noted that there are multiple options for working with both the slack variable, penalties for its non-negativity [325], as well as multiple options for augmenting the Lagrangian with quadratic or higher-order terms in order to improve numerical performance. These are to be explored, yet.

Reformulation to high-dimensional LPs: using techniques that go back to Dantzig and Wolfe [326], one can reformulate mixed-integer programs as in Eq. (13) as a very large linear program (LP). While quantum algorithms for LPs have been somewhat overshadowed by quantum algorithms for SDPs so far (cf. Sec. IV B 1), a continuous development of quantum algorithms for LPs may render this a viable avenue.

Reformulation to high-dimensional SDPs: using moment/sum of squares (SOS) techniques [327, 328], one can reformulate mixed-integer programs as in Eq. (13) to an SDP in a much higher dimension than $n + d$. Indeed, there are instances where this dimension needs to be exponential in $n + d$ [329]. If quantum algorithms for SDPs (cf. Sec. IV B 1) made it possible to solve such instances independent of the dimension, it may offer an interesting avenue for quantum algorithms for mixed-integer programming.

Reformulation to a completely positive problem: a mathematically elegant reformulation of a mixed-integer programming problem results in a linear program over the dual of the cone of copositive matrices [330]. Recall that copositive matrices are real symmetric matrices $Q \in \mathbb{R}^{n \times n}$, corresponding to quadratic forms $x^T Q x$ non-negative on the positive orthant, i.e., for x element-wise non-negative. For a fixed n, these matrices form a convex cone [331], but even the test of membership in this cone is **NP**-hard [332]. These cones have recently attracted attention in mathematical optimization [333]. A possible way to exploit this reformulation in the context of a hybrid solver is to use a quantum computer to solve the difficult cut generation problem, whereas the classi-

cal computer solves a convex problem. This approach has been tried in [334], using a QUBO solver to generate cuts. We note that to prove optimality one eventually needs to show that no cut exists, and this in turn requires an exact solver for the QUBO subproblem. Solving the cut generation problem exactly can be difficult even for classical solvers, see [335] for a discussion of such an approach in the context of power systems.

Next, we review multiple proposals for quantum algorithms for MIP and propose a list of new directions. There are still many open questions to be answered, however, in order to achieve a quantum advantage, which we express in the remainder of this section.

First, can we provide general statements on provable quantum advantage for MIP? Is there a general meta-theorem for combined speedup of solving a convex relaxation and branch-and-bound? What would be the best possible speedup for branch-and-bound? It seems plausible that it could allow for super-quadratic speedup for **NP**-hard problems quite generically, a tantalizing prospect. Quartic speedup has been shown, so far, only for learning majorities [336] and the quantum counterfeit coin problem [337]. Further, how could the Lagrangian approaches benefit from an approximate solution and the test from Aspman *et al.* [324] for the active set remaining fixed subsequently? How does one make design choices within Lagrangian approaches (penalties for non-negativity of slack variables, augmenting the Lagrangian with quadratic or higher-order terms) optimally? And out of the reformulation approaches sketched out above (Dantzig-Wolfe, Moment/SOS, Co(mpletely)-positive), which ones would allow for the best speedup in the continuous optimization (cf. Sec. IV B above)? This entails bounding the ratio of the inscribed to outscribed balls of the reformulations within continuous optimization.

Given the practical importance of MIP, it is crucial to advance our understanding of how to approach this problem class with quantum computers. Most of the proposals in the literature consider a quantum sub-routine inside a known classical algorithm. However, it is unclear what advantage the sub-routine needs to provide to imply an overall advantage of the optimization algorithm for MIP. This could be a combination of speed, quality, but also diversity of solutions in order to accelerate branching. Further, there might opportunities to have more quantum-native algorithms. In any case, it is important to further investigate this problem class and systematically benchmark the performance of available proposals.

D. Dynamic Programming

Dynamic Programming (DP) is a very generic mathematical optimization method that breaks down large optimization problems into smaller ones and combines results for the overall problem in a recursive fashion.

A dynamic programming algorithm consists of repeatedly optimizing the Bellman equation

$$V(x) = \min_a T(x, a) + V(x_a) \tag{14}$$

where a are the actions that are available in the state x and $x_a = F(x, a)$ is the state to which we get by applying a in the state x. The entries $V(x)$ can be arranged into a dynamic programming table. A dynamic programming algorithm then computes all the entries $V(x)$ in this table. The order in which the entries are computed is chosen so that, whenever the algorithm has to compute $V(x)$, all $V(x_a)$ have already been computed and $V(x)$ can be computed from Eq. (14). For example, in the case of TSP [22], x are subsets of the set of city and the entry $V(x)$ is the length of the shortest path through x. Thus, the DP algorithm for TSP consists of computing the shortest paths through every subset of cities.

The complexity of a DP algorithm is $O(Sm)$ where S is the number of possible states x for which we have to optimize the Bellman equation and m is the maximum number of actions in one state. In the case of TSP, this is $\widetilde{O}(2^n)$. DP can be used to address many **NP**-complete problems, some of which are listed in Tab. III. For example, the best algorithm with provable running time for TSP is based on DP [22]. It is also often used as a subroutine in approximation algorithms (e.g., some PTAS), and it can be used to solve Markov Decision Problems (MDPs), as discussed in more detail in Sec. IV E.

Quantum speedups are known for a substantial number of classical DP algorithms. The most important examples are shown in Tab. III. The quantum algorithms are a combination of classical dynamic programming and Grover's search. For example, the quantum algorithm for TSP by Ambainis *et al.* [338] first uses DP to compute the shortest paths through sets of cities that contain up to 24% of all cities and then uses Grover's search to find the shortest combination of those paths that visits all the cities and returns to the starting point. Similar ideas can be applied to other problems in which the classical algorithm uses an exponentially large DP table indexed by subsets of an n-element set. This includes graph coloring [339], minimum Steiner tree [340, 342], tree-width [341] and other problems. The problems amenable to this approach are typically vertex ordering problems (in which the task is to find the optimal ordering of vertices in a graph, as in TSP) or set partitioning problems (in which the task is to find the optimal partition of a set, as in Set Cover or Graph Coloring). Such problems lead to DP tables in the form of Boolean hypercube (2^n entries indexed by $x \in \{0, 1\}^n$) which can often be handled by computing part of the table and then using Grover's search to find the best solution.

One can formulate a generic DP problem with the DP table in the form of hypercube and show that, in the generic case, the quantum speedup can be at most quadratic [338]. It is an open problem whether quantum speedups can be obtained for algorithms with a DP

Table II. A short history of proposed quantum algorithms for mixed-integer optimization. For approaches without available guarantees, we write N/A, although this should not be construed as ruling out guarantees.

Algorithm	Guarantees	Year	Ref.
ADMM	Asymptotic convergence under restrictive conditions. Speedup of QUBO?	2020	[317, 318]
Benders decomposition	N/A	2020	[319]
Branch-and-bound	Quadratic speedup + speedup of convex opt.	2020	[254, 314]
Lagrangian methods	N/A	2021	[320, 321]
Branch-and-bound	Quadratic speedup + speedup of convex opt.	2022	[315]
"Dantzig-Wolfe"	Finite convergence, speedup of convex opt.	2023	Here
Moment/SOS	Asymptotic convergence, speedup by SDPs	2023	Here
Subadditive duals	Asymptotic convergence	2023	Here
Completely-positive	N/A	2023	Here

Table III. Most important provable quantum speedups for dynamic programming, where the notation $\mathcal{O}^*(f(n))$ hides a polynomial factor in n [338].

Problem	Quantum Algorithm	Best Classical Algorithm	Year	Ref.
Generic Vertex Ordering Problem	$O(1.817...^n)$	$O^*(2^n)$	2018	[338]
Traveling Salesperson Problem	$O(1.728...^n)$	$O^*(2^n)$	2018	[338]
Minimum Vertex Cover	$O(\text{poly}(m,n)1.728...^n)$	$O(nm2^n)$	2018	[338]
Graph Coloring	$O(1.7956...^n)$	$O(2^n)$	2023	[339]
Minimum Steiner tree	$O(1.812...^k\text{poly}(n))$	$O(2^k\text{poly}(n))$	2020	[340]
Treewidth	$O(1.538...^n)$	$O(1.755...^n)$	2022	[341]

table of a different structure. For example, the DP algorithm for the Edit Distance problem [343] (in which one is given two strings and has to find the smallest number of symbol insertions/deletions/replacements to transform the first string into the the second string) uses a DP table that is a two-dimensional array. If the size of the array is $n \times n$, the classical DP algorithm takes $O(n^2)$ steps. Quantumly, no quantum speedup is known but an $\widetilde{\Omega}(n^{1.5})$ quantum lower bound is known for the generic case [40, 344].

The broad applicability of DP renders it a very interesting approach. However, many open questions remain in terms of a potential quantum advantage. What structures of state space (besides hypercube) allow general quantum DP algorithms? Can we add stochasticity to quantum DP algorithms or extend them to MDPs (cf. Sec. IV E)? What about the explosion of state space? Can we prove better lower bounds for quantum speedups of generic DP algorithms? In addition to provable algorithms, can we find good quantum approximations or heuristics, such as reinforcement learning? Answering any of these questions will help to further understand the potential and practical applicability of quantum DP algorithms and some of them will also be discussed in the following Sec. IV E.

E.   Optimal Control

Optimization with differential equations as constraints is a subject of optimal control theory. A standard finite horizon optimal control problem takes the following form:

find a control function u with values in a given set $\mathcal{U}$ solving the following optimization problem

$$\min_{u(t) \in \mathcal{U}} \int_0^T \ell(x(t), u(t))dt + h(x(T)) \qquad (15)$$
$$\text{s.t.} \quad \dot{x} = f(x,u), x(t) \in \mathbb{R}^n, \quad t \in [0, T)$$

for some *nice* functions ℓ, f, h, e.g., see Ref. [345]. A globally optimal solution of this problem, the so-called feedback optimal control policy $u^\star(x,t)$ can be found by introducing a value function

$$V(z,t) = \min_{u(t) \in \mathcal{U}} \left\{ \int_t^T \ell(x(s), u(s))ds + h(x(T)), \right.$$
$$\left. \dot{x} = f(x,u), \ x(t) = z \right\}. \qquad (16)$$

Generalizing classical dynamic programming ideas one can demonstrate that V solves the following Hamilton-Jacobi-Bellman (HJB) equation backwards in time:

$$\frac{\partial V}{\partial t} + \min_{u \in \mathcal{U}} \left\{ \ell(x,u) + f^\top(x,u)\nabla_x V(x,t) \right\} = 0, \qquad (17)$$

subject to the *terminal condition* $V(x,T) = h(x)$, for all $x \in \mathbb{R}^n$. Given the solution of Eq. (17) one computes the optimal control policy in feedback form as follows:

$$u^*(t,x) = \underset{u \in \mathcal{U}}{\text{argmin}} \left\{ \ell(x,u) + f^\top(x,u)\nabla_x V(x,t) \right\}. \qquad (18)$$

It should be noted that in many practical situations, $\nabla_x V$ does not exist in the classical sense. For this reason the

solution of HJB Eq. (17) is understood in the viscosity sense: a viscous term represented by $\varepsilon \Delta V$, $\varepsilon > 0$ is introduced in the right-hand-side of Eq. (17) so that the resulting solution V_ε becomes differentiable, and then taking the limit of V_ε as $\varepsilon \downarrow 0$ one gets the viscosity solution of Eq. (17). This construction relates V_ε to the stochastic optimal control [345] as the Laplacian $\varepsilon \Delta V$ can be seen as adding "white noise" to the dynamics of x of variance $\sqrt{\varepsilon}$.

In addition to optimal feedback control design HJB has many applications: for example, it allows one to solve the nonlinear filtering / state estimation problem by constructing reachability sets of nonlinear dynamical systems [346]. The most relevant applications of HJB equations in the context of this paper are in discrete optimization: HJB equation for discrete-time dynamics can be seen as the famous Bellman equation of dynamic programming:

$$V(t,x) = \min_{u \in \mathcal{U}}\{\ell(x,u) + V(t+1, x + f(x,u))\} \quad (19)$$

with final time condition $V(T,x) = h(x)$. Here time t is discrete, and position $x(t)$ is assumed to take only finitely many values (a so-called piece-wise constant approximation), so $x(t)$ can be seen as an index taking values in the set of integers $\mathcal{X}$ [345] and hence $V(t,x)$ can be interpreted as a stack of *dynamic programming tableaus* indexed by (t,x) (cf. Sec. IV D).

Yet another important application is in optimization for discrete stochastic processes: as was mentioned above, continuous time viscosity solution V of Eq. (17) is connected to stochastic optimal control for Markov diffusion processes via regularization. Hence it is not surprising that the discrete-time HJB in Eq. (19) could be connected to discrete stochastic processes too: namely, if instead of deterministic transitions $x(t+1) = x(t) + f(x(t), u(t))$, which are singular in terms of probability distributions, one introduces Markov Decision Process (MDPs) [347], which jumps from the state $x(t) = x$ to state $x' = x(t+1)$ with known probability $P(x'|x, u)$ provided the action u was applied, then Eq. (19) transforms into the famous backwards induction for episodic MDPs:

$$V(t,x) = \min_{u \in \mathcal{U}}\left\{\ell(x,u) + \sum_{x' \in \mathcal{X}} P(x'|x,u)V(t+1, x'))\right\}, \quad (20)$$

with a final time condition $V(T,x) = h(x)$, [347]. MDPs are central object of the Reinforcement learning (RL), a family of algorithms in which an agent aims to learn optimal decisions in unknown environments through the experience of taking actions and observing the rewards gained. In some cases, the environment is not influenced by the actions of the RL agent, in which case P is independent of action u then MDP turns into the uncontrolled MDP [348, 349], and if the transition matrix $P(x'|x)$ is of rank 1 such an MDP becomes the so-

called contextual multi-armed bandit and in this case finding V amounts to solving the optimization problem $\min_{u \in \mathcal{U}} \ell(x, u)$ for each x.

The key problem associated with HJB equation, continuous or discrete, is the *curse of dimensionality*. Solving HJB in Eq. (17) by conventional methods of numerical analysis quickly becomes intractable even in rather modest dimensions, e.g., if $x(t) \in \mathbb{R}^n$ for $n \geq 10$. In other words, direct approach to solving HJB equation has slim chances of succeeding. In what follows we discuss potential quantum optimization approaches to find global solutions to the three aforementioned problems:

1. Contextual multi-armed bandit problems.

2. Markov decision processes (MDPs).

3. Optimization problems constrained by differential equations.

Finding ways of solving the above problems will in turn suggest ways of solving the corresponding HJB equations, continuous or discrete. In fact, solving the discrete HJB in Eq. (19) sheds some light back on the solution of the continuous one and vice-versa, suggesting that algorithms developed in the discrete setting can help construct solutions of continuous optimization problems. Potential algorithms include:

1. Value iteration and policy iteration boil down to optimization problems, wherein the convex optimization of Sec. IV B 1 could be applicable, perhaps combined with the quadratic speedup of approaches of Sec. IV D.

2. Reinforcement learning is a broad family of approaches, rather than a single algorithm, which approximate value or policy iteration, for instance using Least-Squares Temporal-Difference (LSTD). Again, the convex optimization of Sec. IV B 1 could be applicable, perhaps together with the tabular method of Sec. IV D providing additional speedup.

3. Quantum reinforcement learning (heuristics). In the unitary oracle setting, from the perspective of exact methods, that is, algorithms that are guaranteed to identify optimal policies in arbitrary environment, tight quantum upper and lower bounds have been identified. In Refs. [350, 351], lower bounds on the more special task of Multi-armed bandit problems were found, along with essentially matching quantum upper bounds, for a number of RL-related tasks, including finding the optimal q-function. These works achieve a quadratic speedup over best possible classical methods, assuming appropriate oracle access to the task environment.

We can break the above into two main classes of approaches, differing in whether the learning agent has a standard classical access to the environment, or if the interaction itself is quantum. In the latter line, numerous

types of quantum oracular access to the task environment have been explored, e.g., where coherent access to a unitary encoding the transition function is possible [352], or where the environment closely mimics classical RL settings, and allows only actions as inputs and has memory [353].

We note that the algorithms listed above have significant limitations, including the fact that they require fault tolerance, and that quadratic speedups may not translate to wall-clock speedups in early fault tolerant quantum computers. The main conceptual limitation is the requirement to a coherent access to the task environment, which could only be possible in very restricted or constructed cases, and the fact they deal with exact solutions, and in real-world problems, the search spaces are simply too large, such as the state space of all 1 megapixel images. The up-side of the above results is that the speedups concern the number of interactions with the environment, and are not contingent on any assumptions in complexity theory.

In practice, one often foregoes optimal policies for optima within a parameterized policy family, as is found by policy iteration, and policy gradient algorithms. Although the method cannot find an optimal policy in general, one can still be interested in finding optima with respect to the function approximation family available. In Jerbi *et al.* [354] it was shown that quadratic speedups in policy optimization were possible as long as the policies satisfy certain regularity conditions, which are, e.g., satisfied for policies defined by parameterized quantum circuits. The lower bounds for this problem have been established for a more general setting by Cornelissen *et al.* [355] and they show that these quadratic improvements cannot be improved further. However, if the environments are not arbitrary, but special, more significant improvements are possible, including provable exponential speedups [356]. The contrived nature of these environments makes these results unlikely to have implications in practice.

However, the second research line investigates quantum enhancements in reinforcement learning when the access to the task environment is restricted to be classical. Here too speedups are provably possible, at least in special cases, but now subject to standard complexity assumptions. For example in Refs. [357, 358] it was proven that there exist task environments with a super-polynomial advantage for quantum reinforcement learning algorithms (with respect to finding optimal policies and optimal q-functions), unless the discrete logarithm problem has an efficient classical solution. Again, in reality, we are most often not interested in finding provably optimal policies, but rather more effective heuristic solutions which perform well empirically, as is done in the famous examples of AlphaGo [359] and AlphaStar [360]. In this domain, there has been substantial effort in defining various types of parameterized-circuit based QRL algorithms [357, 361, 362].

Closely related to MDPs are optimal stopping problems. These problems are concerned with timing an action for maximizing a reward and are often solved with dynamic programming. Optimal stopping appears, for example, in sequential parameter estimation [363], sequential hypothesis testing [364], and, most famously, in financial derivative pricing [365]. The pricing of American options involves stochastic modeling of asset prices and performing a combination of Monte-Carlo estimation, least-squares regression, and dynamic programming for the optimal stopping time [365]. Given quantum access to a sampler for the stochastic process, a quantum algorithm with a polynomial speedup using Quantum Amplitude Estimation (QAE) [169, 366] was shown by Doriguello *et al.* [367]. Key questions are whether we can extend such stopping time algorithms to other problem classes and to problems with more decisions, and whether we can make the algorithms more near-term quantum friendly.

In the following, we outline open questions and promising directions to further advance our understanding on how quantum computers may provide advantages for optimal control and related problems. In the era of noisy quantum computers, convex optimization algorithms may be hard to use directly. Nevertheless, one may considering sketching the value iteration [368] or (LSTD) [369], wherein both the random projections and the convex optimization could be implemented quantumly. What MDPs could have a bound on the radius of the ball outscribed to the (primal, dual) feasible set? Such MDPs could benefit from speedup of quantum algorithms based on MWU, cf. Sec. IV B 1. Approximating the value function, discrete or continuous, using the concept of inductive bias recently introduced in the context of function approximation by means of quantum kernels [370]. Basic idea is to embed a prior knowledge of certain properties of the value function into the design of the approximator: for example, a rather efficient deep learning approximation of value function implemented in AlphaGo, or an assumed linear approximation of the value function implemented in LinUCB [371]. In this direction one would benefit from maintaining connections between discrete and continuous HJB equations for the optimization problem at hand in order to extract inductive bias, e.g. piece-vise smoothness structure of continuous value function might help in designing a kernel-based function approximation.

F.   Robust Optimization

Robust optimization addresses the issue of data inaccuracy in optimization and allows one to model uncertainty in the constraints. The convex variant was introduced by Ben-Tal and Nemirovski [372] and lead to many follow-up works, for example Refs. [373–375]. The main feature is a set of parameterized constraints and the requirement that these constraints hold for every choice of parameters chosen from some uncertainty set. The problem is

formally defined as

$$\begin{aligned}
\min_{x \in \mathcal{D}} \quad & f_0(x) \\
\text{s.t.} \quad & f_i(x, u_i) \le 0, \, \forall u_i \in \mathcal{U}, \, i \in [m],
\end{aligned} \tag{21}$$

where $f_0, f_1, \ldots, f_m$ are convex functions in the parameter x and $f_1, \ldots, f_m$ are concave in the noise parameter u. Moreover, $\mathcal{D} \subseteq \mathbb{R}^n$ and the uncertainty set $\mathcal{U} \subseteq \mathbb{R}^d$ are both convex. Despite many advances, robust optimization for large-scale problems comes with a significant computational overhead. Ben-Tal *et al.* [376] developed meta-algorithms to approximately solve the robust version of a given optimization problem using an oracle for the solving of the original optimization problem. A quantum version was developed by Lim *et al.* [377], using quantum access to the functions, extending the original algorithm to include a stochastic gradient, and using quantum sampling subroutines with polynomial advantage.

G.   Multi-Objective Optimization

The previous classes were all considering a single objective function, possibly with constraints. However, in practice, decision makers often have to find trade-offs between different objectives. This is addressed by Multi-Objective Optimization (MOO) [378]. MOO is defined as

$$\min_{x \in X} \quad (f_1(x), \ldots, f_m(x)), \tag{22}$$

where X denotes the feasible set and $f_i : X \to \mathbb{R}$, $i = 1, \ldots, m$, denote the multiple objective functions, for instance. In this setting, we first need to define what we consider an optimal solution. This is done using the concept of *Pareto optimality*. A solution x is called Pareto optimal for a problem with multiple objective functions, if there exists one objective function f_i, that cannot be decreased any further without increasing at least one other objective function f_j. The set of all Pareto optimal solutions is called the Pareto frontier or *efficient frontier*.

It is well known that the cardinality of the Pareto set can be substantial [379], and thus, one would like to obtain a PTAS. For certain bi-objective problems, such as the shortest paths problem, Diakonikolas and Yannakakis [380] have shown that computing a set of solutions that approximates the Pareto curve within ϵ is **NP**-hard, if the set is to be smaller than twice the optimum number of solutions, while it is possible to obtain the 2-approximation in polynomial time.

Depending on the application, there are different goals in MOO. Either, the goal is to determine or approximate the complete Pareto front to be able to analyze possible trade-offs. Alternatively, a single Pareto-optimal solution is sought. For instance, objectives might be prioritized

or have fixed weights, which allows one to convert MOO into a (series of) single objective optimization problems. If the full Pareto frontier is required, the number of objective functions plays a crucial role. For a small number of objective functions, there exist efficient strategies, assuming the underlying single objective optimization problems can be solved reasonably well. This is usually achieved by constructing a new objective function as a weighted sum of the individual objective functions and adjusting the weights based on the previous solutions. Alternatively, a single objective can be optimized while the values of the other objectives are controlled via additional constraints. While the former might sometimes be easier to implement, it only finds the convex hull of the efficient frontier. In contrast, the latter can find also Pareto efficient solutions that do not lie on the convex hull. Neither of these strategies is efficient for growing numbers of objective functions. If the number of objective functions increases, MOO problems quickly become very hard.

In contrast to single-objective optimization, there is no approximation ratio that can be used to conveniently quantify the performance of a given solution candidate. However, there are multiple other metrics that allow one to compare different approximations to the efficient frontier. The most commonly used ones are the Hyper Volume (HV) and the Generational Distance (GD) [381]. The HV measures the volume spanned by the found solutions, which is measured as the total volume covered by all boxes defined by a reference point upper bounding (or lower bounding in case of maximization) all objective functions, and the solution candidates. The GD, in contrast, measures the distance between a set of candidates and the optimal Pareto front or a reference solution.

There is only very little literature considering quantum optimization algorithms for MOO. Most of the proposals apply classical strategies and use quantum optimization sub-routines for the resulting single objective optimization problems [382] or sample from quantum states to approximate the efficient frontier [383]. Thus, quantum-native algorithms for MOO present a very interesting domain for future research.

Within this section, we introduced the most important classes of optimization problems, reviewed existing quantum algorithms to approach them and key open questions to be answered to further advance our understanding. In the following, we discuss how to execute them on noisy quantum devices.

V.   EXECUTION ON NOISY DIGITAL HARDWARE AT SCALE

We now discuss best practices to execute quantum optimization on noisy quantum hardware. There are many different platforms to consider for optimization tasks [384–390], here, we focus on superconducting qubits [391]. More precisely, we focus on universal gate-based quantum computing platforms accessible via

the cloud, where quantum algorithms researchers can directly test their algorithms. While there are additional algorithmic challenges to scale to large problems, cf. Sec. IV, the goal of this section is to discuss how to achieve best possible results from available quantum devices.

Superconducting qubits come in many forms, such as frequency-tunable qubits [392] and fixed-frequency qubits with static [393] or tunable coupling elements [394, 395]. Over the past years, superconducting qubits have greatly improved. The number of qubits has increased with system sizes featuring up to 433 qubits. Coherence times have breached the 100 μs barrier with recent devices exhibiting median T_1 and T_2 times approaching 300 μs and 150 μs, respectively. It should be noted that these figures are based on IBM Quantum Eagle devices with 127 qubits [396]. Importantly, the qubits are arranged in planar lattices where each qubit is only connected to its immediate neighbors.

We review each stage of the execution pipeline with scalability and best practices in mind. In Sec. V A, we describe how a quantum algorithm is translated down the quantum stack and executed on hardware. We discuss the impact of modeling, quantum algorithm choice and qubit encoding in Sec. V A 1. Next, Sec. V A 2 discusses transpilation and Sec. V A 3 reviews error mitigation methods in the context of optimization. Sec. V A 4 touches on the pulse-level and the opportunities it presents. In Sec. V B we discuss hardware benchmarks and then briefly discuss an execution example in Sec. V C before summarizing in Sec. V D.

A. Quantum Stack

Hardware constraints and noise limit the scalability and performance of quantum optimization algorithms. An analysis of the quantum stack, the stages shown in Fig. 5 through which an algorithm evolves from its abstract design to its tangible execution on a quantum processor, offers insight into where scalability and performance bottlenecks arise.

At the top of the stack is an abstract expression of the quantum algorithm. It is usually free from any hardware constraints and expressed as mathematical operations and logical constructs that embody the problem-solving approach. The transpilation process then maps the abstract operations to the machine-executable instructions of the chosen target hardware resulting in a hardware-native circuit. Here, circuit instructions may be inserted or modified as part of an error mitigation scheme. The pulse-level compiler converts the circuit instructions to finely-tuned electromagnetic pulses. These pulses manipulate the quantum states to enact the chosen algorithm. Therefore, as the algorithm is lowered through the layers of the quantum stack its representation changes. Each level offers optimization opportunities to extract the most out of the noisy hardware.
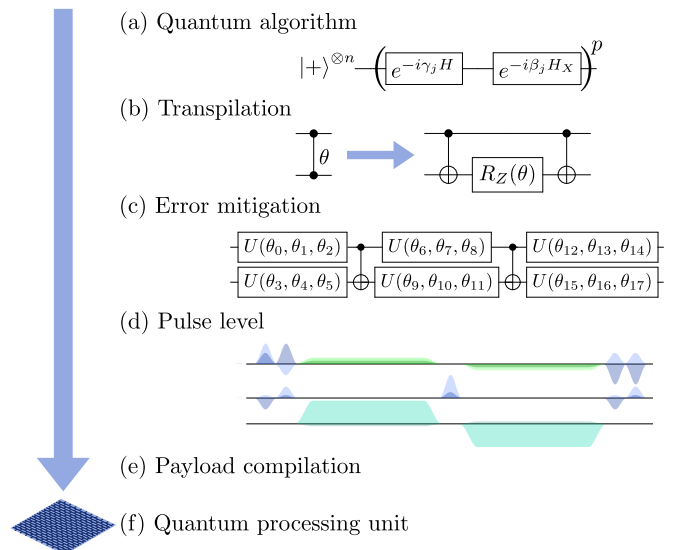

Figure 5. High-level summary of the quantum stack. (a) A quantum algorithm, including an encoding of decision variables in qubits and any hardware-guided problem simplifications, results in a high-level quantum circuit. (b) The instructions in this circuit are then transpiled to hardware native instructions. (c) Error mitigation methods, such as PEC, are often encoded as circuit instructions. (d) At the pulse-level, circuit instructions are represented by the physical pulses that are played on the qubits. (e) Finally, the circuit is compiled into machine-executable waveforms, (f) that are run on the quantum processor.

1. Modeling, quantum algorithms, and encoding

At the top of the stack, an algorithm is described by a high-level circuit which may not be explicit in the number of qubits used and only provides an abstract representation of the quantum gates to run. For example, Fig. 5(a) shows a high-level representation of the QAOA. To arrive at such a circuit one must first appropriately model the optimization problem to solve and correspondingly choose the quantum algorithm. For instance, linear equality constraints can be included in a QUBO by adding them as quadratic penalty terms which then requires appropriately sizing the corresponding multiplicative factor, see Sec. IV A 2. This modeling step also presents opportunities to consider the impact of choices on the lower levels of the quantum stack. For example, due to hardware noise a problem model that results in more qubits with less connectivity could lower the impact of noise compared to a model with fewer densely-connected variables. Alternatively, Dupont *et al.* [229] project a fully-connected QUBO onto the sparse connectivity of a quantum computer. Looking forward, additional research on quantum tailored modeling of optimization problems is needed to find the right balance between model approximations and noise susceptibility.

Since the hardware is noisy, algorithms that efficiently use the quantum resources are required. For exam-

ple, a warm-start reduces the number of QAOA layers [197, 198]. In addition, the QAOA mixer can be changed to include constraints of the optimization problem [258, 397], e.g., for equality constraints [259, 260]. Altering the mixer improves the approximation ratio compared to penalty term constraints (soft constraints) for constrained optimization problems [398]. Furthermore, counter diabatic terms may also improve the approximation ratio [399]. While these mixers may keep the structure of the objective function and lead to cheaper implementations of the cost operators, they may increase mixer circuit depth thereby limiting their near-term utility [398, 400]. Crucially, from a hardware perspective, the best practice is to choose the methods that limit low-level circuit depth and preserve solution quality. For example, adaptive bias-fields [401] and warm-starts [197] only require additional single-qubit rotations in the QAOA mixer which are implemented on hardware with a negligible overhead. Furthermore, it may be useful to research methods that modify the original optimization problem to facilitate its mapping to the hardware. For example, in QAOA one may investigate techniques that modify the problem Hamiltonian H to produce an H' approximating H which is however easier to implement on hardware, i.e., $e^{-i\gamma H'}$ has a lower circuit depth than $e^{-i\gamma H}$.

While both scale and quality are improving over time, quality is typically the limiting factor rather than scale for optimization [402–404]. Nevertheless, for many practically relevant optimization problems the scale of current Quantum Processing Units (QPUs) is likely insufficient to reach a classically hard regime. It may therefore be necessary to consider denser encoding schemes that encode multiple decision variables in a qubit. The Holevo bound states that n qubits are required to faithfully transmit n bits of information [405]. However, encoding schemes such as Quantum Random Access Codes (QRAC) can encode m bits of information in n qubits where $m > n$. The cost is that each bit of information can only be retrieved using Projective Operator Valued Measures (POVMs) with some probability $p > 1/2$ [406–409]. There are many QRACs that, rather than encoding the two states of a single bit into the $|0\rangle$ and $|1\rangle$ states of a qubit, encode more information by utilizing more of the possible states on the Bloch sphere. For example, the $(2, 1, 0.85)$ QRAC encodes $n = 2$ bits into a single qubit ($m = 1$) with the 4 possible states of 2 bits encoded in states that form the corners of a square on the Bloch sphere. Here, each bit is recovered with probability $p \geq 0.85$. For all QRACs, there is a trade-off between the ratio of classical to quantum bits and the retrieval probability. The more bits encoded into a given number of qubits, the lower the probability p of being able to retrieve a given bit and the more complicated the code becomes. As discussed in Sec. IV A 1, this allows the hardware to tackle problems with more decision variables than it has qubits [180, 237, 238, 410]. However, denser encodings tend to have a lower approximation ratio and introduce additional overhead to readout solution candidates.

Research for quantum optimization algorithms running on noisy hardware must carefully consider the hardware constraints. Crucially, many of these methods are heuristic and can thus only be properly researched at scale on actual quantum hardware.

2. Transpilation

A transpiler transforms a quantum circuit into another quantum circuit, typically built from instructions native to the target quantum hardware. Such low-level circuits can be expressed in domain specific languages which include OpenQASM [411], Quil [412], or Blackbird for continuous-variable quantum computers [413]. Since each quantum hardware bears unique characteristics, tailoring the transpiler to individual architectures is fundamental to maximize operational efficacy. Furthermore, the design and optimization of hardware-executable quantum circuits must include all constraints of the target hardware such as the native gate-set and the qubit connectivity. On noisy hardware the transpiler must also account for properties such as gate error-rates, cross-talk, and qubit properties, e.g., T_1 and T_2 times. The transpiler has a large impact on the gate count and, thus, the performance of the algorithm when run on the hardware. In general, the transpiler is responsible for (i) selecting the qubits, (ii) decomposing the high-level circuit into hardware native instructions and (iii) routing the qubits to overcome any limited qubit connectivity.

Step (i) depends on the properties of the hardware and the problem being solved. For example, if we do not need all the qubits on the processor then we may select the best ones according to a metric [414], such as gate fidelity [403], cross-talk [415], or layer fidelity [416]. This metric should be based on the properties of the expected circuit. E.g., if the expected circuit is predominantly composed of two-qubit gates then selecting qubits based on their gate fidelity is reasonable [403]. However, if the algorithm requires many mid-circuit measurements then measurement quality should also impact qubit selection [417]. Furthermore, in approximate optimization a decision variable is often encoded in a qubit. The choice of which decision variable is mapped to which qubit significantly impacts circuit depth [418].

In step (ii), the possibly high-level gate descriptions are synthesized into hardware native instructions [419–421]. To decompose arbitrary unitaries, a quantum computer must support a universal set of basis gates [422], which impacts the optimal synthesis technique. These compilation methods are either deterministic or approximate. For example, any exponential of Pauli operators can be implemented using a local 2-qubit gates on the involved qubits with a Pauli-Z rotation and single-qubit Cliffords [423], as exemplified in Fig. 5(b). However, due to hardware noise it might be beneficial to reduce the circuit depth by allowing for an approximate compila-

tion [424, 425].

Finally, the routing step (iii) must overcome any limited qubit connectivity of the target QPU. Here, general-purpose heuristic algorithms exist to route arbitrary circuits [426–428]. However, certain algorithms may have a predetermined structure. For example, QAOA results in circuits with blocks of commuting two-qubit gates. Exploiting this structure results in shallower quantum circuits [402, 418, 429–431]. More precisely, if the topology of the problem graph does not perfectly match the coupling map of the QPU, SWAP gates are necessary to enable two qubit gates between unconnected qubits. The denser the graph, the more SWAP gates are necessary which results in a deeper circuit [402]. Nevertheless, limited qubit connectivity remains an issue and the problems that scale best on noisy superconducting devices are therefore likely to be those that are not overly dense [386, 403]. A key challenge is thus to identify problem instances that are not dense but are classically hard.

Modern hardware also exposes new circuit instructions such as dynamic circuits [432]. In a dynamic circuit some of the qubits are measured and the outcome of these measurements changes the following circuit instructions within the coherence time of the qubits. For instance, this offers additional possibilities to address limited qubit connectivity [433]. As an example, the SWAP based implementation of the QAOA problem graph may be replaced on grid-like QPU coupling maps by constant depth circuits with a quadratic auxiliary qubit overhead [434]. Additional research is needed to see how to leverage dynamic circuits to reduce the circuit depth and width.

Crucially, not all types of hardware face the same challenges. For example, trapped ion devices typically have all-to-all connectivity which permits denser graphs to be studied without increasing circuit depth [398]. However, at the time of writing trapped ions have not reached the same scale as superconducting devices meaning problems sizes are limited.

3.   Error suppression and mitigation

After transpilation, the quantum circuit may be modified to include additional instruction for error suppression and mitigation, as shown in Fig. 5(c). Error suppression methods reduce the noise in quantum circuits without having to gather more shots. Most prominently, Dynamical Decoupling (DD) suppresses non-Markovian errors by adding pulses in idle times of a quantum circuit [435–437]. This can improve circuit execution quality without introducing additional overhead. In some cases, such as QAOA transpiled with SWAP networks [403], the resulting quantum circuits may be dense which leaves little space for DD pulse sequences. By contrast, DD has a large impact when the circuit has many idle times, e.g., in hardware native circuits on a heavy-hex lattice [438].

Error mitigation techniques use pre- and post-processing of quantum circuits and measured results to reduce the impact of hardware noise. While fully eliminating the error comes at an exponential classical overhead, these methods extend the reach of current devices even at a limited classical cost [14, 439, 440]. This is in contrast with error correction, which aims to correct errors by encoding logical qubits in vast arrays of physical qubits [441]. Typically, most error mitigation methods, such as Probabilistic Error Cancellation (PEC) [15] and Zero-Noise Extrapolation (ZNE) [16, 439, 442], produce noise-mitigated expectation values.

PEC learns the noise channel Λ of layers of Pauli-twirled gates [15]. The inverse of the noise channel Λ^{-1} is usually not physical, but can be implemented by quasi-probability decomposition. PEC thus produces an unbiased estimate of a noiseless observable O. The cost is an increase by a factor of γ^2 in the measured variance. Here, $\gamma \geq 1$ depends on the strength of the terms in the learned noise model and is a measure of the fidelity the quantum circuit can be executed with. The noise strength γ is also closely related to the Layer Fidelity (LF) and the Error Per Layered Gate (EPLG), two related measures of noise in large-scale quantum devices [416]. If $\bar{\gamma}$ is the average γ per qubit then a circuit with n qubits and d layers will have an error mitigation cost that scales as $\bar{\gamma}^{2nd}$.

In ZNE multiple logically equivalent copies of a quantum circuit are executed at different noise levels to compute an expectation value. From the noisy expectation values one extrapolates to the zero-noise limit to obtain a (usually biased) error mitigated expectation value. The first proposals implemented ZNE by stretching pulses [439], requiring a large calibration overhead, or by folding gates, resulting in large stretch factors. Methods such as partial-gate folding [443] or stretching of cross-resonance pulses [444] alleviates some of these issues. Crucially, ZNE does not require more circuits as the size of the problem is increased. However, the circuit depth limits the range of stretch factors which may render the extrapolation unusable. PEC is unfeasible in large scale-experiments since they have a very large γ. For instance, the γ^2 of 60 layers of CNOTs on 127 qubits was 10^{128} in Kim *et al.* [16]. This prompted the development of PEA [16], a probabilistic ZNE, in which the noise is learned like in PEC but then amplified (instead of canceled) to then extrapolate to the zero-noise limit. This may also reduce the bias in the extrapolation. Importantly, circuits designed with regular networks of SWAP gates alternate a small number (e.g., two) of identical layers of CNOT gates [402]. This minimizes the noise-learning overhead of PEC and PEA.

Error mitigation methods that act on expectation values help train variational parameters [403, 439]. Crucially, to find good solutions, quantum (approximate) optimization requires sampling from the optimized circuit. Indeed, the task is to find a good sample x that, e.g., minimizes a cost-function $f(x)$. Therefore, developing new error mitigation methods to produce error-mitigated samples is crucial for the success of quantum optimization.

4.   Pulse-level and compiler

Ultimately quantum instructions are implemented by physical pulses, scheduled in time, to manipulate the quantum states [445], as exemplified in Fig. 5(d). The pulse-level makes the differences between quantum architectures more apparent. For instance, tunable couplers [395] can natively implement exchange-type operations which, on cross-resonance hardware [393], are synthesized from R_{ZX} rotations. Exchange-type gates may help reduce schedule duration for applications like QAOA that need SWAP networks with phases to overcome limited device connectivity.

Pulse control enables many different types of optimizations to reduce schedule durations. This may be as simple as a pulse-efficient transpilation in which variational parameters are encoded in durations of cross-resonance pulses [446]. The resulting reduction in schedule duration improves the sampling performance of QAOA [403]. Furthermore, the extra energy levels of the transmon can help to engineer gates that would otherwise require many CNOTs [447, 448]. Finally, advanced methods of quantum optimal control can help synthesize gates [449, 450], measurements [449, 451], and adiabaticity [120] which may be useful in the context of approximate quantum optimization. However, such methods are not always compatible with quantum error mitigation as discussed by Egger *et al.* [452].

The pulse-level is also responsible for quantum program compilation which entails loading the pulse-schedule into waveform memory. Care must be taken at this step and inefficiencies can be costly to the total run time.

B.   Hardware Benchmarks

In general, the performance of quantum hardware is measured by scale, quality, and speed [453]. Scale is simply the number of qubits. Quality is tracked by low-level performance metrics, e.g., qubit coherence and gate fidelity, and holistic benchmarks such as Layer Fidelity and Error per Layered Gate [416], Quantum Volume (QV) [454] and the γ and $\bar{\gamma}$ of PEC [15]. A QPU with a 2^n QV can reliably execute a n qubit circuit with n layers of random SU(4) gates [454]. The QV accounts for gate error rates, error suppression and mitigation, and qubit connectivity.

Neither QV, Layer Fidelity nor EPLG, however, account for the speed of the computation. Speed is crucial for optimization tasks if the quantum hardware is supposed to deliver a solution with a certain quality *faster* than classical hardware. Furthermore, speed enables the hardware to achieve more accurate estimations of observables within a certain time or to pay the cost of error mitigation which often requires more shots. Therefore, Circuit Layer Operations Per Second (CLOPS) complement quality metrics. The CLOPS metric is inspired

from the floating point operations per second metric used to benchmark classical computers. In essence, CLOPS quantifies the throughput of a device by measuring the number of gate layers that can be executed within a certain time. This measure includes the classical overhead of updating parameters in the circuit and is therefore a crucial indicator for the run time of variational workloads.

QV, EPLG, and CLOPS are by design only proxies for the utility of a quantum device. For example, EPLG and CLOPS on *ibm_sherbrooke* are 1.7% and 2700, respectively. These metrics are general enough to draw comparisons between different hardware architectures and track improvements of a platform. However, these general metrics could fail to predict which quantum device will obtain the best performance, be it in solution quality, execution speed, or any other criterion, for a specific application. Therefore, the best possible measure of a quantum computer's usefulness for optimization is how well it performs on tasks similar to the problem of interest, according to the performance criterion of choice. For example, the ability to create spin-entangled states, i.e., squeezing, relates directly to a QAOA solving a fully connected MAX-CUT problem [455]. We foresee that more application tailored benchmarks are needed, especially since many algorithms are heuristic in nature.

C.   Execution Example

We now exemplify some of the considerations discussed in the previous sections by examining the work by Sack and Egger [403]. Here, the authors executed depth-two QAOA on hardware with Random-Three-Regular (R3R) graphs with up to forty nodes on superconducting qubit processors with 127 qubits. First, since only a third of the physical qubits are needed, the QAOA is run on the best qubits as measured by the two-qubit gate fidelity; the most used and error-prone circuit instruction. Second, the R3R connectivity is engineered with a network of predetermined SWAP gates [386, 402]. This results in dense and shallow quantum circuits that transpile quickly [402]. Third, Sack and Egger [403] also choose a decision-variable-to-physical-qubit mapping that minimizes the number of SWAP gates following the approach of Matsuo *et al.* [418]. These best practices for QAOA are found open source on GitHub [456].

Finally, a machine-learning-based error mitigation of expectation values allowed the optimization of the QAOA variational parameters resulting in $\gamma^\star$ and $\beta^\star$. A noiseless simulation of the expected mean of the samples drawn with $\gamma^\star$ and $\beta^\star$ showed a large gap with respect to samples drawn from the hardware. This highlights the need for an error mitigation of samples for quantum approximate optimization.

D. Summary

Sec. V provides an overview of the crucial parts of the software stack affecting the execution of quantum algorithms at scale on noisy hardware. When considering this stack, it becomes evident that users possess varying degrees of control over the scalability of quantum algorithms on noisy hardware. At the higher levels, such as problem recasting, encoding, and high-level circuit design, users enjoy a favorable degree of control (also depending on the problem being considered), allowing them to shape the algorithm's scalability to an extent. Best practices for noisy hardware are to choose algorithms and their variants that lower the circuit depth without compromising on algorithmic speed and quality.

Moving lower into the quantum stack, transpilation and pulse-level control also offer the opportunity for optimization. SWAP networks, SAT mappings, pulse-efficient transpilation, and problem-tailored gate-sets all help reduce circuit depth and duration. These lower-level aspects demand a profound understanding of quantum mechanics and hardware intricacies. In some cases, these layers may even conceal these complexities behind a primitive such as a sampler or an estimator. It is nevertheless crucial to have transparency on the methods employed. This also enables the design of hardware-optimized algorithms and problem formulations, which may further improve the performance of executing quantum optimization algorithms on noisy hardware.

VI. BENCHMARKS

Benchmarking can provide insights into realistic algorithmic performance when purely analytical methods fail, and it can provide new perspectives that motivate deeper analytical studies. In particular, theoretical complexity analysis typically pertains to general problem classes and considers mostly asymptotic scaling behavior or worst case scenarios. In fact, theoretical worst case bounds, as they are discussed in Sec. II, often fail to describe the difficulties of solving real-world applications that focus on a specific problem instance of practical use, nor do they faithfully describe how well practically relevant problems can be solved for fixed size and data. In this case, we need to rely on meaningful and systematic benchmarking of optimization algorithms with well-defined assumptions considering implementation, data, and metrics — a challenging task. Clear benchmarks that study the utility and robustness of optimization algorithms can also serve to provide trustworthy and verifiable performance indicators for a broader audiences, including policy makers and industry leaders. For a general introduction to computational benchmarking, we refer the interested reader to Refs. [457–462].

The aim of this section is to describe best practices for benchmarking and provide a set of interesting benchmarking problems that provide a thorough, robust, fair,

and reproducible setting for comprehensive comparisons of different classical and quantum approaches. The remainder of this section is structured as follows. First, an overview of related work is given in Sec. VI A. Next, Sec. VI B discusses various reasons for running a benchmark. Notably, the goal behind running a benchmark is crucial for the definition of the setting. Benchmark design choices such as the metrics to be evaluated are presented in Sec. VI C. Then, we elaborate on a variety of optimization problems which are difficult for state-of-the-art solvers and may provide interesting benchmarking problems for quantum methods in Sec. VI D. Finally, Sec. VI E discusses existing quantum optimization demonstrations that may build up to a set of benchmarks.

A. Related Work

Important collections of existing (classical) optimization benchmarks are publicly hosted benchmarking challenges. They have proven as valuable tools for fair and insightful comparisons between different classical solvers and/or platforms. These challenges aim at understanding and improving the practical performance of algorithms for various optimization problems, particularly those that are theoretically hard. They aid in evaluating realistic algorithm performance conditioned on the resources available at the time of the competition. Some of the most relevant, regularly held optimization challenges tackling problems with classical digital computers are the Discrete Mathematics and Theoretical Computer Science (DIMACS) implementation challenges [463] and the Satisfiability (SAT) competition [464]. While the problems targeted in the former vary from competition to competition, the latter always focuses on SAT problems which represent a well-defined class of problems that, beyond their theoretical intrigue, find practical applications across various fields [465, 466]. Another important resource for optimization benchmarks is the Mixed Integer Programming Library (MIPLIB) [467]. This library provides regularly updated pure and mixed integer programs that are chosen under the consideration of solvability and numerical stability. Currently, it holds 240 benchmarking instances that are solvable with existing methods and an even larger collection of unsolved, and numerically difficult instances. The performance of many solvers regarding several classes of optimization problems is regularly compared in [468].

Optimization competitions and libraries of this form that are compatible with (near-term) quantum computers do not exist as of now. It remains an open task to set them up and, thereby, enable streamlined benchmarking of quantum optimization methods. In fact, most existing related work focuses on application-centric quantum hardware benchmarking frameworks that employ optimization problems as exemplary application. A benchmarking framework presented by Lubinski *et al.* [469] suggests the comparison of the output quality to circuit

width and depth for implementations of Shor's factoring algorithm, Grover search, Hamiltonian simulation, etc., executed in numerical simulations or gate-based quantum hardware. Another framework introduced by Tomesh *et al.* [470] measures the qubit connectivity, circuit depth, number of 2-qubit gates, ratio of gates which can be executed in parallel, number of times qubits are acted upon, and number of mid-circuit measurements for algorithms including QAOA, VQE, and Hamiltonian simulation algorithms executed on quantum hardware platforms from different providers. Furthermore, Martiel *et al.* [471] present a metric that measures the maximum problem size for which a problem instance can be solved given a pre-defined accuracy level. Notably, the suggested metric does not incorporate time as an explicit factor. Instead, the metric is conditioned on algorithms to be run in polynomial time. This directly excludes the use of error mitigation and can easily fail to provide a fair and meaningful comparison between different platforms. Further benchmarks that investigate different properties of quantum annealing methods for optimization problems – some including comparisons to classical and quantum algorithms – can be found in Refs. [472–479]. A comparison of Ising machines that checks success probabilities for finding the ground state and the corresponding time-to-solution is presented by Mohseni *et al.* [480].

One particularly interesting work is introduced by Finžgar *et al.* [481]. They compare various QUBO solvers with respect to the solution quality and time-to-solution and as such represents a good baseline for an optimization benchmark. Although the presented publications do not offer holistic optimization benchmarking frameworks, they provide interesting insights that can help to define reproducible, representative, and fair benchmark libraries, which facilitate identifying advantages and disadvantages of various (quantum) solvers.

B. Goals

Reasons for running a benchmark can be diverse. In the following section, we focus on benchmarking goals related to understanding the potential of quantum computers for solving optimization problems in terms of a fair comparison.

One of the main goals of benchmarking in the present context is the comparison of classical and quantum algorithms — aiming at demonstrating quantum advantage. If classical platforms are compared amongst themselves, one often relies on benchmarking problems such as SPEC [461] and LINPACK [462], which are established in the community. However, these problems are not well suited to compare classical and quantum platforms. Instead, the community will have to agree on a set of problem-centric benchmarks that could enable the direct comparison of run times, solution qualities, etc., of classical and quantum solvers. Suggestions for suitable benchmarking problems are given in Sec. VI D.

Benchmarking may also be executed to track the progress in hardware developments (cf. Sec. VI A) as well as algorithmic improvements. This is particularly important for monitoring the impact of the fast development in quantum hardware, supporting software, and algorithmic research. For an example of tracking progress of classical hardware and algorithms, we refer the interested reader to Koch *et al.* [459], which discusses the advancements in mathematical programming solvers over 20 years.

Furthermore, benchmarking can give us insights into an algorithm's scaling behavior: Is there a problem scale at which algorithms fail to provide good solutions — possibly due to hardware limitations? How do the resources needed to find solutions to a problem of a certain quality scale with the system size? An example of MIPLIB problem instances is illustrated in Fig. 6. In this case, many – but not all – larger instances require longer solver run times.
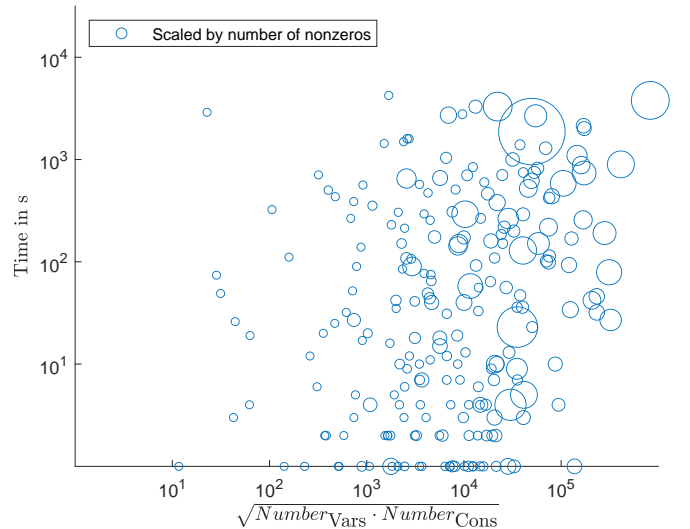

Figure 6. The figure illustrates the time it takes the Gurobi solver [482] to find the optimal solution for MIPLIB 2017 [467] instances of the form $\min_x c^T x$ subject to $Ax \leq b$, x being partly integer. The size of the circles depicts the number of non-zero coefficients in the constraint matrix A and the instances are ordered (on the x-axis) according to the square root of the number of variables times the number of constraints.

C. Design

To ensure reproducibility as well as replicability of a benchmark, it is important to clearly define the specifics in terms of model selection, pre-processing, computational platform, target metrics, and selected algorithm (provably exact, heuristic, etc.) including suitable hyperparameters. A discussion of relevant algorithmic paradigms and quantum algorithms for optimization problem classes can be found in Sec. III and Sec. IV,

respectively. The remaining levels of design choices are discussed in the following section. Starting from the decision whether the benchmark is defined with respect to a particular model, and, if so, which one. Secondly, it usually is beneficial to apply pre-processing to reduce the problem to its challenging core. A possibly applied pre-processing procedure may already consider the computational platform that is employed afterwards to execute the optimization algorithm. Finally, the algorithm execution and resulting outcomes can be captured in a comparable fashion by choosing appropriate benchmarking metrics.

1.   Model-(In)Dependence

At the core of a practically motivated optimization task is the solution of a concrete decision problem such as the selection of an optimal (or at least a good) tour for TSP. In order to treat this problem in a rigorous way a mathematical model must be developed. However, the model choice is usually not unique and may lead to differently behaving algorithms. In particular, certain models may be significantly better or worse suited for a particular optimization algorithm or may even contain characteristics that cannot be handled by a particular algorithm, such as non-linearities or constraints. For benchmarks, it is therefore crucial to decide on the level a problem is defined: either *model-independent*, i.e., on the level of the original optimization problem, which allows different mathematical modeling methods, or *model-dependent*, i.e., on the level of a concrete mathematical model.

Model-independent benchmarks that place few limitations upon admissible solution strategies offer an interesting method to identify quantum advantage for optimization. More specifically, quantum advantage may stem from novel formulations of problems that could not be captured on a model-dependent level. Furthermore, quantum hardware is quickly evolving. To enable comparability and compatibility for different generations of hardware, model-independent benchmarks are better suited, since they may allow one to leverage new features that might not have been available in earlier generations, such as dynamic quantum circuits [433]. We expect these benchmarks to eventually show how quantum optimization algorithms are approaching, meeting, and exceeding the capabilities of classical techniques for solving particular problem instances.

Nevertheless, model-dependent benchmarks are also a useful tool. In particular, they can help to study the relative performance of different algorithms and platforms for fixed model formulations. At this point, it should be noted that model-specific solvers are often significantly faster for the target problem than general solvers. For specific metrics to compare different types of solvers, we refer the interested reader to Berthold and Csizmadia [483]. Some model-dependent benchmarking test in-

stances that are openly available, established, and unsolved are given in MIPLIB [467], the Quadratic Programming Library (QPLIB) [484], and the Satisfiability (SAT) Problem Library [485]. As an example, we can consider (random) SAT instances which exhibit phase transitions. It may now be of particular interest to investigate whether a quantum algorithm is better suited to handle the difficulty of a phase transition. In fact, it has been studied by Yu *et al.* [486] whether an adaptive bias version of QAOA might help to avoid transition phenomena. While the presented study focuses on small problem instances, where classical algorithms can provide solutions, the findings hint at the potential for advantages in the realm of large problem densities and, hence, motivate further investigation.

2.   Pre-processing

The term *pre-processing* or *pre-solving* refers to a set of methods that are applied to a problem instance or data before the actual optimization algorithm is executed. Standard pre-processing approaches require classical computational effort. Therefore, the pre-processing should also be taken into account in a benchmark's performance evaluation.

The goal of pre-processing is to modify the problem in such a way that it allows the optimization algorithm to work more efficiently. How to implement pre-processing in practice depends on the mathematical formulation of the optimization problem, on the data, as well as on the considered optimization algorithm and computational platform. In fact, it is not uncommon for many problems such as Euclidean TSP or Steiner tree problems in graphs that pre-processing is able to remove 95% of the variables and, hence, reduce the problem size as well as resource requirements significantly [487, 488]. How this can be achieved often depends on the context and characteristics of the problem and data. A central aspect of pre-processing is the reduction of the search space, for example by decreasing the number of decision variables, or reducing the number of alternative optimal solutions, or by decomposing the problem into smaller subproblems.

Considering the example of QUBO, various reduction techniques to eliminate optimization variables have been suggested in the literature. A recent review has been provided by Rehfeldt *et al.* [488]. For example, a set of analytic reduction rules is presented by Glover *et al.* [489] that allows one to conditionally fix certain optimization variables, while guaranteeing that at least one optimal solution will remain in the search space. In addition to the number of optimization variables, their degree of correlation, which is determined by the sparsity of the QUBO matrix, is also of interest. In the case that certain groups of decision variables are not correlated, QUBOs can be decomposed trivially into smaller sub-QUBOs to be solved independently.
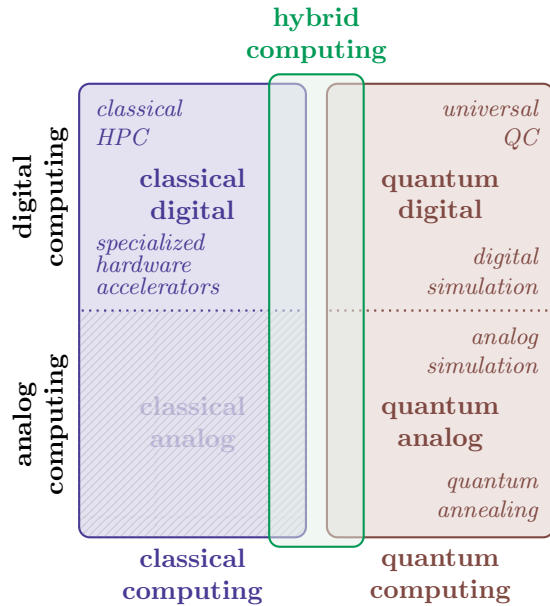
Figure 7. Sketch of the optimization platform ecosystem. We focus on digital classical computing as well as digital and analog quantum computing.

3. Platforms

Different platforms offer different strengths and suffer from different bottlenecks. Hence, some devices could be better suited for certain optimization problems than others. It may not be directly evident which hardware is best suited for a given optimization problem. Furthermore, depending on the platform, the optimization models and applicable algorithms do vary, resulting in better or worse performance for a given problem.

Tailoring the choice of platform to the characteristics of the problem at hand – if possible – can lead to better outcomes. This approach aligns with the idea of *platform-aware* optimization, ensuring that the selected platform is optimally matched to the unique requirements of the problem being addressed. To understand the advantages and disadvantages of different platforms for different optimization problems, performing cross-platform benchmarking is crucial. At the same time, such benchmarks pose a particular challenge, as they must allow a fair comparison between systems that can function very differently. In the following, we are going to elaborate on various hardware types that are relevant for optimization problems, i.e, classical vs. quantum, and analog vs. digital. A sketch of the platform ecosystem is shown in Fig. 7.

Classical digital systems are the workhorse of today's computer technology. The basic units of information are the binary values 0 and 1. The information in classical digital computations is processed by compositions of logical operations such as AND and NOT gates which can

be expressed in a circuit model. Digital classical computational models often rely on a combination of Central Processing Units (CPUs), which excel at sequential and general data processing tasks, and Graphics Processing Units (GPUs), which specialize in parallel processing tasks. There are several other types of processors and specialized hardware accelerators designed for specific tasks and applications, for example Tensor Processing Units (TPUs) for deep learning or Field-Programmable Gate Arrays (FPGAs) for signal processing and embedded systems, just to name two. The combination of CPU, GPU and special-purpose processors forms the foundation for classical high-performance computation in a wide range of applications. Due to their widespread use, classical digital hardware devices span a wide spectrum of computing platforms, including desktop computers, computing clusters, cloud services, mobile phones, and many more. All of these platforms can be used to solve optimization problems.

In additional, special purpose systems for optimization are developed. More specifically, these systems simulate quantum phenomena such as quantum annealing, see Sec. III B, hence, aiming to gradually decrease an energy function. Well-known examples for this type of technology are the Fujitsu Digital Annealer [490, 491] that is based on a hardware-accelerated Markov Chain Monte Carlo approach, the Hitachi CMOS Annealing Machine [492] which corresponds to an in-memory computing architecture that represents an Ising model with local interactions, and the Toshiba Simulated Bifurcation Machine [493, 494]. It should be noted that the latter does not approximate an annealing process but instead implements a highly parallelizable approximation of Ising dynamics with classical hardware (GPU, FPGA). This allows one to compute heuristic solutions to problems with up to millions of variables in a time frame of seconds.

In digital quantum computing [391] the basic units of information are *qubits* instead of bits. The information processing consists of consecutive quantum gate operations such as CNOT gates and single-qubit rotation gates which, together, form a quantum circuit, similar to a classical logical circuit. Important bottlenecks of existing digital quantum computers are limited coherence times, which result in limited circuit depths, limited qubit numbers, and limited qubit connectivity. There are a variety of ways to quantify the performance of a digital quantum computer. For near-term hardware where noise is a dominant factor, one might be tempted to consider metrics such as the coherence times of qubits or the fidelity with which various quantum operations can be performed. Such metrics are valuable, however, they can fail to give an accurate description of a device's potential for useful computation. This has motivated proposals for more holistic error metrics, as discussed in Sec. V B.

Quantum analog hardware is realized by quantum annealing [2, 3, 472, 495, 496] and quantum analog simulation [497–505]. In quantum annealing, the energy function or Hamiltonian of a system is gradually (but not nec-

essarily perfectly adiabatically) deformed from an initial ground state to a final state that can be measured to find the solution to an optimization problem. These systems are designed for QUBOs. Important bottlenecks of these platforms include the coherence time, restrictions on the driver Hamiltonian, and challenges in embedding practical problems into the device connectivity. The coherence time of current publicly available annealing machines is shorter than the annealing time [506]. This means that these platforms do not achieve closed system dynamics but exhibit strong coupling with the environment. Further hardware improvements are therefore required [507]. The primary computational resource is believed to be incoherent quantum tunneling events [508, 509]. Therefore, recent efforts are directed towards engineering quantum drivers that cannot be simulated classically [507, 510]. Quantum analog simulation describes a setup consisting of physical systems with sufficiently controllable parameters such that one can manipulate and probe the dynamics underlying a family of Hamiltonians. More specifically, analog quantum simulators enable the emulation of continuous quantum dynamics of a target system with a simulator that is easily accessible in the lab. This setting has proven itself valuable to simulate physical dynamics including quench problems, and ground state problems [497] but has not been studied in much detail as a platform for optimization applications, yet.

4. Metrics

Due to the technological differences between classical and quantum systems, the selection of a set of metrics that enable a fair comparison is crucial. The four main metric categories that suffice said criteria are *resource cost, run time, quality, and problem complexity.*

The resource cost includes any quantum or classical computational resources employed to solve a given optimization problem including any potential pre- and/or post-processing steps and resources that are being used in parallel. The respective costs may be quantified in various ways. Firstly, a fair metric would be the amount of energy that is used to solve a problem. However, in most cases, the user will not have access to this information, particularly, when specialized computational resources such as CPUs, GPUs, or quantum computers are accessed via the cloud. Similarly, the monetary cost of running an algorithm – including the pricing of specialized computational resources – could provide an interesting metric but will be tough to evaluate on a fair basis as these depend on hardware providers and service contracts. Classical computing resources are typically measured with respect to factors such as the used memory, required memory bandwidth, and central processing unit (CPU) specifications, including clock speed and the number of cores. Quantum computational resources can be measured in terms of the number of circuits that are run, the number of executed gates per circuit, the number of

qubits and shots per circuit execution, and the quality of the qubits. Of course, the resource count should also consider resources that are required for problem-dependent calibrations and error mitigation or – if applicable – correction. The number of qubits in a circuit strongly depends on the mapping of problem variables to qubits. Hence, algorithmic improvements in mapping schemes can lead to a reduction in qubit requirements.

The run time provides an objectively fair metric to compare different computational platforms. It is, hence, of utmost importance to clearly define what determines a *faster* algorithm. The effective run time of an algorithm consists of time used for pre-processing, transpilation, embedding, compilation, execution, and post-processing. Now, it is not a priori clear, which of these time factors should go into a time benchmarking metric: (i) only the time from start to end of the solver (ii) including the generation of the model for the specific solver, (iii) including pre-processing from the raw data. Arguably, a fair comparison considers the total time. Nevertheless, it can also be interesting to analyze the composition of the total time as this can help to identify the most costly steps and potential bottlenecks. Given the current status of software for quantum computing transpilation, embedding, and compilation times will in fact play a non-negligible factor. Hence, understanding the relation of time-to-solution and QPU time gives us insights into the (progress in) efficiency of these steps. However, these are also areas of active research and development and we expect them to be significantly reduced in the near future. Another factor to take into account when measuring the time-to-solution is the potential use of parallelization, which effectively corresponds to a trade-off between computational resources, i.e., cost, and run time [511].

It should also be noted that the nature of a run time benchmark strongly depends on the benchmarking goal, e.g., aiming to reach a feasible, proven optimal, or $\epsilon-$close to optimal solution. If the goal is finding a proven optimal solution, then it has to be taken into account that certifying optimality typically requires a significant amount of time compared to merely finding the optimal solution. In the case of feasibility problems, we must consider what to do if one method finds a solution and one does not. A good approach is to have feasibility problems as a separate class and count how many problems can be solved within a certain time limit per instance — even if sometimes none is found. Setting the right time limit is crucial. Short times < 1s are often difficult to measure together with the setup time. Common choices for classical heuristics are therefore < 1s, 10s, 60s, 600s, 3,600s, and 10,000s.

Different forms of optimization methods, for example deterministic vs. heuristic, also impact the specifics of a run time benchmark. Considering, e.g., non-deterministic methods that produce various solutions, one has to fix whether the run time is calculated as the total time, the average time to find a solution, the mini-

mum time to find a solution, etc. One proposal to address this, is to explore the whole boundary of time-to-target and optimality-gap-at-time values [457, 512]. Another metric that aims at giving a holistic point of view by considering a combined measure for solution quality and total heuristic run time is suggested by Berthold [513].

Quality metrics strongly rely on the nature of a problem. Firstly, one requires a solution to be compatible with the problem constraints, i.e., to be feasible, which can be easily checked. Ultimately, one wants to find the optimal solution or at least a solution that is $\epsilon-$close to the optimal solution. It should be noted that the evaluation of optimality may be resource-intensive or intractable. Given a feasible solution, the evaluation of the optimality gap may be identified via bounds given from continuous relaxations [514] or dual representations [515]. Another quality metric for optimization algorithms is solution diversity. Given multiple solutions with equal cost values, one would expect a fair sampling from these solutions. Interestingly, this property is not necessarily given in quantum annealing algorithms [516–518], and also not easy to realize on current gate-based devices [519, 520]. Similarly, one can measure the success rate of a heuristic algorithm to find solutions that are feasible or suffice a certain optimality distance in a fixed number of algorithm execution runs.

The practical difficulty of a problem is often related to the problem size, density or number of variable connections, and the nature of the constraints. The difficulty may be changed by modifying one or more parameters that characterize the system as a whole. These changes can even lead to computational phase transitions, which are marked by abrupt and significant shifts in the time to find a solution of certain quality, or even the feasibility of finding a solution. Understanding the range of problems and characteristics where phase transition behavior is observed presents a challenging task.

In order to provide a holistic quantum optimization benchmark, metrics that describe algorithm and solution properties according to these four classes should be measured. Depending on the problem, model, algorithm, and execution platform the respective metrics should be wisely chosen and faithfully reported on.

D. Problems

Next, we present a set of model-independent problems that show promise for establishing standardized quantum optimization benchmarks. The problems should be chosen sufficiently hard for state-of-the-art classical solvers to leave a margin for quantum methods but within an intermediate size range where current and/or near-term quantum technologies can be effectively deployed. Ideal problem instances are those for which existing methods are unable to provide provably optimal or feasible solutions. Furthermore, it is crucial that the problem can be formulated as a quantum-compatible model, i.e., with sufficiently few variables/qubits, limited connection density, relatively small coefficients, etc. In other words, we are looking for a sweet spot for current quantum systems that enables the investigation of quantum advantage compared to adequately chosen digital algorithms.

It can also be valuable for tracking the progress in algorithmic and hardware development to identify a family of problems that are similar in their nature but differ in their difficulty. Notably, complexity theory typically makes statements about the hardest instances in a problem class and, therefore, seldom provides descriptive arguments about the hardness of individual instances — which very much depends on the size, structure, and precise coefficients of the instances. Furthermore, the particularly chosen instances may correspond to crafted, random, or real-world problems:

Crafted instances are most likely the best way to find problem instances that are hard to solve for classical methods. However, there is a risk, that there is (or can be developed) also a classical method that takes advantage of the special problem structure, or that the instances are of no practical value.

Random instances are different since they lack structure. This can make them, depending on the problem, particularly easy or very difficult to solve. It certainly sets them apart, but how meaningful these instances are can be debated.

Real-world instances correspond in many practically relevant cases to MIPs that can be solved up to sufficient accuracy with existing solvers. Instances that are relevant and difficult to solve with existing methods are hard to find in a setting that fits the dimensionality and density limitations of current quantum hardware. In fact, these problems are generally hard to find because open problems of industrial relevance are rarely published or may be ill-defined. This highlights a common *selection bias*: We rarely model practically relevant problems in a way we know we cannot solve.

In the following, we list a set of binary problems that are promising candidates for quantum optimization benchmarking problems. These problems are random or crafted instances and relate only to few practical applications. Nevertheless, they offer a good test bed for algorithm development and progress tracking. Classical methods that can be used to tackle these problems include Branch-and-Cut based Integer (linear, non-linear, semi-definite) programming (ILP) [521], Pseudo-Boolean (PBO) optimization [522], SAT-Solving (SAT) [464, 523], Constraint Programming (CP) [524], and several other general techniques [525, 526] as well as a multitude of specific heuristic approaches.

Maximum Independent Set (MIS). Given a weighted graph $G = (V, E, c)$, we are looking to find

$$\begin{aligned} \min_{x \in \{0,1\}^{|V|}} \quad & \sum_{v \in V} c_v x_v \\ \text{s.t.} \quad & x_u + x_v \le 1, \, \forall (u, v) \in E. \end{aligned} \tag{23}$$

MIS can be formulated as a QUBO for a suitably large penalty factor P:

$$\min_{x \in \{0,1\}^{|V|}} \sum_{v \in V} c_v x_v^2 - P \sum_{(u,v) \in E} x_u x_v. \quad (24)$$

This is a classic **NPO**-hard problem even in case $c_v = 1$ and there are known instances that are hard to solve to proven optimality with existing methods starting at problem sizes of several hundred variables [527]. The allure of this problem class for quantum optimization benchmarking is that it is well suited for translation into relatively sparse QUBOs.

A subclass are MIS with unit disc graphs G (UD-MIS) which are defined by vertices V on a two-dimensional plane with edges E connecting all pairs of vertices within a unit distance of each other [528, 529]. These problems can be naturally mapped onto the structure of Rydberg quantum computers [530, 531] via the Rydberg blockade [387]. The optimization itself can then be carried out either in an analogue or digital fashion. It is also worth mentioning here that any QUBO can be mapped constructively to a UD-MIS with weighted vertices (UD-MWIS) [532].

Optimization problems that may be modeled as a MIS are, for example, *Sudoku* and the *multidimensional n-Queens problem*. Interestingly, 3D n-Queens problem instances on a 14^3 board with 2,744 binary variables, and most larger board sizes, have no known proven optimal solution. As there is by definition only one problem instance per board size, we can nicely track how far we get. Furthermore, it should be noted that the *maximum clique* problem can be solved as a MIS on the complementary graph.

(Multi-dimensional) Knapsack / Market Share. Given a set $I = \{1, \ldots, n\}$ of items, weights w_i, $i \in I$, and a capacity C, the basic *knapsack* constraint regarding a set of binary variables x_i, is $\sum_{i \in I} w_i x_i \leq C$. Knapsack problems typically have an objective function, where a profit p_i is assigned to each item, i.e.,

$$\begin{aligned} \max_{x \in \{0,1\}^n} \quad & \sum_{i \in I} p_i x_i \\ \text{s.t.} \quad & \sum_{i \in I} w_i x_i \leq C. \end{aligned} \quad (25)$$

A quadratic objective $\sum_{i,j \in I^2} p_{ij} x_i x_j$ corresponds to the *binary quadratic knapsack* problem [533]. In a *multidimensional knapsack* problem, we have J dimensions such that

$$\begin{aligned} \min_{x \in \{0,1\}^n, s \in \mathbb{R}^J} \quad & \sum_{j \in J} |s_j| \\ \text{s.t.} \quad & \sum_{i \in I} w_{ij} x_i + s_j = C_j, \ \forall j \in J. \end{aligned} \quad (26)$$

If designed with carefully selected weights, we get so-called *market share* problems [534] that are explic-

itly designed to be difficult to be tackled by classical branch-and-bound based methods. These problem instances are dense multi-dimensional knapsack problems [62, e.g. Chapter 31]. Still, there exist instances with less than $n = 100$ variables and $m = 8$ constraints which cannot be solved to proven optimality. However, for this problem class the difficulty depends strongly on the particular coefficients, the ratio c_i/w_i, and C. Problems with comparatively small C can be solved quickly by DPs and if the w_i are small compared to C, heuristics will give good results. For carefully chosen coefficients, however, these problems can be difficult to solve with classical methods to proven optimality. We argue that this problem class provides a good quantum optimization benchmarking candidate because there is clear evidence that it is challenging for classical digital systems to find (approximate) solutions already — even for small system sizes in the multidimensional case.

Low Autocorrelation Binary Sequences (LABS). This is a difficult binary non-linear optimization problem [12]. Given a sequence $S = (s_1, \ldots, s_k)$ of length k with binary $s_i \in \{-1, +1\}$, the autocorrelations of the sequence correspond to

$$\mathcal{A}_j(S) = \sum_{i=1}^{k-j} s_i s_{i+j} \quad (27)$$

for $j \in \{0, \ldots, k-1\}$. Furthermore, the sequence *energy* is defined as

$$E(S) = \sum_{j=1}^{k-1} \mathcal{A}_j^2(S). \quad (28)$$

The goal of the optimization problem is to find a sequence S that minimizes the energy function given above. There is only one particular instance per size N. Hence, the frontier of what is possible is always well defined. So far problems up to size $N = 66$ and $N = 127$ (skew-symmetric) can be solved to optimality. Given the spin-like binary variables, one might be able to find a model formulation that could provide a natural fit for a quantum platform. However, the quartic objective function also imposes some challenges to do so [104].

Quadratic Assignment Problem (QAP). The QAP [535] has remained one of the great challenges in combinatorial optimization. For $I = \{1, \ldots, n\}$

$$\begin{aligned} \min_{x \in \{0,1\}^{n \times n}} \quad & \sum_{i,j,k,l \in I} a_{ij} b_{kl} x_{ik} x_{jl} \\ \text{s.t.} \quad & \sum_{j \in I} x_{ij} = 1, \ \forall i \in I, \\ & \sum_{i \in I} x_{ij} = 1, \ \forall j \in I, \end{aligned} \quad (29)$$

where a_{ij} and b_{kl} describe the problem instance. It is still considered a computationally nontrivial task to solve to

optimality even for modestly sized problems, i.e., $n \approx 30$. If modeled as a binary problem, the number of variables is squared. Modeling this problem as a QUBO results in a dense formulation. Nevertheless, the long and unsuccessful search for improvements in classical methods indicates that there might be room for quantum advantage.

Sports Timetabling Problems. There are regular competitions held to find the best methodology for tackling timetable scheduling [536]. A mathematical formulation that describes problems of this form has been presented, for example by de Werra [537]. These problems describe scheduling problems of sports leagues or tournaments and are generated in a way that ensures at least one feasible solution exists per instance. These problems have a strong combinatorial structure meaning that in different feasible solutions many variables need to have different values.

Interestingly, for several medium sized instances no known (and investigated) method was able to find even a single feasible solution [536, 538]. There are different kinds of requirements, i.e., problems can be generated with increasing difficulty. From pure binary problems, to problems that rely on the counting of integer variables or binary representations thereof. What is particularly interesting about these problems is that one can generate instances at variable size and difficulty to track progress while it is known that sufficiently large problems are difficult for state-of-the-art methods.

Spin glasses. The study of optimization problems with respect to physics Hamiltonians, such as spin glass models [472, 539], is of great importance and has inspired the development of classical solvers, including simulated annealing [540] and simulated quantum annealing [541]. Spin glasses defined over different graphs have already been used to compare the performance of classical optimization with quantum annealing methods [472, 539]. This is enabled by their natural representation as quantum Hamiltonian. More specifically, the mapping of a spin glass problem onto a QUBO is trivial and leads to relatively sparse models. Although classical solvers perform well on spin glasses defined on regular lattices compared to quantum annealing, the possibility of digital quantum optimizers having better performance still exists — possibly on non-regular lattice structures [542]. Moreover, spin glass problems have properties that make them provably average-case hard for various families of classical algorithms [95]. It remains open as to whether quantum algorithms can do better in these average-case settings, and thus, superpolynomial speedups for spin glass problems are still possible. Considering all of the above, we argue that they offer an interesting candidate for quantum benchmarks, especially in a cross-platform setting.

E.   Demonstrations

Experimental realizations of large scale quantum optimization algorithms have to address several challenges. Not only is it difficult to cope with hardware induced limitations – see Sec. V for further details – but it is already a non-trivial task to choose an appropriate benchmarking problem – see Sec. VI D. In fact, most existing hardware demonstrations of quantum algorithms for optimization are considering QUBO-type problems with a grid structure. Several papers argue in the following way: First, it is stated that a problem is difficult according to complexity theory, e.g., it is **NPO**-hard. Then, a feasible – but not provably optimal – solution is found by applying a quantum algorithm. However, finding a feasible but not necessarily optimal solution is not necessarily difficult for state-of-the-art classical methods [473, 488, 543]. Thus, there is an inconsistency in the argument, since classical algorithms are ruled out with a reference to complexity theory while allowing to use quantum heuristics. These results are valuable capability demonstrations, i.e., they can be seen as hardware benchmarks rather than performance benchmarks. Nevertheless, there is a potential for better performing quantum heuristics, and to show this, we require careful and fair systematic benchmarking. In this section, we present an overview of selected state-of-the-art experimental quantum optimization results run on gate-based quantum computers with more than 16 qubits that represent excellent starting points for comprehensive quantum optimization benchmarks.

Tab. IV summarizes existing quantum optimization hardware implementations run with (variants of) QAOA or VQE and problems corresponding either to (weighted) MAXCUT, Sherrington-Kirkpatrick, or other instances of QUBO/PUBO that use more than 15 qubits. The most commonly employed graph structures are random 3-regular (R3R) graphs or graphs that are particularly fitted to hardware layouts such as heavy-hexagon or nearest neighbor grid layout [386, 396]. Besides the target problem and employed algorithm, the table also lists the problem size in terms of number variables (which equals the number of qubits in the considered cases), the density (which quantifies the number connections between variables), the best and mean approximation ratio, the depth of the ansatz in terms of the number QAOA operator repetitions (except for results from Amaro *et al.* [545], which corresponds to the repetitions of a "RealAmplitudes" ansatz with pairwise entanglement [548]), and the year of publication. All presented instances with fewer than 50 qubits employ a standard QAOA or VQE schemes, i.e., they are optimizing the variational parameters using the quantum computer. Since the references report on shot numbers respectively number of parameter update iterations in various ways (or not at all), the respective numbers are difficult to compare and have, therefore, not been added to the table. However, for future benchmarks, providing such numbers in a comparable format would be important. The results of Ma-

40

Table IV. An overview of state-of-the-art experimental realizations of optimization algorithms on gate-based quantum computers with more than 15 variables. In cases where data was not made available in the corresponding publication or the accompanying data repository, we denote this in the respective field with *N/A*. AR denotes the approximation ratio, given based on the *mean* and the *best* sample value of the experiment, *n.n. grid* stands for nearest neighbor grid. Furthermore, JSP, FVQE, QAMPA, GQAOA, and NDAR abbreviate job shop scheduling problem, filtering variational quantum eigensolver, quantum alternate mixer-phaser ansatz, greedy QAOA, and noise-directed adaptive remapping, respectively.

Problem	Algorithm	Qubits	Density	AR		Depth	Year	Ref.
				mean	best			
Sherrington–Kirkpatrick	QAOA	17	100%	0.61	N/A	$1 \le p \le 3$	2021	[386]
MAXCUT (R3R)	QAOA	20	16%	0.64	1	$p = 2$	2023	[403]
MAXCUT (R3R)	QAOA	20	16%	0.94	1	$p \le 10$	2023	[544]
MAXCUT (R3R)	QAOA	22	14%	0.67	N/A	$1 \le p \le 3$	2021	[386]
MAXCUT (n.n. grid)	QAOA	23	13%	0.72	N/A	$1 \le p \le 5$	2021	[386]
QUBO (JSP)	FVQE	23	N/A	0.88	N/A	$1 \le p \le 2$	2022	[545]
MAXCUT (heavy-hex.)	QAOA	27	8%	N/A	1	$p = 2$	2022	[402]
MAXCUT (R3R)	QAOA	30	10%	0.59	0.83	$p = 2$	2023	[403]
MAXCUT (R3R)	QAOA	32	10%	0.88	1	$p \le 10$	2023	[544]
MAXCUT (R3R)	QAOA	32	10%	N/A	1	$p = 2$	2023	[226]
MAXCUT (R3R)	QAOA	40	8%	0.58	0.78	$p = 2$	2023	[403]
Sherrington-Kirkpatrick	QAMPA	50	100%	0.55	0.83	$p = 2$	2023	[264]
Sherrington-Kirkpatrick	QAOA	50	100%	0.54	0.84	$p = 2$	2023	[264]
Sherrington-Kirkpatrick	GQAOA	72	100%	N/A	0.92	$p = 1$	2023	[229]
Sherrington-Kirkpatrick	NDAR-QAOA	82	100%	0.87	0.97	$p = 1$	2024	[546]
QUBO (heavy-hex.)	QAOA	127	2%	0.67	0.85	$1 \le p \le 2$	2023	[438]
PUBO (heavy-hex.)	QAOA	127	2%	0.65	0.84	$1 \le p \le 2$	2023	[438]
PUBO (heavy-hex.)	QAOA	127	2%	0.73	0.89	$1 \le p \le 5$	2023	[547]
QUBO (heavy-hex.)	QAOA	414	0.6%	0.57	0.69	$p = 1$	2023	[547]
PUBO (heavy-hex.)	QAOA	414	0.6%	0.56	0.68	$p = 1$	2023	[547]

ciejewski *et al.* [264] with 50 qubits consider both QAOA, and QAMPA [263], a hardware-efficient ansatz derived from QAOA and using the same number of entangling gates. The 72 qubit experiment by Dupont *et al.* [229] on the other hand, employs a greedy procedure. In each iteration cycle, several variables are set to values that are evaluated with a classical procedure, hence, producing a smaller optimization problem to be solved on the quantum computer. Further, the demonstrations with more than 100 qubits/variables [438, 547] employ either a gridsearch over ~7000 parameter configurations or *concentration of parameters* [222–224], also cf. Sec. IV A 1, where QAOA on large instances can be run on quantum hardware using parameters found for classically trained smaller (e.g., 16-qubit) instances [547]. Finally, the recently proposed Noise-Directed Adaptive Remapping meta-heuristic improves algorithm performance on noisy hardware by adaptively gauge transforming the cost Hamiltonian, with its effectiveness demonstrated for QAOA with 82 qubits [546].

The table illustrates that most experiments with higher qubit numbers have lower problem densities. Increasing the density for a larger number of qubits would require additional SWAP gates, cf. Sec. V, hence, resulting in increased hardware noise. This phenomenon manifests itself also in the resulting approximation ratios. Considering the approximation ratios, it should be noted that for common graph problems such as MAXCUT the objective values usually lie between 0 and a maximum value $C_{\max}$, while for general problems such as Ising models usually the objective values are inside an arbitrary interval $[C_{\min}, C_{\max}]$. Therefore, a fair comparison of approximation ratios necessitates that the data of each reference is expressed consistently. Thus, we first convert a problem into a minimization problem, and then, we normalize by the range $|C_{\max} - C_{\min}|$, i.e., we define the approximation ratio as $(C_{\max} - \langle C \rangle)/(C_{\max} - C_{\min})$, where $\langle C \rangle$ denotes the achieved objective value. In fact, some published values do not correspond to range-normalized data, such as the ones from Harrigan *et al.* [386]. Therefore, the values listed in the table are computed from the corresponding data repositories. Moreover, several references give the approximation ratio based on the best sample (which may improve with a higher number of shots) and others on the mean sample value $\langle C \rangle$. Hence, these values are listed separately in Tab. IV.

In addition, we would like to highlight a work by Fuller *et al.* [237] and Moses *et al.* [226], where strategies have been investigated to address problems with more variables than qubits. In the former, MAXCUT problems on a planar graph with up to 40 variables are solved with QRAO using up to 15 qubits, cf. also Sec. IV A 1. In fact, the approach achieves an approximation ratio of 0.905 for the 40 variable problem executed on quantum hardware. In the latter, a variant of QAOA with mid-circuit measurements and qubit re-use has been tested which could

be scaled to 130 variables on 32 qubits, achieving an approximation ratio around 0.8. In general, the intrinsic differences between the method employed in this work and standard QAOA / VQE approaches make it difficult to compare these results with the ones presented in the rest of this section. Nevertheless, we believe that increasing the variable-to-qubit-ratio beyond one is a very interesting direction to scale towards relevant problems.

To conclude, these demonstrations build a foundation for future quantum optimization benchmarks. They also highlight the difficulty as well as importance to agree on common metrics to achieve comparable and fair benchmarks. This section is a first attempt for a coordinated effort to achieve this goal.

VII. ILLUSTRATIVE APPLICATIONS

Identifying optimization problems that are promising candidates for a practically relevant quantum advantage is challenging. This is due to the many open questions about future performance of quantum optimization algorithms, as well as the heuristic nature of many of them. Further, benchmarking problem instances known to be hard for today's solvers are often (hand-)crafted. Thus, even if a quantum advantage could be shown for one, it would not necessarily generalize to other problem instances. While these benchmarks are still crucial to identify promising problem structures and track algorithmic progress, we also need to look into real-world problem instances. Further, there might be a *selection bias*, as already discussed in Sec. VI D. We are often focusing on models where we have an idea about how to solve them. New formulations of a problem, for which no solver may exist yet, are often unconsciously discarded.

Thus, within this section, we discuss two exemplary industries, i.e., finance and sustainable energy — two fields that have many outstanding (optimization) challenges. For general discussions on the potential applicability of quantum algorithms for finance and sustainability we refer the interested reader to the following resources [549–554]. Here, we discuss families of optimization problems arising in the two industries with increasing model complexity. Importantly, the goal of the following discussion is not to claim that the selected use cases are the most likely candidates for near-term quantum advantage. Instead, they should serve as a playground of (possibly simplified) real-world problems that can help to set the course for research and link to problem classes discussed in Sec. IV. Further, the discussion highlights that being practically relevant requires taking into account many details and complications that most published results in the related quantum optimization literature are lacking. Thus, defining a family of problems with increasing complexity helps to determine progress on the quest towards quantum advantage in optimization.

A. Financial Asset Allocation

Optimization problems in finance span a wide range of applications, from the quantification and management of risks, to asset allocation, option pricing, macroeconomic modeling, algorithmic trading, lending and more [555]. The goal of this section is to present a variety of challenging optimization problems from the financial realm that may provide interesting use cases for quantum algorithmic development, as the technology advances.

Firstly, we give a short overview of the field and a discussion of advancements in quantitative financial theory pertaining to optimization. We provide a few example problems and a discussion of their practical limitations in Sec. VII A 1. Then, we present the current state of quantum optimization and early investigations of use cases in finance in Sec. VII A 2. Next, we provide a deep-dive into a particularly hard problem with many unresolved questions and evident practical relevance: the *asset allocation problem* and different formulations thereof, with increasing levels of complexity in Sec. VII A 3. Finally, in Sec. VII A 4, we conclude and outline further research directions that could act as an exploration ground for quantum optimization applications in finance.

1. Overview

The global financial system is the backbone of economics, embedding into every part of society through either direct digital channels, such as monetary transactions and contracts, or in the form of indirect physical carriers, such as transferable goods and services. It represents the world's largest human-made regulated stochastic network, hosting trillions of interactions between market makers, traders, consumers, regulatory control and monitoring entities. The system is powered by a variety of distributed computing technologies and operates on the shoulders of over 120 years of financial theory, heuristic analysis, and a broad range of mathematical tools and probabilistic models informing decision making — within regulatory and business constraints. This enables one to manage risk, counter financial crime, derive investment and business decisions, and service consumer needs with personalized experiences [556].

Despite a century of progress, many models in regulated financial institutions suffer from different degrees of approximations and model trade-offs due to irregular and/or scarce information and the need to balance model complexity, computational feasibility, and regulatory, as well as business constraints. This is generally driven by challenges originating from fundamental unknowns underlying the interaction dynamics between market participants [557, 558], data quality anomalies in observables and the overall fast-paced change of the world. The latter may gradually become a notable challenge for many modeling trade-offs relied on today, and thus, could influence the financial system's systemic resilience [559–561]. For

example, more complex probabilistic drivers are emerging from new consumer-banking interactions [562], climate change and energy transmission risk guidelines [563], to the economic impact of changing geopolitical dynamics [564]. This demands the development and analysis of new modeling techniques and optimization algorithms to better inform decision making under uncertainty. In many finance problems, even *small improvements* can have a *significant impact*. Although, it is still unclear if and how a potential quantum advantage could be realized for financial optimization problems – even as the underlying technology advances – a thorough understanding of the broad spectrum of classical models and their limitations in the financial industry could help guide the development of quantum optimization approaches for use cases that deal with uncertainty in objective functions, decision variables or constraints.

While not aiming for rigor or completeness, the following provides a bigger picture and short recap of advances and challenges in financial optimization problem domains for their respective input dependencies. This links to the roots of quantitative financial theory and the understanding that we cannot explain with confidence *why* the prices of financial assets move, but only attempt to model *how* they move. As a result, the modeling of price dynamics and volatility has become a core focal point in finance with many models aiming to capture its essence: from Brownian motion [557] to perfect delta-hedging — assuming no price jumps and financial crashes [565]; capturing volatility clustering phenomena [566] with GARCH [567] and Heston [568] models — assuming volatility fluctuations decay over one timescale; Multifractal Random Walks [569] and Rough (Heston) Volatility models [570–572] capturing most *stylized facts* [573] from empirical observations of financial time series, such as heavy-tailed probability distributions [574]. This list summarizes just a few milestones over a century of financial modeling, effectively demonstrating that capturing market dynamics with such complex input patterns is difficult.

There are also many optimization problems in finance that do not need to account for input uncertainties to prove valuable for business, such as linear programming problems (see Sec. IV B), quadratic programming problems (see Sec. IV A), and (mixed) integer programming problems (see Sec. IV C). However, for most instances corresponding to these classes, there are established, well-performing classical solvers, such as interior-point and simplex methods for linear and quadratic programs, or branch-and-bound and combinations with cutting-plane methods for integer programs. Nevertheless, these problems form an excellent ground to benchmark and develop quantum optimization algorithms and potentially detect value in specific financial applications.

Financial optimization problems of substantial complexity that face limitations when being approached with state-of-the-art solvers may, hence, be interesting subjects for the exploration of quantum algorithms and are

summarized here with a few examples:

- Dynamic programming problems (see Sec. IV D) for pricing and hedging of derivatives on binomial lattices, or structuring of collateralized mortgage obligations with maximized profits from issuance;

- Stochastic programming problems (see Sec. IV E) for minimizing bond portfolio credit risk using a Conditional Value-at-Risk measure, or asset-liability management maximizing wealth, or retiring outstanding debt at minimal cost, or creating a synthetic option strategy to reach desired payoff at the end of a planning horizon;

- Robust programming problems (see Sec. IV F) for optimizing portfolios while taking estimation risk of input parameters into account, or determining lower and upper bounds on the price of a security;

- Multi-objective programming problems (see Sec. IV G) for dynamic margin-volume balancing to price mortgages combining optimal revenue, balance sheet and business objectives, or optimizing an investment portfolio of stocks combining return, risk diversification, incentivizing decarbonization and Environmental, Social, and Corporate Governance (ESG) [575] objectives.

2.   Related Work

The majority of existing quantum optimization research in finance focuses on investigating quantum variants of linear and quadratic optimization problems and simplified (mixed) integer programs. Most publications employ a QUBO formulation, combined with a ground state solver such as QAOA or VQE (see Sec. IV A) that is executed on numerical simulators, gate-based hardware, or annealers, typically for systems with less than 50 variables. In the following, we present a selection of the corresponding papers.

Brandhofer *et al.* [576] study portfolio optimization using QAOA for quadratic binary optimization constrained by the number of assets. That is, the presence or absence of a particular stock in a portfolio has a value of 1 or 0 respectively, and the sum of the assets is the value of the investment, which is a (simplified) budget constraint. Risk is minimized by having the smallest combined covariance between all pairs of stock values as represented by a covariance matrix; this minimum ensures portfolio diversification.

In a numerical study, Hodson *et al.* [577] compare a Quantum Alternating Operator Ansatz to QAOA for models that incorporate trading and investment constraints. Baker and Radha [400] evaluated the solution quality as functions of qubit number and circuit depth. Mugel *et al.* [578] compare methods for dynamic portfolio optimization on quantum processors using 8 years

of data for 52 assets with quantum annealing, VQE, and a quantum-inspired optimizer based on Tensor Networks using pre-processing to reduce the problem dimension. Venturelli et al. [579] compare the performance of reverse quantum annealing for portfolio optimization problems formulated as QUBOs with classically executed genetic algorithms. Constrained QAOA solutions to portfolio optimization are also presented in Herman *et al.* [265], who use Zeno dynamics with rapid sampling to restrict the solutions to a constrained subspace. Giron *et al.* [580] use QAOA to study the problem of collateral optimization — the optimal allocation of financial assets to satisfy obligations at minimum cost. In a broader context, a quantum optimized portfolio allocation could feed into a risk model evaluated on a quantum computer [581–583].

Other finance optimization applications that have been proposed for quantum algorithms include transaction settlements, where the goal is to find the maximum number of cash transactions between multiple parties given their liquidity [320], and Anti Money Laundering, which matches network motifs on snapshots of transaction graphs [584].

3. Deep Dive: Optimal Asset Allocation

Among the hardest problems in the financial industry is the market adaptive search for an optimal portfolio composition with an investment allocation in financial assets that maximizes the return of investment within a given risk appetite and time frame, while complying to possibly other objectives and constraints. Reasons for its complexity relate to the stochastic nature of the financial system and various statistical and computational challenges as discussed in Sec. VII A 1. It is also a problem where marginal improvements in practical solutions can have substantial impact. In this context, the term *solution* should be discussed. It is difficult to find a provable or universal solution, instead one accepts an outcome with different forms of trade-offs to encompass particular use case requirements, market environments, asset class restrictions or investment scenarios. While over the past 70 years, both academic researchers and industry practitioners have created a large repertoire of targeted heuristic approaches and rigorous mathematical tools, it remains one of the most hunted challenges in finance [585]. This partially explains why the quantum computing community is exploring this use case in various quantum optimization for finance publications [576, 577, 586, 587]. However, we stress that it remains unclear how to systematically advance towards the complexity and value frontier at which a real-world practical impact can be realized.

In this section, we introduce the concept of value-guided *levels* with examples representing incremental increase in use case complexity by respective problem formulations — for reasons of simplification in the mean-variance framework, as illustrated in Fig. 8. The aim is
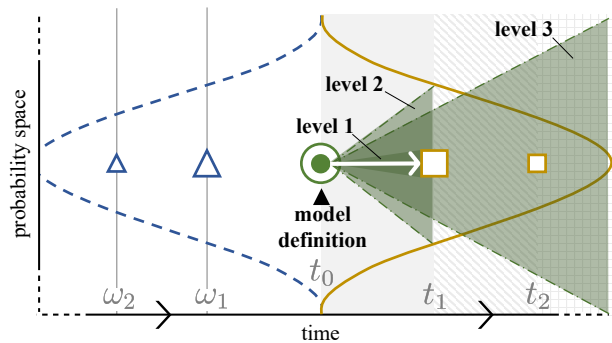
Figure 8. An illustration of the optimal asset allocation at time t_0 (model definition), considering perfect predictors of asset valuation at the end of investment horizon t_1 given prior statistics $\omega_1, \omega_2, \ldots$ (level 1), then taking uncertainties of these predictors into account but still with a limited view until t_1 (level 2), before anticipating a multi-period allocation strategy and respective uncertainties beyond t_1 (level 3).

to systematically explore different challenges of this use case and possibly inspire the exploration and research of quantum optimization algorithms in this domain.

Model Definition. From the ground up, it needs to be well understood within which environment we are attempting to construct an asset allocation problem. For instance, we obviously cannot comprehend all the markets with interactions between millions of traders with each a different investment style, behavior, risk limits or trading frequency. But rather than accepting the entire universe of possibly feasible portfolios $\mathcal{S}$, a preliminary *steering of the downstream complexity* for all other levels can help build up and realistically frame the use case. The field of investing is found with very different business domains, expectations and requirements that yield various degrees of complexity to start with. For example, a wealth manager for retail, affluent or high-net-worth individuals may have different requirements for composing or rebalancing investment portfolios than say, a large asset manager. While this seems rather obvious, it is key to formulate the model purpose, and thus, its expected performance. In particular, an informed pre-selection of diversified asset classes, industry exposures, risk profiles or expected dynamics for a given investment time horizon helps build a more robust starting point, t_0, for the optimization problem.

Level 1. At this stage we begin with the simplest problem formulation based on Harry Markowitz's introduction of modern portfolio theory in 1952 [588], by formalizing portfolio diversification and optimal return-risk trade-offs with a mean-variance framework. It assumes a risk-averse myopic investor in a frictionless, one-period financial market with exact knowledge of parameters capturing asset price dynamics. The aim is to find an optimal portfolio $x = (x_1, ..., x_N) \in \mathbb{R}^N$ from permissible portfolios $\mathcal{S}$ with fractions x_i of a fixed budget to be fully invested into N pre-selected, non-redundant as-

sets at time t_0, based on estimates of expected returns $\mu \in \mathbb{R}^N$ and their positive semi-definite covariance matrix $\Sigma \in \mathbb{R}^{N \times N}$ at the end of one investment period t_1. This can be formulated in three equivalent variants as *quadratic programming problems* with linear constraints and without uncertainty in the parameters:

$$\begin{aligned}
\min_{x \in \mathbb{R}^N} \quad & \frac{1}{2} x^\top \Sigma x \\
\text{s.t.} \quad & \mu^\top x \geq r_{\min}, \\
& \mathbb{1}^\top x = 1, \\
& x \in \mathcal{S},
\end{aligned} \tag{30}$$

where the convex quadratic objective minimizes the portfolio risk $x^T \Sigma x$ with an expected minimal portfolio return $r_{\min}$. Instead, one can also maximize the expected return $\mu^T x$ while bounding the risk $x^T \Sigma x \leq \gamma^2$ as a quadratic constraint, which can be transformed into conic form [589]. Another alternative is to maximize a concave quadratic utility function $\mu^T x - \lambda/2 \; x^T \Sigma x$ with the risk-aversion factor λ for steering the return-risk trade-off and variable focus on tail risk. However, all of these formulations are relatively trivial to solve, especially in the continuous setting, where a closed-form solution may be derived using Lagrangian multipliers [590] considering equality constraints [591] and with Karush-Kuhn-Tucker conditions considering also inequality constraints [592]. Notably, realistic market conditions are not captured by this model formulation. Nevertheless, this simple model offers excellent benchmarking opportunities to evaluate advances in quantum technology. In fact, the model complexity can also be increased in a controlled manner by adding additional constraints that might match a pre-defined investment strategy and, as the term $x \in \mathcal{S}$ is meant to reflect, to further constrain the universe of portfolio configurations and test with an increasing number of additional linear constraints that match a pre-defined investment strategy. In practice, for a sufficiently large number of constraints one often has to rely on numerical approaches. As discussed in Sec. VII A 2, so far, most quantum approaches to this use case employ QUBO formulations with binary or integer encodings, which result in an **NPO**-hard combinatorial problem and constraints that are converted into penalty terms [593] of the objective function. Since these penalty terms only facilitate approximate encoding of the constraints, additional substantial post-processing would be required to achieve reliable results.

Level 2. A severe limitation of Level 1 is the ignorance of uncertainty in the inputs and instability of the optimization [594]. Thus, it is of limited practical value to derive actual investment decisions. In fact, the optimization is highly sensitive to *small changes* in the estimated expected returns μ and covariance matrix Σ, leading to *large differences* in the resulting optimal portfolio x. At this stage, the problem formulation is extended to take the stochastic nature of underlying ob-

servables into account. The root challenge comes from the portfolio-specific operating regime of the observation ratio $q = N/T$ given the prior number of empirical (sample) realizations T for the N considered assets at t_0. In the large limit with $q \to 0$, the sample estimators μ and Σ would converge to the *true* realized (population) μ_{true} and Σ_{true}, respectively. But in reality most asset allocation problems suffer from a lack of data (except maybe in high-frequency trading) with $q > 0$. For example, a realistic scenario for stocks could be $N = 600$ with $T = 2500$ of about 10 years of daily returns and, thus, $q = 0.24$. In case of a few thousand stocks, the issue of stock lifetime and structural evolution of markets over time becomes relevant, shifting the problem into the large dimensional limit regime with $q \sim 1$ and fast diverging estimators with $q > 1$. If one would aim to increase the T statistics with intra-day observations, one may not capture the dynamics for lower frequency trading strategies anymore and possibly introduce biases in the resulting optimal x. This makes the problem very complex, even for a small number of assets, and usually forces the industry to simplify $\mathcal{S}$ at the level of model definition. In the following, we are going to present several approaches to address this challenge, ordered by increasing levels of complexity: first we de-bias and reduce the estimation error of μ and Σ prior to the optimization, then adding regularizing techniques to the optimization itself, before combing both. Among various methods, the Black-Litterman model [595] could be leveraged to create an expected return prediction by combining actual market returns with independent expert beliefs about their future, taking into account, for instance, recent news, economic forecasts, analyst ratings, or financial statements [596, 597]. Alternatively, a robust optimization problem can be formulated to achieve a stabilizing effect on μ during the optimization itself. For this purpose, we define the ellipsoidal uncertainty set $\mathcal{U}_\mu = \{\mu \mid (\mu - \mu_0)^\top Q_\mu^{-1} (\mu - \mu_0) \leq \kappa^2\}$, where Q_μ is the covariance matrix of estimation errors in μ, and κ is the uncertainty aversion defining the width of the uncertainty. This induces a new optimization problem by replacing μ with:

$$\tilde{\mu}(x) := \operatorname*{argmin}_{\mu \in \mathcal{U}_\mu} \mu_0^\top x - \kappa \sqrt{x^\top Q x}. \tag{31}$$

Even if the resulting improved estimate of μ, denoted by $\tilde{\mu}$, still substantially deviates from μ_{true}, finding the minimal out-of-sample risk portfolio for it may still be valuable in practice. The sample covariance matrix Σ for $q \in (0, 1]$ can suffer from small eigenvalues near zero, leading to a potential underestimation of the realized (out-of-sample) risk exposure at t_1. To address this issue, Random Matrix Theory [598–600] provides efficient tools to de-bias the eigenvalues of Σ such that the resulting $\widetilde{\Sigma}$ is closer to Σ_{true} [601–605]. A factor modeling method to represent the portfolio returns with a small number M of common factors [606] can additionally help to circumvent the curse of dimensionality if $M \ll N, T$. Furthermore,

one may want to further regularize Σ by adding L_1- or L_2-norm constraints to the optimization. Arguably one of the most important ones is the consideration of a transaction cost function $\mathcal{C}(x \mid x_0)$ that estimates the cost for rebalancing the portfolio x_0 at t_0 towards the optimal portfolio x at t_1, taking for instance broker fees and implementation shortfall into account. This results in the following problem formulation:

$$
\begin{aligned}
\max_{x \in \mathbb{R}^N} \quad & \tilde{\mu}^\top x - \mathcal{C}(x \mid x_0) - \frac{\lambda}{2} x^\top \widetilde{\Sigma} x \\
\text{s.t.} \quad & \mathbb{1}^\top x + \mathcal{C}(x \mid x_0) = 1, \\
& x \in \mathcal{S},
\end{aligned}
\tag{32}
$$

where $\mathcal{C}$ determines the complexity of the problem and depends on how realistic transaction costs should be represented. For example, a piecewise linear non-convex function capturing transaction volume and fixed costs for buying and selling an asset can be formulated, or a more realistic variant with stronger penalization on turnover considers costs proportional to the L_1-norm of rebalancing with $\mathcal{C}(x \mid x_0) = \beta \|x - x_0\|_1$, where $\beta > 0$ denotes a cost parameter. While formulations of this problem that are based on $\mathcal{C}$ employ strong regularization on Σ, depending on the respective q regime set out in the model definition, the resulting x may still be biased and sensitive to estimation errors. This can be further addressed with two methods. The first one uses a non-parametric bootstrap resampling approach [607] and the second one uses a nested clustering approach [608] that partitions $\widetilde{\Sigma}$ by forming clusters of highly-correlated assets. The latter approach can help to prevent error propagation of intra-cluster noise across clusters. Due to a reduction of dimensionality and better control of noise isolation with smaller sample sizes, the exploration of near-term variational quantum algorithms with realistic stock portfolio sizes may become possible in this setting.

Level 3. Until this point we have restricted ourselves to a static one-period market environment. At this level, we introduce the next complexity layer by considering market dynamics and temporal dependency. For example, similar to volatility, correlations fluctuate, evolve with time and jump or even flip signs [609, 610]. However, our understanding of correlation dynamics is nowhere near our understanding of volatility, which creates an interesting domain to explore with quantum computing. Similarly, this is the case for multi-period optimization, which could help capture inter-temporal effects, such as accounting for time-varying constraints, return forecasts, cost-favorable positions for trading in following periods, or foreseeable events impacting risk, trading volume or liquidity. Although there is a lot of research interest, given the computational hardness of the multi-period domain, most practitioners, as well as classical enterprise solver suites, cannot effectively tackle these use cases [585]. Essentially, the h-period optimiza-

tion problem can be *generally* formulated as follows:

$$
\begin{aligned}
\max_{x_{t+1}, \ldots, x_{t+h}} \quad & \mathbb{E}\left[\mathcal{X}(x_{t+1}, \ldots, x_{t+h}) \mid \mathcal{F}_t\right] \\
\text{s.t.} \quad & x \in \Theta,
\end{aligned}
\tag{33}
$$

where $\mathcal{X}$ is the inter-temporal utility function given the filtration $\mathcal{F}$ associated to the probability space at the tth period with a set Θ of constraints. If we can assume that $\mathcal{X}$ is separable in time, then the objective function can be written as $\min_x \{a(x) + b(x)\}$ with a static (forward-looking) part $a(x) = \sum_\tau a_\tau(x_\tau)$ and a dynamic (coupling) part $b(x) = \sum_\tau b_\tau(x_{\tau-1}, x_\tau)$ with $\tau = t + 1, \ldots, t + h$, and their respective separation of constraints becomes $x \in \Theta^{(a)} \cap \Theta^{(b)}$. Here, a_τ represents a one-period optimization program such as the mean-variance terms discussed in Levels 1 and 2, while b_τ serves as an intra-period regularization penalty such as the $\mathcal{C}(x_\tau \mid x_{\tau-1})$ transaction cost functions discussed in Level 2. Problem formulations of this form and other multi-objective settings are gaining substantial interest in the financial industry, and will play a key role in the coming years. Thus, it is interesting to investigate new approaches for this use case with quantum technology and inspire the development of new quantum algorithms.

This concludes our short deep-dive illustration with increasing levels of complexity and selected examples within the mean-variance framework. But notably, this can be extended as our exploration of quantum solutions evolve. In particular, the development of new quantum optimization solutions should transition from current Level 1 investigations to Level 2 and 3 challenges, and with that drive research towards handling optimization problems under uncertainty in order to aim for valuable contributions to the financial industry.

4.   Outlook

The financial industry offers a broad spectrum of hard optimization problems and demands the exploration of new solution approaches to tackle many of today's trade-offs. This short condensed review provides a satellite view of open optimization challenges relevant to the financial industry. A selection of use cases are discussed along with their classical challenges and how optimization problems can be constructed with increasing complexity levels. The main challenge remains to effectively combine both the modeling and optimization of problems to achieve practically relevant results for industry use cases. However, there are many more directions to investigate, such as handling skewed, non-Gaussian heavy tailed probability distributions, Bayesian approaches and industry sector or subgroup specific priors, or non-Markovian decision processes. All of these aim to inspire and guide the exploration of research, development and benchmarking of potential industry-specific quantum optimization applications.

B.   Sustainable Energy Transition

Sustainability describes the goal of a responsible use of resources to guarantee a long-term existence of humanity on earth in a healthy environment. While sustainability has multiple dimensions, a common focus is put on environmental problems, such as countering climate change. For this paper, we focus on a subsection of sustainability, that of sustainable transition of the energy sector. It is a particularly relevant area as it is rich with optimization problems with societal benefit. We discuss multiple problems in this domain that can serve as illustrative test cases with different challenges to benchmark optimization algorithms. The use cases described here serve as an interesting playground to generate examples derived from real-world data to study corresponding quantum optimization algorithms. While they might not necessarily be candidates for a practical quantum advantage in near future, they can help to improve our understanding about problem characteristics where quantum optimization may have some potential over classical algorithms and where not.

1.   Power Grid Overview

For decades, society has relied on carbon-intense and centralized energy sources [611]. Few powerful generation sites were responsible for meeting energy demands, and adjusting energy generation was relatively simple with fossil fuels. However, the energy sector is changing rapidly. Technological advancements and climate-change awareness have led to an increase in renewable energy sources. Power grid digitization [612, 613] has resulted in greater connectivity and interactions between customers, producers, utilities, and grid operators. This has revolutionized how we manage and optimize energy resources by enabling real-time monitoring, data-driven decision-making, and enhanced control over energy consumption and distribution.

However, the transition to renewable energy sources and the digitization of the power grid are not without challenges. Renewable energy sources such as solar and wind are intermittent and probabilistic in nature, and, they require the coordination of a large number of assets; for example, to match energy supply with demand. Energy storage solutions are essential for managing the variability associated with renewable energy production, but they present their own set of challenges, including cost and scalability. Additionally, power grid digitization introduces multiple challenges including cybersecurity and data privacy. Adapting grid systems to this new paradigm will necessitate substantial investment and innovative solutions, potentially surpassing the capabilities of classical computer systems.

In this section, we discuss three types of optimization problems in power grids. We start with a brief overview of the area in Sec. VII B 1 and related quantum optimization work, in Sec. VII B 2. In Sec. VII B 3, we introduce *electric mobility* (e-Mobility) and present three families of e-Mobility use cases and how they connect to the problem classes and algorithms in Sec. IV. Finally, Sec. VII B 4 concludes and outlines potential research directions.

2.   Related Work

Research in quantum optimization for power grids is currently in its early stages, with a predominant focus on the unit commitment problem (UCP) [312, 614]. Koretsky *et al.* [615] consider the UCP for power networks by combining quantum and classical methods: QAOA handles the binary on-off variables for each unit and a classical optimizer handles the continuous variables for how much power each unit should provide. Takahashi *et al.* [616] writes the UCP as a QUBO problem for network switches equal to 0 or 1 if open or closed, subject to connection, voltage, and maximum current constraints. They model the constraints as penalties in the objective function and solve the problem with annealing. Similarly, Halffmann *et al.* [617] and Braun *et al.* [618] formulate the UCP as a QUBO with penalty terms. Mahroo and Kargarian [619] decompose the UCP into a sequence of quadratic continuous and binary (unconstrained) subproblems, similarly to ADMM [317, 318], cf. Sec. IV C. Other non-UCP quantum optimization works in power grid include [620, 621], which formulate a power flow problem as a QUBO with constraints encoded as penalties in the objective, and [622], which addresses an energy supply scheduling problem with storage, where the optimization is formulated as a constrained quadratic program and solved with annealing.

All of the works mentioned above have in common that they focus on small, illustrative problems and that they explore simple mathematical models (mostly QUBOs) that do not capture the complexity of real-world use cases. Although these works may not have a direct practical impact yet, they could help to pave the way for advancing the corresponding theory, problem formulations, algorithms, and to understand practical requirements on optimization solutions to have practical impact.

3.   Leveraging the Flexibility of e-Mobility

As outlined in Sec. VII B 1, the world is progressively adopting carbon-neutral energy generation methods [623, 624]. While renewable sources like wind and solar power play a vital role, they often produce energy when demand is low. To address this disparity, it is crucial to invest in energy storage solutions, such as batteries, and the optimized coordination thereof.

Electric vehicles (EVs) are an energy storage solution of increasing importance. In 2022, the number of EVs on the road exceeded 26 million [625, pp. 14], and the

global fleet of EVs is expected to grow to about 240 million in 2030 — 10% of the road vehicle fleet [625, pp. 109]. To put this opportunity in perspective, EV batteries can typically store 20 to 100 kWh and can have an output power of 10 kWh when connected to a Type 2 charger. Thus, 400 EVs, in principle, can deliver 4 MWh of energy to the grid, which is approximately the same output power of an energy storage substation [626].

To leverage e-Mobility's potential to stabilize the energy network and balance supply and demand, it is crucial to address three challenges: The first challenge is to increase the EVs' penetration, for example, by facilitating access to charging infrastructure [627]. This is particularly important in densely populated areas where overnight charging may only be available for some EVs. The second challenge is to effectively control how EVs store energy [628]. Unlike conventional energy storage systems, EVs consume energy from the grid, and their batteries must be sufficiently charged by strict deadlines. The final challenge is to integrate EVs into the power grid infrastructure [628], enabling energy retailers to store energy in EVs that they can then inject back into the grid, i.e., an EV acts as a "virtual power plant."

Next, we discuss three optimization-related e-Mobility use cases, each addressing one of the challenges mentioned earlier. We start with a "core" mathematical problem and then extend this to a family of problems with increasing practical relevance, linking it to the problem classes discussed in Sec. IV.

Use case 1: Scheduling EVs in a parking site. This use case consists of allocating EVs to parking spaces in a parking site [629, 630]. In short, consider a parking site that has two types of parking spaces: regular parking spaces and parking spaces where EVs can be plugged in. EVs arrive at the parking site indicating the kWh they want to replenish at each time, e.g., in each hour. With that information, the parking site administrator decides where to allocate the EVs to maximize the charging volumes and thereby profit. To simplify the problem formulation even further, let us assume that the EVs arrival and departure times are known in advance. Figure 9 shows the problem schematically, where the y-axis in the figure represents the EVs. The goal is to select EVs such that the number of EVs at a given time does not exceed the number of chargers and the input power of the site.

We can formalize the problem as follows. Let N be the number of EVs and K the total number of intervals (e.g., 24 intervals, where each interval corresponds to an hour). Next, let $u_n \in \{0,1\}^K$ be a vector that indicates the (known) presence of EV $n \in \{1, \ldots, N\}$ at time $k \in \{0, \ldots, K-1\}$, and $v_n \in \mathbb{R}^K$ be the value that the parking site will obtain by admitting EV n (e.g., dollars). Thus, the inner product $v_n^T u_n$ denotes the total value for the parking site if admitting EV n. Also, let $d_n \in \mathbb{R}^K$, where the k-th entry indicates the energy requested by EV n at time k. The parking site has a maximum input power of E kWh and has a maximum capacity of M EVs. Then,
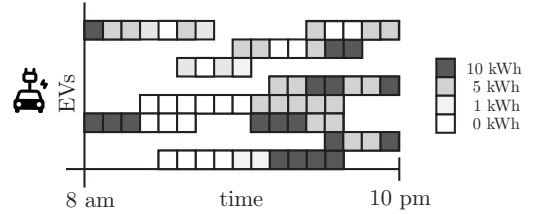
Figure 9. Illustrating a possible instance for the problem of scheduling the charging of EVs in a parking site (use case 1).

the optimization problem is the following:

$$\max_{x \in \{0,1\}^N} \quad \sum_{n=1}^{N} x_n (v_n^T u_n)$$
$$\text{s.t.} \quad \sum_{n=1}^{N} x_n u_{n,k} \leq M, \, \forall k, \qquad (34)$$
$$\sum_{n=1}^{N} x_n d_{n,k} u_{n,k} \leq E, \, \forall k,$$

where x_n denotes the decision variable of whether to accept EV n to a parking space where it can be plugged in or not. That is, Eq. (34) is a binary optimization with $2K$ knapsack constraints; K knapsacks for the parking spaces, and K knapsacks for the energy available in the parking site.

Some instances of the multi-dimensional knapsack problems can be challenging to solve, even when the problem instances are relatively small, cf. Sec. VI D. Thus, studying problems of the form in Eq. (34) can be of interest for near-term quantum optimization algorithms with a potential practical application. Of course, we need to benchmark specific practically relevant problem instances to assess whether they are difficult classically. If the instances prove difficult, we can proceed to explore whether there is an opportunity for a practically relevant quantum advantage. Since reality and real world scenarios are often more delicate, there will likely be additional constraints and cost function adjustments to Eq. (34) thereby increasing its general difficulty for classical solvers. Some interesting problem extensions for Eq. (34) towards practically relevant applications are the following:

1. The power available in a parking site may vary over time; for example, when the parking site shares its connection to the grid with another infrastructure or a local renewable energy source is considered, such as a solar power plant.

2. The decision of admitting/rejecting EVs is carried out via a pricing policy that affects the willingness of the drivers to use the parking site.

3. Increase the EV charging flexibility. EVs can have a desired state of charge at one or multiple points in time (e.g., at least 60 kWh by 6 am).

4. The EV arrivals and departure times may not be known in advance, as described at the beginning of the section. Additionally, the forecast of renewable energies includes uncertainties if considered in (1).

The original problem in Eq. (34) can be modeled as a QUBO. The original objective function is linear, i.e., the corresponding cost matrix will be a diagonal matrix, cf. Sec. IV A 1, and the constraints can, for instance, be encoded as penalties into the objective, cf. Sec. IV A 1, which, however, will immediately imply a dense cost matrix. The extensions (1)-(4) affect the difficulty of Eq. (34) in different ways: (1) imposes additional constraints that may result in an increasing problems size, which may result in more qubits or having to encode more binary variables per qubit. (2) will imply adding continuous variables and results in a MIP. (3) adds flexibility to how EVs charge and discharge, which may further increase the number of decision variables. Finally, (4) introduces uncertainty to the optimization problem, potentially necessitating a shift in the formulation from offline optimization to online or stochastic optimization such as MDP, cf. Sec. IV E.

Use case 2: Charging and discharging EVs. Consider the scenario where a fleet of EVs are plugged in from 6pm to 6am the next day, e.g., delivery vehicles. The fleet owner allows the energy supplier to use the EVs as energy storage subject to the constraint that the EVs are fully charged at departure time (i.e., by 6am the next day). The goal of the energy supplier is to provide electricity to the end customers at minimum cost. In particular, the energy supplier would like to store energy in the EVs when this is cheap, and use the energy in the EVs to power the grid when energy production is low or expensive to buy from the energy markets. The problem is challenging because the energy demand and energy prices may vary over time.

We can formulate the problem above as a *deterministic* discrete-time control problem, cf. Sec. IV E. In short, we divide time in slots $k \in \{0, \ldots, K-1\}$ of equal duration, where K defines the total considered time horizon. The duration of a slot can be, for example, 1 hour, and the horizon 12 hours. Next, let N be the number of EVs and $x_k \in \mathbb{R}_{\geq 0}^N$ represent the state of charge of the EVs at time k. In each time slot k, the controller selects an action from the action set $A(x_k) \subset \mathbb{R}^N$ containing the possible input/output powers of each EV, e.g., ± 10 kWh. The action set at time k depends on the state x_k since, for example, it is not possible to discharge an EV if its battery is empty. The choice of action $a_k \in A(x_k)$ affects the EVs' state of charge in the next time slot $k+1$. In particular, $x_{k+1} = F(x_k, a_k)$ where F is a function that deterministically maps a state x_k given an action a_k to the next state x_{k+1}. Next, let $T_k(x_k, a_k)$ be the cost obtained by selecting action a_k in state x_k at time k.

The resulting optimization problem is given as following:

$$
\begin{aligned}
\min_{a_k \in A(x_k)} \quad & \sum_{k=0}^{K-1} T_k(x_k, a_k) \\
\text{s.t.} \quad & x_K = x', \\
& x_{k+1} = F(x_k, a_k),
\end{aligned}
\tag{35}
$$

where $x_0 \in \mathbb{R}_{\geq 0}^N$ denotes the EVs' initial state of charge and x' the state that all EVs are fully charged. Thus, the goal is to select actions $a_k \in A(x_k)$ to minimize the sum of the costs given that $x_{k+1} = F(x_k, a_k)$ and the final state, x_K, is equal to x'. We have modeled the problem as a discrete optimal control problem, cf. Sec. IV E, but other formulations are possible as well. For instance, we can formulate it as a MIP with $K+1$ constraints for the dynamics, plus the constraints for enforcing feasibility of actions. However, unlike the previous use case, Eq. (35) is a less promising problem candidate for demonstrating industrial quantum advantage in the short term. The problem formulation can be mapped to a UCP (via DP) [614], which commercial solvers can solve efficiently with hundreds of units [631, Sec. 6]. Nonetheless, we can add features to Eq. (35) to make it more difficult to solve classically, and at the same time, closer to real-world scenarios:

1. Energy constraints that affect how an EV or group of EVs can charge or discharge at a given time.

2. EVs arriving and departing at different (stochastic) times instead of being parked overnight.

3. Stochastic cost function that captures, for example, the uncertainty in the electricity demand.

We can approach the problem in Eq. (35), for instance, with DP. As mentioned in Sec. IV D, DP requires searching over all possible states and actions, which is often exponential in size. One approach to speedup DP is using Grover's search algorithm as a subroutine. Similar to [338] for the TSP, the quantum algorithm could first use DP to compute multiple paths from state x_0 to $x_K = s'$ and then use Grover's search to find a combination of those paths that yields lower cost. Nonetheless, such an approach is unlikely to yield practical speedups in the short term since Grover's search requires FTQC and a quadratic speedup might not be sufficient for the considered problems and scales. An alternative is to model the problem as a MIP, and try algorithms discussed in Sec. IV C.

The extensions (1)-(3) add different complications to Eq. (35): (1-2) add constraints to enforce certain target states at given times or feasible actions, e.g., minimum constant charge or discharge times, which affects both, DP and MIP formulations as shown for uni-directional charging by Federer *et al.* [632]. In case of stochastic arrival/departure times, (2) will require more complex approaches, e.g. MDP, to handle the uncertainty. Similar

to the stochastic case in (2), (3) will require approaches like MDP to compute a policy that minimizes the costs under uncertainty. MDPs quickly become very difficult to solve due to the curse of dimensionality, cf. Sec. IV E. Thus, this might be an interesting domain for first small test cases to analyze the potential of corresponding quantum approaches. Since the result of MDP is a policy, the optimization and inference are split, which may allow one to relax requirements on the time-to-solution.

Use case 3: Energy retailing with storage. Consider an energy retailer that delivers energy to end customers. The energy supplier buys energy from energy producers in advance, e.g., on the day-ahead energy trading market, in one-hour time intervals, where the electricity price depends on several factors such as energy demand, renewable energy generation, etc. Additionally, the energy supplier has access to an energy storage system, e.g., a fleet of EVs that act as a *virtual power plant*, that can accumulate energy when this is cheap, to later use it when energy demand is high. The energy supplier's goal is to deliver energy to its customers at the lowest possible cost. Thus, it is incentivized to leverage the flexibility provided by its energy storage assets. The problem is challenging because the electricity demand and the energy available in storage are not exactly known in advance. Furthermore, if the energy demand is not met, the energy supplier has to buy energy from other retailers that often sell energy at a higher price.

We can formulate the problem mathematically as follows. Let $e_k \in \mathbb{R}_{\geq 0}$ be the decision variable indicating the MWh the energy supplier buys in each hour $k \in \{0, 1, \ldots, K-1\}$ in advance, usually with $K = 24$, and $p_k^{\text{low}} \in \mathbb{R}$ the price of the *cheap* energy (which sometimes might even be negative!). Similarly, let D_k, B_k, P_k^{high} be random variables that capture, respectively, the electricity demand at time k in MWh, the energy available in storage at time k in MWh, and the price of the *expensive* energy at time k. The optimization problem is the following:

$$
\min_{e_k \geq 0} \quad \sum_{k=0}^{K-1} \left(p_k^{\text{low}} e_k + \mathbf{E}\left[P_k^{\text{high}} [D_k - e_k - B_k]^+ \right] \right)
\tag{36}
$$

where $[\cdot]^+ := \max\{0, \cdot\}$ and the expectation is with respect to D_k, B_k, and P_k^{high}. That is, we have a stochastic unconstrained optimization where the objective function at time k consists of two terms: the cost of buying cheap energy, and the expected cost of buying expensive energy. Note that $[D_k - e_k - B_k]^+$ captures the energy demand that cannot be met with the energy bought (i.e., e_k) and the energy in storage (i.e., B_k).

The optimization in Eq. (36) is convex since the objective function is piece-wise linear and convex. Thus, if D_k, B_k, and P_k^{high} are random variables *independent* of the decision variables, we can efficiently solve the problem with conventional convex optimization tools, in fact, by solving K separate problems — one for every term in

the sum in Eq. (36). Hence, Eq. (36), in its current form, is *not* a promising problem candidate for demonstrating quantum advantage. Yet, we can make it more challenging and closer to real-world problems with the following three extensions:

1. The energy available in storage at time k depends on the previous decisions taken; for example, on how EVs charge and discharge, as in the previous use case 2.

2. Convex or concave cost functions, i.e., the cost to buy energy might depend on the quantity ordered.

3. The energy supply has to meet the energy demand.

The extensions add different layers of complexity. In particular, (1) *couples* the random variables with the decision variables and "links" the K time steps, i.e., terms in the objective function Eq. (36). This is similar to the system dynamics function in optimal control, cf. Sec. IV E. This extension is important since assuming that the energy in storage (i.e., B_k) is independent of the decision variables does not allow us to capture, for example, that the energy retailer buys energy in advance to keep in storage for later use, i.e., the decisions in the past affect the events in the future. We can recast Eq. (36) as an optimal control problem where the system dynamics are given by the algorithm in the second use case, i.e., the energy available by how EVs charge and discharge. That is, we are coupling this use case with the previous one, and, indirectly, inheriting the quantum challenges already discussed. In other words, we are adding the complications of use case 2 on top of use case 3. (2) adds a constraint of the form $g(D_k - e_k - B_k) \leq 0$, where g is a function such as CVaR that measures risk. The problem falls within the realm of robust optimization, see Sec. IV F, or chance constraints [633]. In case of (3), depending on the exact properties of cost functions, the problem may remain convex or not. In case of a non-convex problem, this can quickly become very challenging to be solved.

4.   Outlook

Quantum computing has the potential to significantly advance optimization problems related to the sustainable energy transition. However, identifying the most promising applications is an ongoing challenge. Here, we have introduced illustrative examples of optimization problems that may serve as test cases derived from real-world data. Under certain assumptions on the problem data, some instances may prove to be classically challenging even with few variables. We have shown how the problem classes introduced in Sec. IV relate to these families of use cases with increasing complexity and of increasing relevance to real-world applications. The discussion underlines the need for additional research on quantum optimization algorithms to approach the complications

required for practical impact. It further shows that it is crucial to benchmark the problem instances to identify which ones are truly difficult classically and where there is room for a potential quantum advantage.

VIII.   CONCLUSION & OUTLOOK

In this paper, we provided a comprehensive overview of the potential, challenges, and emerging research areas in quantum optimization. We observed that, while complexity theory is useful to guide towards provable performance guarantees, it may not always be useful for finding practical quantum advantage. This underscores the need to develop and analyze quantum optimization heuristics to better understand their effectiveness. We highlighted key problem classes in optimization, reviewed existing algorithms and suggested new research directions. Further, we discussed the challenges of executing and scaling these algorithms on noisy hardware and the importance of benchmarking for identifying quantum advantages. Lastly, we explored two illustrative application domains, emphasizing the need to expand our focus beyond problem classes like QUBO to truly impact these fields.

As quantum hardware evolves with more qubits, reduced errors, and faster circuit execution, a new era is emerging where progress in quantum algorithm research comes from both theoretical analysis and empirical methods. While there remains a strong need to discover new algorithms with provable performance guarantees, we must also prioritize the development of quantum algorithms with no such guarantees, and benchmark their performance on real quantum computers. The improving abilities to test ideas in practice unlocks unprecedented opportunities for the advancement of quantum optimization. Even if it does not immediately imply a practical quantum advantage, the intuition gained from running experiments on real hardware empowers one to quickly validate proposals.

In addition to technical advances in quantum optimization, there is also the question of responsible research and use of this new technology. There are a variety of causes, arising from different applications. Thus, further research should be done with the awareness that optimization use cases occur in a multitude of contexts — from those with positive societal impact, to those with nefarious intent. We hope that open scientific discussions encourage advancement in quantum optimization and advocate responsible use of the insights presented in this work. While we showcase where benefits for optimization algorithms may lie, it is important not to overstate or misinterpret the results of this paper and its *potential* applications. Such considerations motivate the need to establish clear benchmarks, cf. Sec. VI, which enable a reliable interpretation of the scientific insights by a broader audience. Moreover, where use case applications of quantum optimization algorithms are chosen and funded, we

encourage the prioritization of those with positive social impact, for example, sustainable energy transition as discussed in Sec. VII B.

In conclusion, quantum optimization holds vast potential for various applications, but significant challenges remain in demonstrating a practical quantum advantage. Once a tangible quantum advantage is demonstrated, we expect that quantum optimization would rapidly influence many domains given the widespread relevance of optimization. This paper has provided a blueprint to define and measure progress in quantum optimization, which unquestionably has exciting prospects.

[1] Lov K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (Association for Computing Machinery, New York, NY, USA, 1996) p. 212–219.

[2] Tadashi Kadowaki and Hidetoshi Nishimori, "Quantum annealing in the transverse Ising model," Physical Review E **58**, 5355 (1998).

[3] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser, "Quantum computation by adiabatic evolution," (2000), arXiv:quant-ph/0001106.

[4] Maciej Liśkiewicz and Martin R Schuster, "A new upper bound for the traveling salesman problem in cubic graphs," Journal of Discrete Algorithms **27**, 1–20 (2014).

[5] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook, "On the solution of traveling salesman problems." Documenta Mathematica , 645–656 (1998).

[6] David L. Applegate, Robert E. Bixby, Vasek Chvátal, and William J. Cook, "Finding Tours in the TSP," (1999).

[7] William J Cook, William H Cunningham, William R Pulleyblank, and Alexander Schrijver, "Wiley-interscience series in discrete mathematics and optimization," in *Combinatorial Optimization* (John Wiley & Sons, Ltd, 1997) pp. 356–356.

[8] William J Cook, David L Applegate, Robert E Bixby, and Vasek Chvatal, *The traveling salesman problem: a computational study* (Princeton university press, 2011).

[9] William J Cook, *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation* (Princeton university press, 2012).

[10] "3D Star TSPs," (2023), accessed Dec 3, 2023.

[11] Jakob Puchinger, Günther R. Raidl, and Ulrich Pferschy, "The Multidimensional Knapsack Problem: Structure and Algorithms," INFORMS J. on Computing **22**, 250–265 (2010).

[12] Tom Packebusch and Stephan Mertens, "Low autocorrelation binary sequences," Journal of Physics A: Mathematical and Theoretical **49**, 165001 (2016).

[13] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder, "High-threshold and low-overhead fault-tolerant quantum memory," (2023), arXiv:2308.07915 [quant-ph].

[14] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta, "Error mitigation for short-depth quantum circuits," Physical Review Letters **119** (2017).

[15] Ewout van den Berg, Zlatko K. Minev, Abhinav Kandala, and Kristan Temme, "Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors," Nature Physics **19**, 1116–1121 (2023).

[16] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala, "Evidence for the utility of quantum computing before fault tolerance," Nature **618**, 500–505 (2023).

[17] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.

[18] Craig Gidney and Martin Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," Quantum **5**, 433 (2021).

[19] Sanjeev Arora and Boaz Barak, *Computational Complexity: A Modern Approach* (Cambridge University Press, 2009).

[20] John Watrous, "Quantum Computational Complexity," in *Encyclopedia of Complexity and Systems Science*, edited by Robert A. Meyers (Springer New York, New York, NY, 2009) pp. 7174–7201.

[21] Ronald de Wolf, "Quantum computing: Lecture notes," (2023), arXiv:1907.09415 [quant-ph].

[22] Richard Bellman, "Dynamic programming treatment of the travelling salesman problem," J. ACM **9**, 61–63 (1962).

[23] Michael Held and Richard M. Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of the Society for Industrial and Applied Mathematics **10**, 196–210 (1962).

[24] Andreas Björklund, Petteri Kaski, and Lukasz Kowalik, "Fast Witness Extraction using a Decision Oracle," in *Algorithms - ESA 2014/Lecture notes in computer science*, Vol. 8737, edited by Andreas Schulz and Dorothea Wagner (Springer, 2014) pp. 149–160.

[25] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi, *Complexity and approximation: Combinatorial optimization problems and their approximability properties* (Springer Science & Business Media, 2012).

[26] Ethan Bernstein and Umesh Vazirani, "Quantum complexity theory," SIAM Journal on Computing **26**, 1411–1473 (1997).

[27] Takashi Yamakawa and Mark Zhandry, "Verifiable quantum advantage without structure," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022) pp. 69–74.

[28] Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner, "Quantum depth in the random oracle model," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (2023) pp. 1111–1124.

[29] Richard M. Karp, "Reducibility among combinatorial

problems," in *Complexity of Computer Computations* (Springer US, 1972) pp. 85–103.

[30] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf, "Quantum lower bounds by polynomials," Journal of the ACM (JACM) **48**, 778–797 (2001).

[31] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, "Strengths and weaknesses of quantum computing," SIAM journal on Computing **26**, 1510–1523 (1997).

[32] Scott Aaronson, "How much structure is needed for huge quantum speedups?" (2022), talk delivered at the 28th Solvay Physics Conference, arXiv:2209.06930 [quant-ph].

[33] Sevag Gharibian, "The 7 faces of quantum NP," (2023), arXiv:2310.18010 [quant-ph].

[34] There is a slight technical complication here, because we deal with a class that has a promise: either there exists a quantum proof that makes the quantum verifier accept with high probability or for every proof the verifier rejects with high probability. In the strict sense such a promise class is not known to contain complete problems. In order to still use the notion of completeness a framework of promise problems is introduced [634] for additional technical details.

[35] 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, "Quantum computational advantage with a programmable photonic processor," Nature **606**, 75–81 (2022).

[36] Chris Marriott and John Watrous, "Quantum arthur–merlin games," computational complexity **14**, 122–152 (2005).

[37] Sanjeev Arora and Boaz Barak, *Computational Complexity* (Cambridge University Press, 2009).

[38] Ran Raz and Avishay Tal, "Oracle separation of bqp and ph," ACM Journal of the ACM (JACM) **69**, 1–21 (2022).

[39] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li, "The complexity of NISQ," Nature Communications **14**, 6001 (2023).

[40] Harry Buhrman, Subhasree Patro, and Florian Speelman, "A framework of quantum strong exponential-time hypotheses," in *38th International symposium on theoretical aspects of computer science (stacs 2021)* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021).

[41] Michael R. Garey and David S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., USA, 1990).

[42] R. G. Downey and M. R. Fellows, *Parameterized Complexity* (Springer New York, 1999).

[43] Liming Cai and Jianer Chen, "On fixed-parameter tractability and approximability of np optimization problems," Journal of Computer and System Sciences **54**, 465–474 (1997).

[44] Jianer Chen, Iyad A. Kanj, and Ge Xia, "Improved upper bounds for vertex cover," Theoretical Computer Science **411**, 3736–3756 (2010).

[45] Michael J. Bremner, Zhengfeng Ji, Ryan L. Mann, Luke Mathieson, Mauro E. S. Morales, and Alexis T. E.

[46] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi, *Kernelization: theory of parameterized preprocessing* (Cambridge University Press, 2019).

[47] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos, "(meta) kernelization," J. ACM **63** (2016), 10.1145/2973749.

[48] David S. Johnson, "Approximation algorithms for combinatorial problems," Journal of Computer and System Sciences **9**, 256–278 (1974).

[49] Johan Håstad, "Some optimal inapproximability results," Journal of the ACM **48**, 798–859 (2001).

[50] Michel X. Goemans and David P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," J. ACM **42**, 1115–1145 (1995).

[51] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell, "Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?" SIAM Journal on Computing **37**, 319–357 (2007).

[52] Subhash Khot, "On the power of unique 2-prover 1-round games," in *34th Annual ACM Symposium on Theory of Computing STOC'02* (2002) pp. 767–775.

[53] Carsten Lund and Mihalis Yannakakis, "On the hardness of approximating minimization problems," Journal of the ACM **41**, 960–981 (1994).

[54] Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani, "The detectability lemma and quantum gap amplification," (2008), arXiv:0811.3412 [quant-ph].

[55] Dorit Aharonov and Tomer Naveh, "Quantum np - a survey," (2002), arXiv:quant-ph/0210077 [quant-ph].

[56] Dorit Aharonov, Itai Arad, and Thomas Vidick, "Guest column: The quantum pcp conjecture," SIGACT News **44**, 47–79 (2013).

[57] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi, *Classical and quantum computation*, Vol. 47 (American Mathematical Soc., 2002).

[58] Anurag Anshu, Nikolas P. Breuckmann, and Chinmay Nirkhe, "Nlts hamiltonians from good quantum codes," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023 (Association for Computing Machinery, New York, NY, USA, 2023) p. 1090–1096.

[59] Marek Karpinski, Michael Lampis, and Richard Schmied, "New inapproximability bounds for tsp," Journal of Computer and System Sciences **81**, 1665–1677 (2015).

[60] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan, "A (slightly) improved approximation algorithm for metric tsp," in *53rd Annual ACM SIGACT Symposium on Theory of Computing STOC'21* (2021) pp. 32–45.

[61] Arthur Lee and Bruce Xu, "Classifying approximation algorithms: Understanding the apx complexity class," (2021), arXiv:2111.01551 [cs.CC].

[62] Teofilo F. Gonzalez, ed., *Handbook of Approximation Algorithms and Metaheuristics: Methologies and Traditional Applications*, Vol. 1 (CRC Press, 2018).

[63] Sanjeev Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," Journal of the ACM (JACM) **45**, 753–782 (1998).

[64] Daniel A Spielman and Shang-Hua Teng, "Smoothed

analysis: an attempt to explain the behavior of algorithms in practice," Communications of the ACM **52**, 76–84 (2009).

[65] Andrej Bogdanov, Luca Trevisan, *et al.*, "Average-case complexity," Foundations and Trends® in Theoretical Computer Science **2**, 1–106 (2006).

[66] Niklas Pirnay, Vincent Ulitzsch, Frederik Wilde, Jens Eisert, and Jean-Pierre Seifert, "An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems," (2022), arXiv:2212.08678.

[67] Mario Szegedy, "Quantum advantage for combinatorial optimization problems, Simplified," (2022), arXiv:2212.12572 [quant-ph].

[68] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan, "Average-case fine-grained hardness," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017) pp. 483–496.

[69] Kaining Zhang, Liu Liu, Min-Hsiu Hsieh, and Dacheng Tao, "Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits," Advances in Neural Information Processing Systems **35**, 18612–18627 (2022).

[70] Enrico Fontana, Dylan Herman, Shouvanik Chakrabarti, Niraj Kumar, Romina Yalovetzky, Jamie Heredge, Shree Hari Sureshbabu, and Marco Pistoia, "The adjoint is all you need: Characterizing barren plateaus in quantum ansätze," (2023), arXiv:2309.07902 [quant-ph].

[71] Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martin Larocca, and M. Cerezo, "A unified theory of barren plateaus for deep parametrized quantum circuits," arXiv preprint - arXiv:2309.09342 (2023).

[72] Jarrod McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nature Communications **9** (2018).

[73] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," Nature Communications **12** (2021), 10.1038/s41467-021-21728-w.

[74] M Cerezo and Patrick J Coles, "Higher order derivatives of quantum neural networks with barren plateaus," Quantum Science and Technology **6** (2021).

[75] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," PRX Quantum **3** (2022), 10.1103/PRXQuantum.3.010313.

[76] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe, "Entanglement-induced barren plateaus," PRX Quantum **2**, 040316 (2021).

[77] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles, "Noise-induced barren plateaus in variational quantum algorithms," Nature Communications **12** (2021).

[78] John Napp, "Quantifying the barren plateau phenomenon for a model of unstructured variational ansätze," (2022), arXiv:2203.06174 [quant-ph].

[79] A V Uvarov and J D Biamonte, "On barren plateaus and cost function locality in variational quantum algorithms," Journal of Physics A: Mathematical and Theoretical **54**, 245301 (2021).

[80] Alistair Letcher, Stefan Woerner, and Christa Zoufal, "From tight gradient bounds for parameterized quantum circuits to the absence of barren plateaus in qgans," (2023), arXiv:2309.12681 [quant-ph].

[81] Christos H. Papadimitriou and Mihalis Yannakakis, "Optimization, approximation, and complexity classes," Journal of Computer and System Sciences **43**, 425–440 (1991).

[82] Mark W. Krentel, "The complexity of optimization problems," Journal of Computer and System Sciences **36**, 490–509 (1988).

[83] Martin Dyer and Leen Stougie, "Computational complexity of stochastic programming problems," Mathematical Programming **106**, 423–432 (2006).

[84] Leslie G Valiant, "The complexity of computing the permanent," Theoretical computer science **8**, 189–201 (1979).

[85] Christos H. Papadimitriou, "Games against nature," Journal of Computer and System Sciences **31**, 288–301 (1985).

[86] Yunseok Lee, Holger Boche, and Gitta Kutyniok, "Computability of optimizers," (2023), arXiv:2301.06148 [math.OC].

[87] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale, "Complexity and real computation: A manifesto," International Journal of Bifurcation and Chaos **6**, 3–26 (1996).

[88] Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore, and Matthias Troyer, "Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm," Quantum **4**, 287 (2020).

[89] David Layden, Guglielmo Mazzola, Ryan V Mishmash, Mario Motta, Pawel Wocjan, Jin-Sung Kim, and Sarah Sheldon, "Quantum-enhanced Markov chain Monte Carlo," Nature **619**, 282–287 (2023).

[90] Philip Intallura, Georgios Korpas, Sudeepto Chakraborty, Vyacheslav Kungurtsev, and Jakub Marecek, "A survey of quantum alternatives to randomized algorithms: Monte carlo integration and beyond," (2023), arXiv:2303.04945 [quant-ph].

[91] Sami Boulebnane and Ashley Montanaro, "Solving boolean satisfiability problems with the quantum approximate optimization algorithm," arXiv preprint arXiv:2208.06909 (2022).

[92] David Gamarnik, "The overlap gap property: A topological barrier to optimizing over random structures," Proceedings of the National Academy of Sciences **118**, e2108492118 (2021).

[93] David Gamarnik, Aukosh Jagannath, and Alexander S Wein, "Hardness of random optimization problems for boolean circuits, low-degree polynomials, and langevin dynamics," arXiv preprint arXiv:2004.12063 (2020).

[94] Brice Huang and Mark Sellke, "Tight lipschitz hardness for optimizing mean field spin glasses," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022) pp. 312–322.

[95] Joao Basso, David Gamarnik, Song Mei, and Leo Zhou, "Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022) pp. 335–343.

[96] Antares Chen, Neng Huang, and Kunal Marwaha, "Local algorithms and the failure of log-depth quantum advantage on sparse random csps," arXiv preprint arXiv:2310.01563 (2023).

[97] Christoph Durr and Peter Hoyer, "A quantum algorithm for finding the minimum," (1999), arXiv:quant-ph/9607014 [quant-ph].

[98] D. Bulger, W. P. Baritompa, and G. R. Wood, "Implementing pure adaptive search with grover's quantum algorithm," Journal of Optimization Theory and Applications **116**, 517–529 (2003).

[99] W. P. Baritompa, D. W. Bulger, and G. R. Wood, "Grover's quantum algorithm applied to global optimization," SIAM Journal on Optimization **15**, 1170–1184 (2005).

[100] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandao, and Garnet Kin-Lic Chan, "Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution," Nature Physics **16**, 205–210 (2020).

[101] Austin Gilliam, Stefan Woerner, and Constantin Gonciulea, "Grover adaptive search for constrained polynomial binary optimization," Quantum **5**, 428 (2021).

[102] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, "Strengths and weaknesses of quantum computing," SIAM Journal on Computing **26**, 1510–1523 (1997).

[103] Chris Cade, Marten Folkertsma, Ido Niesen, and Jordi Weggemans, "Quantifying grover speed-ups beyond asymptotic analysis," Quantum **7**, 1133 (2023).

[104] Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, Yuri Alexeev, Joan M. Dreiling, John P. Gaebler, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Nathan Hewitt, Chandler V. Horst, Shaohan Hu, Jacob Johansen, Mitchell Matheny, Tanner Mengle, Michael Mills, Steven A. Moses, Brian Neyenhuis, Peter Siegfried, Romina Yalovetzky, and Marco Pistoia, "Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem," (2023), arXiv:2308.02342 [quant-ph].

[105] Andrew J Sommese, Charles W Wampler, *et al.*, *The Numerical solution of systems of polynomials arising in engineering and science* (World Scientific, 2005).

[106] M. Born and V. Fock, "Beweis des Adiabatensatzes," Zeitschrift fur Physik **51**, 165–180 (1928).

[107] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler, "Bounds for the adiabatic approximation with applications to quantum computation," Journal of Mathematical Physics **48** (2007).

[108] Daniel A Lidar, Ali T Rezakhani, and Alioscia Hamma, "Adiabatic approximation with exponential accuracy for many-body systems and quantum computation," Journal of Mathematical Physics **50** (2009).

[109] Donny Cheung, Peter Høyer, and Nathan Wiebe, "Improved error bounds for the adiabatic approximation," Journal of Physics A: Mathematical and Theoretical **44**, 415302 (2011).

[110] A.B. Finnila, M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," Chemical Physics Letters **219**, 343–348 (1994).

[111] Victor Bapst, Laura Foini, Florent Krzakala, Guilhem Semerjian, and Francesco Zamponi, "The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective," Physics Reports **523**, 127–205 (2013).

[112] Sergey Knysh, "Zero-temperature quantum annealing bottlenecks in the spin-glass phase," Nature communications **7**, 12370 (2016).

[113] Sergei V Isakov, Guglielmo Mazzola, Vadim N Smelyanskiy, Zhang Jiang, Sergio Boixo, Hartmut Neven, and Matthias Troyer, "Understanding quantum tunneling through quantum monte carlo simulations," Physical review letters **117**, 180402 (2016).

[114] Naeimeh Mohseni, Carlos Navarrete-Benlloch, Tim Byrnes, and Florian Marquardt, "Deep recurrent networks predicting the gap evolution in adiabatic quantum computing," Quantum **7**, 1039 (2023).

[115] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer, "Defining and detecting quantum speedup," Science **345**, 420–424 (2014).

[116] EJ Crosson and DA Lidar, "Prospects for quantum enhancement with diabatic quantum annealing," Nature Reviews Physics **3**, 466–489 (2021).

[117] Mustafa Demirplak and Stuart A. Rice, "Adiabatic population transfer with control fields," The Journal of Physical Chemistry A **107**, 9937–9945 (2003).

[118] Adolfo del Campo, "Shortcuts to adiabaticity by counterdiabatic driving," Phys. Rev. Lett. **111**, 100502 (2013).

[119] Pieter W. Claeys, Mohit Pandey, Dries Sels, and Anatoli Polkovnikov, "Floquet-engineering counterdiabatic protocols in quantum many-body systems," Phys. Rev. Lett. **123**, 090602 (2019).

[120] T. Chasseur, L. S. Theis, Y. R. Sanders, D. J. Egger, and F. K. Wilhelm, "Engineering adiabaticity at an avoided crossing with optimal control," Phys. Rev. A **91**, 043421 (2015).

[121] Hale F Trotter, "On the product of semi-groups of operators," Proceedings of the American Mathematical Society **10**, 545–551 (1959).

[122] Masuo Suzuki, "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations," Physics Letters A **146**, 319–323 (1990).

[123] Masuo Suzuki, "General theory of fractal path integrals with applications to many-body theories and statistical physics," Journal of Mathematical Physics **32**, 400–407 (1991).

[124] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[125] Daniel S. Abrams and Seth Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," Phys. Rev. Lett. **83**, 5162–5165 (1999).

[126] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon, "Simulated quantum computation of molecular energies," Science **309**, 1704–1707 (2005), https://www.science.org/doi/pdf/10.1126/science.1113479.

[127] Seunghoon Lee, Joonho Lee, Huanchen Zhai, Yu Tong,

Alexander M. Dalzell, Ashutosh Kumar, Phillip Helms, Johnnie Gray, Zhi-Hao Cui, Wenyuan Liu, Michael Kastoryano, Ryan Babbush, John Preskill, David R. Reichman, Earl T. Campbell, Edward F. Valeev, Lin Lin, and Garnet Kin-Lic Chan, "Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry," Nature Communications 14, 1952 (2023).

[128] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in Proceedings 35th Annual Symposium on Foundations of Computer Science (1994) pp. 124–134.

[129] Shuxian Jiang, Keith A. Britt, Alexander J. McCaskey, Travis S. Humble, and Sabre Kais, "Quantum annealing for prime factorization," Scientific Reports 8 (2018), 10.1038/s41598-018-36058-z.

[130] Vivian Phan, Arttu Pönni, Matti Raasakka, and Ilkka Tittonen, "On quantum factoring using noisy intermediate scale quantum computers," (2022), arXiv:2208.07085 [quant-ph].

[131] Kyungtaek Jun and Hyunju Lee, "HUBO and QUBO models for prime factorization," Scientific Reports 13 (2023), 10.1038/s41598-023-36813-x.

[132] Niklas Pirnay, Vincent Ulitzsch, Frederik Wilde, Jens Eisert, and Jean-Pierre Seifert, "An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems," (2023), arXiv:2212.08678 [quant-ph].

[133] Philippe Suchsland, Francesco Tacchino, Mark H. Fischer, Titus Neupert, Panagiotis Kl. Barkoutsos, and Ivano Tavernelli, "Algorithmic error mitigation scheme for current quantum processors," Quantum 5, 492 (2021).

[134] William Kirby, Mario Motta, and Antonio Mezzacapo, "Exact and efficient lanczos method on a quantum computer," Quantum 7, 1018 (2023).

[135] Seth Lloyd, "Universal quantum simulators," Science 273, 1073–1078 (1996).

[136] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman, "Exponential algorithmic speedup by a quantum walk," in Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03 (Association for Computing Machinery, New York, NY, USA, 2003) p. 59–68.

[137] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders, "Efficient quantum algorithms for simulating sparse hamiltonians," Communications in Mathematical Physics 270, 359–371 (2007).

[138] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. 103, 150502 (2009).

[139] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Simulating hamiltonian dynamics with a truncated taylor series," Phys. Rev. Lett. 114, 090502 (2015).

[140] Andrew M Childs, Robin Kothari, and Rolando D Somma, "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision," SIAM Journal on Computing 46, 1920–1950 (2017).

[141] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic linear algebra subprograms for fortran usage," ACM Trans. Math. Softw. 5, 308–323 (1979).

[142] Guang Hao Low and Isaac L. Chuang, "Optimal hamiltonian simulation by quantum signal processing," Phys. Rev. Lett. 118, 010501 (2017).

[143] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics," in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (2019) pp. 193–204.

[144] Stuart Geman and Donald Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6, 721–741 (1984).

[145] Florent Krzakała, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová, "Gibbs states and the set of solutions of random constraint satisfaction problems," Proceedings of the National Academy of Sciences 104, 10318–10323 (2007).

[146] Larry Stockmeyer, "The complexity of approximate counting," in Proceedings of the fifteenth annual ACM symposium on Theory of computing - STOC '83, STOC '83 (ACM Press, 1983).

[147] Daniel Zhang, Jan Lukas Bosse, and Toby Cubitt, "Dissipative quantum gibbs sampling," (2023), arXiv:2304.04526 [quant-ph].

[148] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller, "Equation of state calculations by fast computing machines," The Journal of Chemical Physics 21, 1087–1092 (1953).

[149] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," Biometrika 57, 97–109 (1970).

[150] David Poulin and Pawel Wocjan, "Sampling from the thermal quantum gibbs state and evaluating partition functions with a quantum computer," Physical review letters 103, 220502 (2009).

[151] Dorit Aharonov, Itai Arad, and Thomas Vidick, "The quantum pcp conjecture," (2013), arXiv:1309.7495 [quant-ph].

[152] Sergey Bravyi, Anirban Chowdhury, David Gosset, and Pawel Wocjan, "Quantum Hamiltonian complexity in thermal equilibrium," Nature Physics 18, 1367–1370 (2022).

[153] Barbara M Terhal and David P DiVincenzo, "Problem of equilibration and the computation of correlation functions on a quantum computer," Physical Review A 61, 022301 (2000).

[154] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, "Quantum metropolis sampling," Nature 471, 87–90 (2011).

[155] Man-Hong Yung and Alán Aspuru-Guzik, "A quantum–quantum metropolis algorithm," Proceedings of the National Academy of Sciences 109, 754–759 (2012).

[156] Cambyse Rouzé, Daniel Stilck França, and Álvaro M. Alhambra, "Efficient thermalization and universal quantum computing with quantum gibbs samplers," (2024), arXiv:2403.12691 [quant-ph].

[157] Ainesh Bakshi, Allen Liu, Ankur Moitra, and Ewin Tang, "High-temperature gibbs states are unentangled and efficiently preparable," (2024), arXiv:2403.16850 [quant-ph].

[158] Jingxiang Wu and Timothy H. Hsieh, "Variational thermal quantum simulation via thermofield double states," Phys. Rev. Lett. 123, 220502 (2019).

[159] Youle Wang, Guangxi Li, and Xin Wang, "Variational quantum gibbs state preparation with a truncated taylor series," Phys. Rev. Appl. **16**, 054035 (2021).

[160] Mirko Consiglio, Jacopo Settino, Andrea Giordano, Carlo Mastroianni, Francesco Plastina, Salvatore Lorenzo, Sabrina Maniscalco, John Goold, and Tony J. G. Apollaro, "Variational gibbs state preparation on nisq devices," (2023), arXiv:2303.11276 [quant-ph].

[161] Ada Warren, Linghua Zhu, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou, "Adaptive variational algorithms for quantum gibbs state preparation," (2022), arXiv:2203.12757 [quant-ph].

[162] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner, "Variational quantum boltzmann machines," Quantum Machine Intelligence **3**, 7 (2021).

[163] Dominik S. Wild, Dries Sels, Hannes Pichler, Cristian Zanoci, and Mikhail D. Lukin, "Quantum sampling algorithms for near-term devices," Phys. Rev. Lett. **127**, 100504 (2021).

[164] F. G. S. L. Brandao and K. M. Svore, "Quantum speedups for solving semidefinite programs," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017) pp. 415–426.

[165] Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, "Quantum SDP-solvers: Better upper and lower bounds," Quantum **4**, 230 (2020).

[166] F. G. S. L. Brandao, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, K. M. Svore, and Xiaodi Wu, "Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning," in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Vol. 132, edited by Christel Baier, Paola Flocchini, and Stefano Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019) p. 27.

[167] Joran van Apeldoorn and András Gilyén, "Improvements in quantum sdp-solving with applications," in *Proceedings of 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019* (2019).

[168] Sergey Bravyi, Anirban Chowdhury, David Gosset, Vojtech Havlicek, and Guanyu Zhu, "Quantum complexity of the kronecker coefficients," (2023), arXiv:2302.11454 [quant-ph].

[169] Ashley Montanaro, "Quantum speedup of monte carlo methods," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **471**, 20150301 (2015).

[170] Aram W Harrow and Annie Y Wei, "Adaptive quantum simulated annealing for bayesian inference and estimating partition functions," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, 2020) pp. 193–212.

[171] Srinivasan Arunachalam, Vojtech Havlicek, Giacomo Nannicini, Kristan Temme, and Pawel Wocjan, "Simpler (classical) and faster (quantum) algorithms for gibbs partition functions," Quantum **6**, 789 (2022).

[172] Arjan Cornelissen and Yassine Hamoudi, "A sublinear-time quantum algorithm for approximating partition functions," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, 2023) pp. 1245–1264.

[173] Fernando GS L Brandao, Richard Kueng, and Daniel Stilck França, "Faster quantum and classical sdp approximations for quadratic binary optimization,"

[174] Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis Zuluaga, "Solving the semidefinite relaxation of qubos in matrix multiplication time, and faster with a quantum computer," (2023), arXiv:2301.04237 [quant-ph].

[175] Joran van Apeldoorn and András Gilyén, "Quantum algorithms for zero-sum games," (2019), arXiv:1904.03180 [quant-ph].

[176] Adam Bouland, Yosheb M Getachew, Yujia Jin, Aaron Sidford, and Kevin Tian, "Quantum speedups for zero-sum games via improved dynamic gibbs sampling," in *International Conference on Machine Learning* (PMLR, 2023) pp. 2932–2952.

[177] Minbo Gao, Zhengfeng Ji, Tongyang Li, and Qisheng Wang, "Logarithmic-regret quantum learning algorithms for zero-sum games," (2023), arXiv:2304.14197 [quant-ph].

[178] Li Li, Minjie Fan, Marc Coram, Patrick Riley, and Stefan Leichenauer, "Quantum optimization with a novel gibbs objective function and ansatz architecture search," Phys. Rev. Res. **2**, 023074 (2020).

[179] Michel X. Goemans and David P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," J. ACM **42**, 1115–1145 (1995).

[180] Taylor L. Patti, Jean Kossaifi, Anima Anandkumar, and Susanne F. Yelin, "Quantum goemans-williamson algorithm with the hadamard test and approximate amplitude constraints," Quantum **7**, 1057 (2023).

[181] Fernando G.S L. Brandã o, Richard Kueng, and Daniel Stilck França, "Faster quantum and classical SDP approximations for quadratic binary optimization," Quantum **6**, 625 (2022).

[182] Ewin Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (ACM, 2019).

[183] Nai-Hui Chia, András Gilyén, Han-Hsuan Lin, Seth Lloyd, Ewin Tang, and Chunhao Wang, "Quantum-Inspired Algorithms for Solving Low-Rank Linear Equation Systems with Logarithmic Dependence on the Dimension," in *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 181, edited by Yixin Cao, Siu-Wing Cheng, and Minming Li (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2020) pp. 47:1–47:17.

[184] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," Nature Communications **5** (2014), 10.1038/ncomms5213.

[185] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, "A quantum approximate optimization algorithm," (2014), arXiv:1411.4028.

[186] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," New Journal of Physics **18**, 023023 (2016).

[187] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, *et al.*, "Variational quantum algorithms," Nature Re-

views Physics **3**, 625–644 (2021).

[188] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, *et al.*, "Noisy intermediate-scale quantum (nisq) algorithms (2021)," arXiv preprint arXiv:2101.08448 (2021).

[189] Lennart Bittel and Martin Kliesch, "Training variational quantum algorithms is NP-hard," Physical Review Letters **127** (2021), 10.1103/physrevlett.127.120502.

[190] John Golden, Andreas Bärtschi, Stephan Eidenbenz, and Daniel O'Malley, "Numerical Evidence for Exponential Speed-up of QAOA over Unstructured Search for Approximate Constrained Optimization," in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (2023) pp. 496–505.

[191] Avrim L. Blum and Ronald L. Rivest, "Training a 3-node neural network is NP-complete," Neural Networks **5**, 117–127 (1992).

[192] Sammy Khalife, Hongyu Cheng, and Amitabh Basu, "Neural networks with linear threshold activations: structure and algorithms," Mathematical Programming (2023), 10.1007/s10107-023-02016-5.

[193] Vincent Froese and Christoph Hertrich, "Training neural networks is np-hard in fixed dimension," (2023), arXiv:2303.17045 [cs.CC].

[194] Lennart Binkowski, Gereon Koßmann, Timo Ziegler, and René Schwonnek, "Elementary proof of qaoa convergence," (2023), arXiv:2302.04968 [quant-ph].

[195] Vyacheslav Kungurtsev, Georgios Korpas, Jakub Marecek, and Elton Yechao Zhu, "Iteration complexity of variational quantum algorithms," (2023), arXiv:2209.10615 [quant-ph].

[196] Daniel Mastropietro, Georgios Korpas, Vyacheslav Kungurtsev, and Jakub Marecek, "Fleming-Viot helps speed up variational quantum algorithms in the presence of barren plateaus," (2023), arXiv:2311.18090 [quant-ph].

[197] Daniel J Egger, Jakub Mareček, and Stefan Woerner, "Warm-starting quantum optimization," Quantum **5**, 479 (2021).

[198] R. Tate, M. Farhadi, C. Herold, G. Mohler, and S. Gupta, "Bridging classical and quantum with SDP initialized warm-starts for QAOA," ACM Transactions on Quantum Computing **4**, 1–39 (2023).

[199] Yves Crama, Sourour Elloumi, Amélie Lambert, and Elisabeth Rodriguez-Heck, "Quadratization and convexification in polynomial binary optimization," (2022), working paper or preprint.

[200] Christa Zoufal, Ryan V. Mishmash, Nitin Sharma, Niraj Kumar, Aashish Sheshadri, Amol Deshmukh, Noelle Ibrahim, Julien Gacon, and Stefan Woerner, "Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection," Quantum **7**, 909 (2023).

[201] Andrew Lucas, "Ising formulations of many NP problems," Frontiers in Physics **2** (2014).

[202] Stuart Hadfield, "On the representation of Boolean and real functions as Hamiltonians for quantum computing," ACM Transactions on Quantum Computing **2**, 1–21 (2021).

[203] F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics I: a tutorial on formulating and using QUBO models." Ann Oper Res **314**, 141–183

[204] Pedro Maciel Xavier, Pedro Ripper, Tiago Andrade, Joaquim Dias Garcia, Nelson Maculan, and David E Bernal Neira, "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization," arXiv preprint arXiv:2307.02577 (2023).

[205] Sahar Karimi and Pooya Ronagh, "Practical integer-to-binary mapping for quantum annealers," Quantum Information Processing **18** (2019), 10.1007/s11128-019-2213-x.

[206] Nicolas PD Sawaya, Albert T Schmitz, and Stuart Hadfield, "Encoding trade-offs and design toolkits in quantum algorithms for discrete optimization: coloring, routing, scheduling, and other problems," Quantum **7**, 1111 (2023).

[207] JD Biamonte, "Nonperturbative k-body to two-body commuting conversion hamiltonians and embedding problem instances into ising spins," Physical Review A **77**, 052331 (2008).

[208] Ryan Babbush, Bryan O'Gorman, and Alán Aspuru-Guzik, "Resource efficient gadgets for compiling adiabatic quantum optimization problems," Annalen der Physik **525**, 877–888 (2013).

[209] Francisco Barahona, "On the computational complexity of ising spin glass models," Journal of Physics A: Mathematical and General **15**, 3241 (1982).

[210] A.P. Punnen, *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications* (Springer International Publishing, 2022).

[211] Sanjeev Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," Journal of the ACM **45**, 753–782 (1998).

[212] Seth Lloyd, "Quantum approximate optimization is computationally universal," (2018), arXiv:1812.11075 [quant-ph].

[213] Jonathan Wurtz and Peter Love, "Maxcut quantum approximate optimization algorithm performance guarantees for," Phys. Rev. A **103**, 042612 (2021).

[214] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou, "The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size," Quantum **6**, 759 (2022).

[215] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou, "The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model," in *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 232, edited by François Le Gall and Tomoyuki Morimae (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022) pp. 7:1–7:21.

[216] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," Phys. Rev. X **10**, 021067 (2020).

[217] Stefan H Sack and Maksym Serbyn, "Quantum annealing initialization of the quantum approximate optimization algorithm," Quantum **5**, 491 (2021).

[218] Giuseppe Scriva, Nikita Astrakhantsev, Sebastiano Pilati, and Guglielmo Mazzola, "Challenges of variational

quantum optimization with measurement shot noise," (2023), arXiv:2308.00044 [quant-ph].

[219] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh, "Accelerating quantum approximate optimization algorithm using machine learning," (2020), arXiv:2002.01089 [cs.ET].

[220] Michael Streif and Martin Leib, "Training the quantum approximate optimization algorithm without access to a quantum processing unit," (2019), arXiv:1908.08862 [quant-ph].

[221] Tomislav Begušić, Kasra Hejazi, and Garnet Kin-Lic Chan, "Simulating quantum circuit expectation values by clifford perturbation theory," (2023), arXiv:2306.04797 [quant-ph].

[222] Fernando G. S. L. Brandao, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," (2018), arXiv:1812.04170 [quant-ph].

[223] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, "Parameter concentrations in quantum approximate optimization," Physical Review A 104 (2021), 10.1103/physreva.104.l010401.

[224] Alexey Galda, Xiaoyuan Liu, Danylo Lykov, Yuri Alexeev, and Ilya Safro, "Transferability of optimal qaoa parameters between random graphs," (2021), arXiv:2106.07531 [quant-ph].

[225] Sami Boulebnane and Ashley Montanaro, "Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit," (2021), arXiv:2110.10685 [quant-ph].

[226] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. Campora III au2, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, A. Chernoguzov, E. Chertkov, J. Colina, J. P. Curtis, R. Daniel, M. DeCross, D. Deen, C. Delaney, J. M. Dreiling, C. T. Ertsgaard, J. Esposito, B. Estey, M. Fabrikant, C. Figgatt, C. Foltz, M. Foss-Feig, D. Francois, J. P. Gaebler, T. M. Gatterman, C. N. Gilbreth, J. Giles, E. Glynn, A. Hall, A. M. Hankin, A. Hansen, D. Hayes, B. Higashi, I. M. Hoffman, B. Horning, J. J. Hout, R. Jacobs, J. Johansen, L. Jones, J. Karcz, T. Klein, P. Lauria, P. Lee, D. Liefer, C. Lytle, S. T. Lu, D. Lucchetti, A. Malm, M. Matheny, B. Mathewson, K. Mayer, D. B. Miller, M. Mills, B. Neyenhuis, L. Nugent, S. Olson, J. Parks, G. N. Price, Z. Price, M. Pugh, A. Ransford, A. P. Reed, C. Roman, M. Rowe, C. Ryan-Anderson, S. Sanders, J. Sedlacek, P. Shevchuk, P. Siegfried, T. Skripka, B. Spaun, R. T. Sprenkle, R. P. Stutz, M. Swallows, R. I. Tobey, A. Tran, T. Tran, E. Vogt, C. Volin, J. Walker, A. M. Zolot, and J. M. Pino, "A race track trapped-ion quantum processor," (2023), arXiv:2305.03828 [quant-ph].

[227] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang, "Obstacles to variational quantum optimization from symmetry protection," Phys. Rev. Lett. 125, 260505 (2020).

[228] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang, "Hybrid quantum-classical algorithms for approximate graph coloring," Quantum 6, 678 (2022).

[229] Maxime Dupont, Bram Evert, Mark J. Hodson, Bhuvanesh Sundar, Stephen Jeffrey, Yuki Yamaguchi, Dennis Feng, Filip B. Maciejewski, Stuart Hadfield, M. Sohaib Alam, Zhihui Wang, Shon Grabbe, P. Aaron Lott, Eleanor G. Rieffel, Davide Venturelli, and Matthew J. Reagor, "Quantum-enhanced greedy combinatorial optimization solver," Science Advances 9 (2023), 10.1126/sciadv.adi0487.

[230] Lucas T. Brady and Stuart Hadfield, "Iterative quantum algorithms for maximum independent set: A tale of low-depth quantum algorithms," (2023), arXiv:2309.13110 [quant-ph].

[231] Jernej Rudi Finžgar, Aron Kerschbaumer, Martin J. A. Schuetz, Christian B. Mendl, and Helmut G. Katzgraber, "Quantum-informed recursive optimization algorithms," (2023), arXiv:2308.13607 [quant-ph].

[232] Yash J Patel, Sofiene Jerbi, Thomas Bäck, and Vedran Dunjko, "Reinforcement learning assisted recursive qaoa," EPJ Quantum Technology 11, 6 (2024).

[233] Edward Farhi, David Gamarnik, and Sam Gutmann, "The quantum approximate optimization algorithm needs to see the whole graph: A typical case," (2020), arXiv:2004.09002 [quant-ph].

[234] Kunal Marwaha and Stuart Hadfield, "Bounds on approximating max k xor with quantum and classical local algorithms," Quantum 6, 757 (2022).

[235] Aditi Misra-Spieldenner, Tim Bode, Peter K. Schuhmacher, Tobias Stollenwerk, Dmitry Bagrets, and Frank K. Wilhelm, "Mean-field approximate optimization algorithm," PRX Quantum 4 (2023), 10.1103/prxquantum.4.030335.

[236] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer, "A review on quantum approximate optimization algorithm and its variants," (2023), arXiv:2306.09198 [quant-ph].

[237] Bryce Fuller, Charles Hadfield, Jennifer R. Glick, Takashi Imamichi, Toshinari Itoko, Richard J. Thompson, Yang Jiao, Marna M. Kagele, Adriana W. Blom-Schieber, Rudy Raymond, and Antonio Mezzacapo, "Approximate Solutions of Combinatorial Problems via Quantum Relaxations," (2021), arXiv:2111.03167 [quant-ph].

[238] Kosei Teramoto, Rudy Raymond, Eyuri Wakakuwa, and Hiroshi Imai, "Quantum-relaxation based optimization algorithms: Theoretical extensions," (2023), arXiv:2302.09481 [quant-ph].

[239] Taylor L. Patti, Jean Kossaifi, Anima Anandkumar, and Susanne F. Yelin, "Variational quantum optimization with multi-basis encodings," (2022), arXiv:2106.13304 [quant-ph].

[240] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan, "Variational ansatz-based quantum simulation of imaginary time evolution," npj Quantum Information 5, 75 (2019).

[241] Christa Zoufal, "Generative quantum machine learning," (2021), arXiv:2111.12738 [quant-ph].

[242] Sheng-Hsuan Lin, Rohit Dilip, Andrew G. Green, Adam Smith, and Frank Pollmann, "Real- and Imaginary-Time Evolution with Compressed Quantum Circuits," PRX Quantum 2, 010342 (2021).

[243] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin, "Theory of variational quantum simulation," Quantum 3, 191 (2019).

[244] Marcello Benedetti, Mattia Fiorentini, and Michael

Lubasch, "Hardware-efficient variational quantum algorithms for time evolution," Physical Review Research **3**, 033083 (2021).

[245] Lucas Slattery, Benjamin Villalonga, and Bryan K. Clark, "Unitary block optimization for variational quantum algorithms," Physical Review Research **4**, 023072 (2022).

[246] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo, "Quantum Natural Gradient," Quantum **4**, 269 (2020).

[247] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner, "Stochastic approximation of variational quantum imaginary time evolution," (2023), arXiv:2305.07059 [quant-ph].

[248] Stefano Barison, Filippo Vicentini, and Giuseppe Carleo, "An efficient quantum algorithm for the time evolution of parameterized circuits," Quantum **5**, 512 (2021).

[249] Pauline J. Ollitrault, Alexander Miessen, and Ivano Tavernelli, "Molecular quantum dynamics: A quantum computing perspective," Accounts of Chemical Research **54**, 4229–4238 (2021).

[250] Christa Zoufal, David Sutter, and Stefan Woerner, "Error bounds for variational quantum time evolution," Phys. Rev. Appl. **20**, 044059 (2023).

[251] Panagiotis Kl. Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner, "Improving Variational Quantum Optimization using CVaR," Quantum **4**, 256 (2020).

[252] Samantha V. Barron, Daniel J. Egger, Elijah Pelofske, Andreas Bärtschi, Stephan Eidenbenz, Matthis Lehmkuehler, and Stefan Woerner, "Provable bounds for noise-free expectation values computed from noisy samples," (2023), arXiv:2312.00733 [quant-ph].

[253] Anthony Angone, Xioayuan Liu, Ruslan Shaydulin, and Ilya Safro, "Hybrid quantum-classical multilevel approach for maximum cuts on graphs," (2023), arXiv:2309.08815 [quant-ph].

[254] Ashley Montanaro, "Quantum speedup of branch-and-bound algorithms," Physical Review Research **2**, 013056 (2020).

[255] Andreas Bärtschi and Stephan Eidenbenz, "Short-Depth Circuits for Dicke State Preparation," in IEEE International Conference on Quantum Computing and Engineering QCE'22 (2022) pp. 87–96.

[256] Google AI Quantum, Collaborators, et al., "Hartree-fock on a superconducting qubit quantum computer," Science **369**, 1084–1089 (2020).

[257] Stuart Hadfield, Zhihui Wang, Eleanor G Rieffel, Bryan O'Gorman, Davide Venturelli, and Rupak Biswas, "Quantum approximate optimization with hard and soft constraints," in Proceedings of the Second International Workshop on Post Moores Era Supercomputing (2017) pp. 15–21.

[258] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," Algorithms **12**, 34 (2019).

[259] Zhihui Wang, Nicholas C. Rubin, Jason M. Dominy, and Eleanor G. Rieffel, "xy mixers: Analytical and numerical results for the quantum alternating operator ansatz," Physical Review A **101**, 012320 (2020).

[260] Jeremy Cook, Stephan Eidenbenz, and Andreas Bärtschi, "The Quantum Alternating Operator Ansatz on Maximum k-Vertex Cover," in IEEE International Conference on Quantum Computing and Engineering QCE'20 (2020) pp. 83–92.

[261] Andreas Bärtschi and Stephan Eidenbenz, "Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation," in IEEE International Conference on Quantum Computing and Engineering QCE'20 (2020) pp. 72–82.

[262] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al., "Quantum optimization using variational algorithms on near-term quantum devices," Quantum Science and Technology **3**, 030503 (2018).

[263] Ryan LaRose, Eleanor Rieffel, and Davide Venturelli, "Mixer-phaser ansätze for quantum optimization with hard constraints," Quantum Machine Intelligence **4**, 17 (2022).

[264] Filip B Maciejewski, Stuart Hadfield, Benjamin Hall, Mark Hodson, Maxime Dupont, Bram Evert, James Sud, M Sohaib Alam, Zhihui Wang, Stephen Jeffrey, et al., "Design and execution of quantum circuits using tens of superconducting qubits and thousands of gates for dense ising optimization problems," arXiv preprint arXiv:2308.12423 (2023), arXiv:2308.12423 [quant-ph].

[265] Dylan Herman, Ruslan Shaydulin, Yue Sun, Shouvanik Chakrabarti, Shaohan Hu, Pierre Minssen, Arthur Rattew, Romina Yalovetzky, and Marco Pistoia, "Constrained optimization via quantum zeno dynamics," Communications Physics **6**, 219 (2023).

[266] Fred Glover, Gary Kochenberger, Moses Ma, and Yu Du, "Quantum bridge analytics ii: Qubo-plus, network optimization and combinatorial chaining for asset exchange," Annals of Operations Research **314**, 185–212 (2022).

[267] Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif, "No quantum speedup over gradient descent for non-smooth convex optimization," in 12th Innovations in Theoretical Computer Science Conference (ITCS 2021) (Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021).

[268] Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif, "Near-optimal lower bounds for convex optimization for all orders of smoothness," in Advances in Neural Information Processing Systems, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Curran Associates, Inc., 2021) pp. 29874–29884.

[269] Martin Grötschel, László Lovász, and Alexander Schrijver, Geometric algorithms and combinatorial optimization, Vol. 2 (Springer Science & Business Media, 2012).

[270] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, "Convex optimization using quantum oracles," Quantum **4** (2020), 10.22331/q-2020-01-13-220.

[271] Shouvanik Chakrabarti, Andrew M Childs, Tongyang Li, and Xiaodi Wu, "Quantum algorithms and lower bounds for convex optimization," Quantum **4**, 221 (2020).

[272] Sanjeev Arora, Elad Hazan, and Satyen Kale, "Fast algorithms for approximate semidefinite programming using the multiplicative weights update method," in 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05) (IEEE, 2005) pp. 339–348.

[273] Joran van Apeldoorn, *A quantum view on convex optimization*, Ph.D. thesis, University of Amsterdam Institute for Logic, Language and Computation (ILLC) (2020).

[274] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous, "Qip = pspace," J. ACM **58** (2011), 10.1145/2049697.2049704.

[275] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone, "Quantum random access memory," Phys. Rev. Lett. **100**, 160501 (2008), arxiv:0708.1879 [quant-ph].

[276] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone, "Architectures for a quantum random access memory," Phys. Rev. A **78**, 052310 (2008).

[277] Scott Aaronson, "Read the fine print," Nature Phys **11**, 291–293 (2015).

[278] Olivia Di Matteo, Vlad Gheorghiu, and Michele Mosca, "Fault tolerant resource estimation of quantum random-access memories," IEEE Trans. Quantum Eng. **1**, 1–13 (2020), arxiv:1902.01329 [quant-ph].

[279] B. David Clader, Alexander M. Dalzell, Nikitas Stamatopoulos, Grant Salton, Mario Berta, and William J. Zeng, "Quantum Resources Required to Block-Encode a Matrix of Classical Data," (2022), arxiv:2206.03505 [quant-ph].

[280] Samuel Jaques and Arthur G. Rattew, "QRAM: A Survey and Critique," (2023), arxiv:2305.10310 [quant-ph].

[281] Michael D. Grigoriadis and Leonid G. Khachiyan, "Approximate solution of matrix games in parallel," in *Advances in Optimization and Parallel Computing*, edited by Panos M. Pardalos (Elsevier, 1991).

[282] Michael D. Grigoriadis and Leonid G. Khachiyan, "A sublinear-time randomized approximation algorithm for matrix games," Operations Research Letters **18**, 53–58 (1995).

[283] Jakub Marecek and Albert Akhriev, "A cutting-plane method for semidefinite programming with potential applications on noisy quantum devices," (2021), arXiv:2110.03400 [quant-ph].

[284] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery, "The power of block-encoded matrix powers: Improved regression techniques via faster hamiltonian simulation," Leibniz international proceedings in informatics **132** (2019).

[285] Patrick Rebentrost, Maria Schuld, Leonard Wossnig, Francesco Petruccione, and Seth Lloyd, "Quantum gradient descent and newton's method for constrained polynomial optimization," New Journal of Physics **21**, 073023 (2019).

[286] Iordanis Kerenidis and Anupam Prakash, "A quantum interior point method for lps and sdps," ACM Transactions on Quantum Computing **1** (2020), 10.1145/3406306.

[287] Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis F Zuluaga, "Quantum interior point methods for semidefinite optimization," Quantum **7**, 1110 (2023).

[288] Mohammadhossein Mohammadisiahroudi, Ramin Fakhimi, and Tamás Terlaky, "Efficient use of quantum linear system algorithms in interior point methods for linear optimization," (2023), arXiv:2205.01220 [quant-ph].

[289] Zeguan Wu, Mohammadhossein Mohammadisiahroudi, Brandon Augustino, Xiu Yang, and Tamás Terlaky, "An inexact feasible quantum interior point method for linearly constrained quadratic optimization," Entropy **25**, 330 (2023).

[290] Joran van Apeldoorn, Arjan Cornelissen, András Gilyén, and Giacomo Nannicini, "Quantum tomography using state-preparation unitaries," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, 2023) pp. 1265–1318.

[291] Simon Apers and Sander Gribling, "Quantum speedups for linear programming via interior point methods," (2023), arXiv:2311.03215 [quant-ph].

[292] Jan Van Den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang, "Minimum cost flows, mdps, and l1-regression in nearly linear time for dense instances," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (2021) pp. 859–869.

[293] Simon Apers and Ronald de Wolf, "Quantum speedup for graph sparsification, cut approximation, and Laplacian solving," SIAM Journal on Computing **51**, 1703–1742 (2022).

[294] Sander Gribling and Harold Nieuwboer, "Improved quantum lower and upper bounds for matrix scaling," in *Proceedings of 39th International Symposium on Theoretical Aspects of Computer Science (STACS'22)*, Vol. 219 (2022) pp. 35:1–35:23, arXiv:2109.15282v1.

[295] Nathan Linial, Alex Samorodnitsky, and Avi Wigderson, "A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents," Combinatorica **20**, 545–568 (2000).

[296] Edward Anderson, Zhaojun Bai, Christian Bischof, Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, *et al.*, *LAPACK Users' guide* (SIAM, 1999).

[297] Joran van Apeldoorn, Sander Gribling, Yinan Li, Harold Nieuwboer, Michael Walter, and Ronald de Wolf, "Quantum algorithms for matrix scaling and matrix balancing," in *Proceedings of 48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*, Vol. 198 (2021) pp. 110:1–110:17, arXiv:2011.12823v1.

[298] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu, "Quantum hamiltonian descent," arXiv preprint arXiv:2303.01471 (2023).

[299] Andre Wibisono, Ashia C Wilson, and Michael I Jordan, "A variational perspective on accelerated methods in optimization," proceedings of the National Academy of Sciences **113**, E7351–E7358 (2016).

[300] Brandon Augustino, Jiaqi Leng, Giacomo Nannicini, Tamás Terlaky, and Xiaodi Wu, "A quantum central path algorithm for linear optimization," arXiv preprint arXiv:2311.03977 (2023).

[301] Leo Liberti, "Undecidability and hardness in mixed-integer nonlinear programming," RAIRO-Operations Research **53**, 81–109 (2019).

[302] Frank H. Clarke, *Optimization and Nonsmooth Analysis* (Society for Industrial and Applied Mathematics, 1990) https://epubs.siam.org/doi/pdf/10.1137/1.9781611971309.

[303] Jiajin Li, Anthony Man-Cho So, and Wing-Kin Ma, "Understanding notions of stationarity in nonsmooth optimization: A guided tour of various constructions of subdifferential for nonsmooth functions," IEEE Signal Processing Magazine **37**, 18–31 (2020).

[304] Chenyi Zhang and Tongyang Li, "Quantum lower bounds for finding stationary points of nonconvex func-

tions," in *International Conference on Machine Learning* (PMLR, 2023) pp. 41268–41299.

[305] Aaron Sidford and Chenyi Zhang, "Quantum speedups for stochastic optimization," (2023), arXiv:2308.01582 [quant-ph].

[306] Iordanis Kerenidis and Anupam Prakash, "Quantum gradient descent for linear systems and least squares," Phys. Rev. A **101**, 022316 (2020).

[307] Keren Li, Shijie Wei, Pan Gao, Feihao Zhang, Zengrong Zhou, Tao Xin, Xiaoting Wang, Patrick Rebentrost, and Guilu Long, "Optimizing a polynomial function on a quantum processor," npj Quantum Information **7**, 16 (2021).

[308] Rolando D Somma, Sergio Boixo, Howard Barnum, and Emanuel Knill, "Quantum simulations of classical annealing processes," Physical review letters **101**, 130504 (2008).

[309] Satoshi Morita and Hidetoshi Nishimori, "Mathematical foundation of quantum annealing," Journal of Mathematical Physics **49** (2008).

[310] Yizhou Liu, Weijie J Su, and Tongyang Li, "On quantum speedups for nonconvex optimization via quantum tunneling walks," Quantum **7**, 1030 (2023).

[311] Immanuel M. Bomze, Jianqiang Cheng, Peter J. C. Dickinson, Abdel Lisser, and Jia Liu, "Notoriously hard (mixed-)binary qps: empirical evidence on new completely positive approaches," Comput. Manag. Sci. **16**, 593–619 (2019).

[312] Wim Van Ackooij, Irene Danti Lopez, Antonio Frangioni, Fabrizio Lacalandra, and Milad Tahanan, "Large-scale unit commitment under uncertainty: an updated literature survey," Annals of Operations Research **271**, 11–85 (2018).

[313] Ashley Montanaro, "Quantum walk speedup of backtracking algorithms," arXiv preprint arXiv:1509.02374 (2015).

[314] Andris Ambainis and Martins Kokainis, "Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017) pp. 989–1002.

[315] Shouvanik Chakrabarti, Pierre Minssen, Romina Yalovetzky, and Marco Pistoia, "Universal quantum speedup for branch-and-bound, branch-and-cut, and tree-search algorithms," arXiv preprint arXiv:2210.03210 (2022).

[316] Alexander M Dalzell, Nicola Pancotti, Earl T Campbell, and Fernando GSL Brandão, "Mind the gap: Achieving a super-grover quantum speedup by jumping to the end," in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (2023) pp. 1131–1144.

[317] Claudio Gambella, Andrea Simonetto, Martin Mevissen, and Jakub Marecek, "Mixed-binary constrained optimization on quantum computers," (2020), uS Patent App. 16/395,560.

[318] Claudio Gambella and Andrea Simonetto, "Multiblock admm heuristics for mixed-binary optimization on classical and quantum computers," IEEE Transactions on Quantum Engineering **1**, 1–22 (2020).

[319] Chin-Yao Chang, Eric Jones, Yiyun Yao, Peter Graf, and Rishabh Jain, "On hybrid quantum and classical computing algorithms for mixed-integer programming," arXiv preprint arXiv:2010.07852 (2020).

[320] Lee Braine, Daniel J. Egger, Jennifer Glick, and Ste-

fan Woerner, "Quantum algorithms for mixed binary optimization applied to transaction settlement," IEEE Transactions on Quantum Engineering **2**, 1–8 (2021).

[321] Thinh Viet Le and Vassilis Kekatos, "Variational Quantum Eigensolver with Constraints (VQEC): Solving Constrained Optimization Problems via VQE," (2023), arXiv:2311.08502 [quant-ph].

[322] Dimitri P Bertsekas, *Constrained optimization and Lagrange multiplier methods* (Academic press, 2014).

[323] Ernesto G Birgin and José Mario Martínez, *Practical augmented Lagrangian methods for constrained optimization* (SIAM, 2014).

[324] Johannes Aspman, Gilles Bareilles, Vyacheslav Kungurtsev, Jakub Marecek, and Martin Takáč, "Hybrid methods in polynomial optimisation," (2023), arXiv:2305.16122 [math.OC].

[325] Michal Kočvara and Michael Stingl, "Pennon: A code for convex nonlinear and semidefinite programming," Optimization methods and software **18**, 317–333 (2003).

[326] George B Dantzig and Philip Wolfe, "Decomposition principle for linear programs," Operations research **8**, 101–111 (1960).

[327] Pablo A Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization* (California Institute of Technology, 2000).

[328] Jean B. Lasserre, "Global optimization with polynomials and the problem of moments," SIAM Journal on optimization **11**, 796–817 (2001).

[329] Grant Schoenebeck, "Linear level lasserre lower bounds for certain k-csps," in *2008 49th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2008) pp. 593–602.

[330] Samuel Burer, "On the copositive representation of binary and continuous nonconvex quadratic programs," Mathematical Programming **120**, 479–495 (2009).

[331] Peter JC Dickinson, "Geometry of the copositive and completely positive cones," Journal of Mathematical Analysis and Applications **380**, 377–395 (2011).

[332] Peter JC Dickinson and Luuk Gijben, "On the computational complexity of membership problems for the completely positive cone and its dual," Computational optimization and applications **57**, 403–415 (2014).

[333] Immanuel M Bomze, "Copositive optimization–recent developments and applications," European Journal of Operational Research **216**, 509–520 (2012).

[334] Robin Brown, David E. Bernal Neira, Davide Venturelli, and Marco Pavone, "A copositive framework for analysis of hybrid ising-classical algorithms," (2023), arXiv:2207.13630 [math.OC].

[335] Cheng Guo, Merve Bodur, and Joshua A. Taylor, "Copositive duality for discrete markets and games," (2021), arXiv:2101.05379 [math.OC].

[336] Aleksandrs Belovs, "Quantum algorithms for learning symmetric juntas via the adversary bound," computational complexity **24**, 255–293 (2015).

[337] Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Junichi Teruyama, "Quantum counterfeit coin problems," Theoretical Computer Science **456**, 51–64 (2012).

[338] Andris Ambainis, Kaspars Balodis, Janis Iraids, Martins Kokainis, Krisjanis Prusis, and Jevgenijs Vihrovs, "Quantum speedups for exponential-time dynamic programming algorithms," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, Jan-*

uary 6-9, 2019, edited by Timothy M. Chan (SIAM, 2019) pp. 1783–1793.

[339] Serge Gaspers and Jerry Zirui Li, "Quantum algorithms for graph coloring and other partitioning, covering, and packing problems," (2023), arXiv:2311.08042 [cs.DS].

[340] Masayuki Miyamoto, Masakazu Iwamura, Koichi Kise, and François Le Gall, "Quantum speedup for the minimum steiner tree problem," in *International Computing and Combinatorics Conference* (Springer, 2020) pp. 234–245.

[341] Vladislavs Klevickis, Krisjanis Prusis, and Jevgenijs Vihrovs, "Quantum speedups for treewidth," in *17th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2022, July 11-15, 2022, Urbana Champaign, Illinois, USA*, LIPIcs, Vol. 232, edited by François Le Gall and Tomoyuki Morimae (Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022) pp. 11:1–11:18.

[342] Kazuya Shimizu and Ryuhei Mori, "Exponential-time quantum algorithms for graph coloring problems," in *LATIN 2020: Theoretical Informatics - 14th Latin American Symposium, São Paulo, Brazil, January 5-8, 2021, Proceedings*, Lecture Notes in Computer Science, Vol. 12118, edited by Yoshiharu Kohayakawa and Flávio Keidi Miyazawa (Springer, 2020) pp. 387–398.

[343] Robert A Wagner and Michael J Fischer, "The string-to-string correction problem," Journal of the ACM (JACM) **21**, 168–173 (1974).

[344] Andris Ambainis, Kaspars Balodis, Janis Iraids, Kamil Khadiev, Vladislavs Klevickis, Krisjanis Prusis, Yixin Shen, Juris Smotrovs, and Jevgenijs Vihrovs, "Quantum bounds for 2d-grid and dyck language," Quantum Inf. Process. **22**, 194 (2023).

[345] Martino Bardi, Italo Capuzzo Dolcetta, *et al.*, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Vol. 12 (Springer, 1997).

[346] John S Baras and A Kurzhanski, "Nonlinear filtering: The set-membership (bounding) and the h∞ techniques," IFAC Proceedings Volumes **28**, 409–415 (1995).

[347] Martin L Puterman, *Markov decision processes: discrete stochastic dynamic programming* (John Wiley & Sons, 2014).

[348] Giuseppe C. Calafiore and Fabrizio Dabbene, "A probabilistic analytic center cutting plane method for feasibility of uncertain lmis," Automatica **43**, 2022–2033 (2007).

[349] Jonathan P Epperlein, Roman Overko, Sergiy Zhuk, Christopher King, Djallel Bouneffouf, Andrew Cullen, and Robert Shorten, "Reinforcement learning with algorithms from probabilistic structure estimation," Automatica **144**, 110483 (2022).

[350] Daochen Wang, Xuchen You, Tongyang Li, and Andrew M. Childs, "Quantum exploration algorithms for multi-armed bandits," Proceedings of the AAAI Conference on Artificial Intelligence **35**, 10102–10110 (2021).

[351] Daochen Wang, Aarthi Sundaram, Robin Kothari, Ashish Kapoor, and Martin Roetteler, "Quantum algorithms for reinforcement learning with a generative model," in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 139, edited by Marina Meila and Tong Zhang (PMLR, 2021) pp. 10916–10926.

[352] Arjan Cornelissen, "Quantum gradient estimation and its application to quantum reinforcement learning," TU

Delft Library (2018).

[353] Vedran Dunjko, Jacob M. Taylor, and Hans J. Briegel, "Quantum-enhanced machine learning," Phys. Rev. Lett. **117**, 130501 (2016).

[354] Sofiene Jerbi, Arjan Cornelissen, Maris Ozols, and Vedran Dunjko, "Quantum Policy Gradient Algorithms," in *18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 266, edited by Omar Fawzi and Michael Walter (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2023) pp. 13:1–13:24.

[355] Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi, "Near-optimal quantum algorithms for multivariate mean estimation," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022 (Association for Computing Machinery, New York, NY, USA, 2022) pp. 33–43.

[356] Vedran Dunjko, Yi-Kai Liu, Xingyao Wu, and Jacob M. Taylor, "Exponential improvements for quantum-accessible reinforcement learning," (2017), arXiv:1710.11160.

[357] Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko, "Parametrized quantum policies for reinforcement learning," in *Advances in Neural Information Processing Systems*, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Curran Associates, Inc., 2021) pp. 28362–28375.

[358] Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko, "Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning," Quantum **6**, 720 (2022).

[359] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature **529**, 484–489 (2016).

[360] Kai Arulkumaran, Antoine Cully, and Julian Togelius, "Alphastar: an evolutionary computation perspective," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19 (ACM, 2019).

[361] Owen Lockwood and Mei Si, "Reinforcement learning with quantum variational circuits," (2020), arXiv:2008.07524.

[362] Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan, "Variational quantum circuits for deep reinforcement learning," IEEE Access **8**, 141007–141024 (2020).

[363] Alexey Muravlev and Mikhail Zhitlukhin, "A Bayesian sequential test for the drift of a fractional Brownian motion," Advances in Applied Probability **52**, 1308–1324 (2020).

[364] Constantinos Daskalakis and Yasushi Kawase, "Optimal stopping rules for sequential hypothesis testing," in *25th Annual European Symposium on Algorithms (ESA)* (2017).

[365] Francis A. Longstaff and Eduardo S. Schwartz, "Valuing american options by simulation: a simple least-squares approach," The review of financial studies **14**, 113–147

(2001).

[366] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp, "Quantum amplitude amplification and estimation," Contemporary Mathematics **305**, 53–74 (2002).

[367] J. F. Doriguello, Alessandro Luongo, Jinge Bao, Patrick Rebentrost, and Miklos Santha, "Quantum Algorithm for Stochastic Optimal Stopping Problems with Applications in Finance," in *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 232, edited by François Le Gall and Tomoyuki Morimae (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022) pp. 2:1–2:24.

[368] Yangchen Pan, Erfan Sadeqi Azer, and Martha White, "Effective sketching methods for value function approximation," (2017), arXiv:1708.01298 [cs.LG].

[369] Haifang Li, Yingce Xia, and Wensheng Zhang, "Finite sample analysis of lstd with random projections and eligibility traces," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018) pp. 2390–2396.

[370] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf, "The inductive bias of quantum kernels," Advances in Neural Information Processing Systems **34**, 12661–12673 (2021).

[371] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (JMLR Workshop and Conference Proceedings, 2011) pp. 208–214.

[372] Aharon Ben-Tal and Arkadi Nemirovski, "Robust convex optimization," Mathematics of operations research **23**, 769–805 (1998).

[373] Aharon Ben-Tal and Arkadi Nemirovski, "Robust optimization–methodology and applications," Mathematical programming **92**, 453–480 (2002).

[374] Aharon Ben-Tal and Arkadi Nemirovski, "Selected topics in robust convex optimization," Mathematical Programming **112**, 125–158 (2008).

[375] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis, "Theory and applications of robust optimization," SIAM review **53**, 464–501 (2011).

[376] Aharon Ben-Tal, Elad Hazan, Tomer Koren, and Shie Mannor, "Oracle-based robust optimization via online learning," Operations Research **63**, 628–638 (2015).

[377] Debbie Lim, J. F. Doriguello, and Patrick Rebentrost, "Quantum algorithm for robust optimization via stochastic-gradient online learning," (2023), arXiv:2304.02262 [quant-ph].

[378] Nyoman Gunantara, "A review of multi-objective optimization: Methods and its applications," Cogent Engineering **5**, 1502242 (2018).

[379] Christos H Papadimitriou and Mihalis Yannakakis, "On the approximability of trade-offs and optimal access of web sources," in *Proceedings 41st annual symposium on foundations of computer science* (IEEE, 2000) pp. 86–92.

[380] Ilias Diakonikolas and Mihalis Yannakakis, "Small approximate pareto sets for biobjective shortest paths and other problems," SIAM Journal on Computing **39**, 1340–1371 (2010).

[381] Nery Riquelme, Christian Von Lücken, and Benjamin Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)* (2015) pp. 1–11.

[382] Shao-Hen Chiew, Kilian Poirier, Rajesh Mishra, Ulrike Bornheimer, Ewan Munro, Si Han Foon, Christopher Wanru Chen, Wei Sheng Lim, and Chee Wei Nga, "Multi-objective optimization and network routing with near-term quantum computers," (2023), arXiv:2308.08245 [quant-ph].

[383] Linus Ekstrom, Hao Wang, and Sebastian Schmitt, "Variational quantum multi-objective optimization," (2024), arXiv:2312.14151 [quant-ph].

[384] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Yu Chen, and et al., "Digitized adiabatic quantum computing with a superconducting circuit," Nature **534**, 222–226 (2016).

[385] Clemens Dlaska, Kilian Ender, Glen Bigan Mbeng, Andreas Kruckenhauser, Wolfgang Lechner, and Rick van Bijnen, "Quantum optimization via four-body rydberg gates," Phys. Rev. Lett. **128**, 120503 (2022).

[386] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, and *et al.*, "Quantum approximate optimization of nonplanar graph problems on a planar superconducting processor," Nat. Phys. **17**, 332–336 (2021).

[387] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, "Quantum optimization of maximum independent set using Rydberg atom arrays," Science **376**, 1209–1215 (2022).

[388] Guido Pagano, Aniruddha Bapat, Patrick Becker, Katherine S. Collins, Arinjoy De, Paul W. Hess, Harvey B. Kaplan, Antonis Kyprianidis, Wen Lin Tan, Christopher Baldwin, Lucas T. Brady, Abhinav Deshpande, Fangli Liu, Stephen Jordan, Alexey V. Gorshkov, and Christopher Monroe, "Quantum approximate optimization of the long-range ising model with a trapped-ion quantum simulator," Proceedings of the National Academy of Sciences **117**, 25396–25401 (2020).

[389] Phillip C. Lotshaw, Kevin D. Battles, Bryan Gard, Gilles Buchs, Travis S. Humble, and Creston D. Herold, "Modeling noise in global mølmer-sørensen interactions applied to quantum approximate optimization," Physical Review A **107** (2023), 10.1103/physreva.107.062406.

[390] Elijah Pelofske, Andreas Bärtschi, John Golden, and Stephan Eidenbenz, "High-Round QAOA for MAX k-SAT on Trapped Ion NISQ Devices," in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (2023) pp. 506–517.

[391] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P. Orlando, Simon Gustavsson, and William D. Oliver, "A quantum engineer's guide to superconducting qubits," Appl. Phys. Rev. **6**, 021318 (2019).

[392] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and John M. Martinis, "Superconducting quantum circuits at the surface code threshold for fault tol-

erance," Nature **508**, 500–503 (2014).

[393] Chad Rigetti and Michel Devoret, "Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies," Phys. Rev. B **81**, 134507 (2010).

[394] David C. McKay, Stefan Filipp, Antonio Mezzacapo, Easwar Magesan, Jerry M. Chow, and Jay M. Gambetta, "Universal gate for fixed-frequency qubits via a tunable bus," Phys. Rev. Appl. **6**, 064007 (2016).

[395] M. Ganzhorn, G. Salis, D. J. Egger, A. Fuhrer, M. Mergenthaler, C. Müller, P. Müller, S. Paredes, M. Pechal, M. Werninghaus, and S. Filipp, "Benchmarking the noise sensitivity of different parametric two-qubit gates in a single superconducting quantum computing platform," Phys. Rev. Res. **2**, 033447 (2020).

[396] IBM, "IBM Quantum Platform - Compute resources," (2023).

[397] S. Hadfield, T. Hogg, and E. G. Rieffel, "Analytical framework for quantum alternating operator ansätze," Quantum Science and Technology **8**, 015017 (2022).

[398] P. Niroula, R. Shaydulin, R. Yalovetsky, P. Minssen, D. Herman, S. Hu, and M. Pistoia, "Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer," Scientific Reports **12** (2022), 10.1038/s41598-022-20853-w.

[399] P. Chandarana, N. N. Hegade, K. Paul, F. Albarrán-Arriagada, E. Solano, A. del Campo, and Xi Chen, "Digitized-counterdiabatic quantum approximate optimization algorithm," Phys. Rev. Res. **4**, 013141 (2022).

[400] J. S. Baker and S. K. Radha, "Wasserstein solution quality and the quantum approximate optimization algorithm: A portfolio optimization case study," arXiv:2202.06782 (2022).

[401] Yunlong Yu, Chenfeng Cao, Carter Dewey, Xiang-Bin Wang, Nic Shannon, and Robert Joynt, "Quantum approximate optimization algorithm with adaptive bias fields," Phys. Rev. Res. **4**, 023249 (2022).

[402] Johannes Weidenfeller, Lucia C Valor, Julien Gacon, Caroline Tornow, Luciano Bello, Stefan Woerner, and Daniel J Egger, "Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware," Quantum **6**, 870 (2022).

[403] Stefan H. Sack and Daniel J. Egger, "Large-scale quantum approximate optimization on non-planar graphs with machine learning noise mitigation," (2023), arXiv:2307.14427 [quant-ph].

[404] Aidan Pellow-Jarman, Shane McFarthing, Ilya Sinayskiy, Anban Pillay, and Francesco Petruccione, "QAOA Performance in Noisy Devices: The Effect of Classical Optimizers and Ansatz Depth," (2023), arXiv:2307.10149 [quant-ph].

[405] A. S. Holevo, "On capacity of a quantum communications channel," Problems of Information Transmission **15**, 247–253 (1979).

[406] A. Ambainis, A. Nayak, A. Tashma, and U. Vazirani, "Dense quantum coding and a lower bound for 1-way quantum automata," in *Proceedings of 31st ACM Symposium on Theory of Computing* (ACM, 1999) pp. 376–383.

[407] A. Ambainis, A. Nayak, A. Ta-shma, and U. Vazirani, "Dense quantum coding and quantum finite automata," Journal of the ACM **49**, 496–511 (2002).

[408] T. Imamichi and R. Raymond, "Constructions of quantum random access codes," in *Asian Quantum Informa-tion Symposium (AQIS)*, Vol. 66 (2018).

[409] Laurin E. Fischer, Daniel Miller, Francesco Tacchino, Panagiotis Kl. Barkoutsos, Daniel J. Egger, and Ivano Tavernelli, "Ancilla-free implementation of generalized measurements for qubits embedded in a qudit space," Phys. Rev. Res. **4**, 033027 (2022).

[410] Maxime Dupont and Bhuvanesh Sundar, "Extending relax-and-round combinatorial optimization solvers with quantum correlations," Physical Review A **109**, 012429 (2024).

[411] Andrew Cross, Ali Javadi-Abhari, Thomas Alexander, Niel De Beaudrap, Lev S Bishop, Steven Heidel, Colm A Ryan, Prasahnt Sivarajah, John Smolin, Jay M Gambetta, *et al.*, "Openqasm 3: A broader and deeper quantum assembly language," ACM Transactions on Quantum Computing **3**, 1–50 (2022).

[412] Robert S. Smith, Michael J. Curtis, and William J. Zeng, "A Practical Quantum Instruction Set Architecture," (2017), arXiv:1608.03355 [quant-ph].

[413] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook, "Strawberry fields: A software platform for photonic quantum computing," Quantum **3**, 129 (2019).

[414] Paul D. Nation and Matthew Treinish, "Suppressing quantum circuit errors due to system variability," PRX Quantum **4**, 010327 (2023).

[415] Fei Hua, Yuwei Jin, Ang Li, Yanhao Chen, Chi Zhang, Ari Hayes, Hang Gao, and Eddy Z. Zhang, "A synergistic compilation workflow for tackling crosstalk in quantum machines," (2022), arXiv:2207.05751 [quant-ph].

[416] David C. McKay, Ian Hincks, Emily J. Pritchett, Malcolm Carroll, Luke C. G. Govia, and Seth T. Merkel, "Benchmarking Quantum Processor Performance at Scale," (2023), arXiv:2311.05933 [quant-ph].

[417] Fei Hua, Yuwei Jin, Yanhao Chen, Suhas Vittal, Kevin Krsulich, Lev S. Bishop, John Lapeyre, Ali Javadi-Abhari, and Eddy Z. Zhang, "Exploiting qubit reuse through mid-circuit measurement and reset," (2023), arXiv:2211.01925 [quant-ph].

[418] A. Matsuo, S. Yamashita, and D. J. Egger, "A SAT approach to the initial mapping problem in SWAP gate insertion for commuting gates," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences , 2022EAP1159 (2023).

[419] Jun Zhang, Jiri Vala, Shankar Sastry, and K. Birgitta Whaley, "Minimum construction of two-qubit quantum operations," Phys. Rev. Lett. **93**, 020502 (2004).

[420] Byron Drury and Peter Love, "Constructive quantum Shannon decomposition from Cartan involutions," Journal of Physics A: Mathematical and Theoretical **41**, 395305 (2008).

[421] Dmitri Maslov, "Basic circuit compilation techniques for an ion-trap quantum machine," New Journal of Physics **19**, 023035 (2017).

[422] David P. DiVincenzo, "The physical implementation of quantum computation," Fortschritte der Physik **48**, 771–783 (2000).

[423] Gushu Li, Anbang Wu, Yunong Shi, Ali Javadi-Abhari, Yufei Ding, and Yuan Xie, "Paulihedral: A generalized block-wise compiler optimization framework for quantum simulation kernels," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASP-

LOS '22 (Association for Computing Machinery, New York, NY, USA, 2022) pp. 554–569.

[424] Liam Madden and Andrea Simonetto, "Best Approximate Quantum Compiling Problems," ACM Transactions on Quantum Computing **3** (2022), 10.1145/3505181, place: New York, NY, USA Publisher: Association for Computing Machinery.

[425] Ethan Smith, Marc Grau Davis, Jeffrey Larson, Ed Younis, Lindsay Bassman Oftelie, Wim Lavrijsen, and Costin Iancu, "LEAP: Scaling Numerical Optimization Based Synthesis Using an Incremental Approach," ACM Transactions on Quantum Computing **4** (2023), 10.1145/3548693, place: New York, NY, USA Publisher: Association for Computing Machinery.

[426] Gushu Li, Yufei Ding, and Yuan Xie, "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19 (Association for Computing Machinery, New York, NY, USA, 2019) pp. 1001–1014.

[427] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the ibm qx architectures," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **38**, 1226–1236 (2019).

[428] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah, "On the Qubit Routing Problem," (2019).

[429] Lingling Lao and Dan E. Browne, "2qan: A quantum compiler for 2-local qubit hamiltonian simulation algorithms," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22 (Association for Computing Machinery, New York, NY, USA, 2022) pp. 351–365.

[430] Yuwei Jin, Jason Luo, Lucent Fong, Yanhao Chen, Ari B. Hayes, Chi Zhang, Fei Hua, and Eddy Z. Zhang, "A Structured Method for Compilation of QAOA Circuits in Quantum Computing," (2022), arXiv:2112.06143 [quant-ph].

[431] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh, "Circuit compilation methodologies for quantum approximate optimization algorithm," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2020) pp. 215–228.

[432] A. D. Córcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Minev, Jerry M. Chow, and Jay M. Gambetta, "Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits," Phys. Rev. Lett. **127**, 100501 (2021).

[433] Elisa Bäumer, Vinay Tripathi, Derek S. Wang, Patrick Rall, Edward H. Chen, Swarnadeep Majumder, Alireza Seif, and Zlatko K. Minev, "Efficient long-range entanglement using dynamic circuits," (2023), arXiv:2308.13065 [quant-ph].

[434] Anette Messinger, Michael Fellner, and Wolfgang Lechner, "Constant Depth Code Deformations in the Parity Architecture," in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (2023) pp. 120–130.

[435] L. Viola and S. Lloyd, "Dynamical suppression of decoherence in two-state quantum systems," Phy. Rev. A **58**, 2733 (1998).

[436] D. Vitali and P. Tombesi, "Using parity kicks for deco-

[437] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, "Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits," Phys. Rev. Lett. **121**, 220502 (2018).

[438] Elijah Pelofske, Andreas Bärtschi, and Stephan Eidenbenz, "Short-depth QAOA circuits and quantum annealing on higher-order ising models," npj Quantum Information **10**, 30 (2024).

[439] Abhinav Kandala, Kristan Temme, Antonio D. Corcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," Nature **567**, 491–495 (2018).

[440] Christophe Piveteau, David Sutter, and Stefan Woerner, "Quasiprobability decompositions with reduced sampling overhead," npj Quantum Information **8** (2022), 10.1038/s41534-022-00517-3.

[441] Daniel Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," (2009), arXiv:0904.2557 [quant-ph].

[442] Ying Li and Simon C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," Physical Review X **7** (2017).

[443] Ryan LaRose, Andrea Mari, Sarah Kaiser, Peter J. Karalekas, Andre A. Alves, Piotr Czarnik, Mohamed El Mandouh, Max H. Gordon, Yousef Hindy, Aaron Robertson, Purva Thakre, Misty Wahl, Danny Samuel, Rahul Mistri, Maxime Tremblay, Nick Gardner, Nathaniel T. Stemen, Nathan Shammah, and William J. Zeng, "Mitiq: A software package for error mitigation on noisy quantum computers," Quantum **6**, 774 (2022).

[444] Almudena Carrera Vazquez, Daniel J. Egger, David Ochsner, and Stefan Woerner, "Well-conditioned multi-product formulas for hardware-friendly hamiltonian simulation," Quantum **7**, 1067 (2023).

[445] Thomas Alexander, Naoki Kanazawa, Daniel J Egger, Lauren Capelluto, Christopher J Wood, Ali Javadi-Abhari, and David C McKay, "Qiskit pulse: programming quantum computers through the cloud with pulses," Quantum Science and Technology **5**, 044006 (2020).

[446] Nathan Earnest, Caroline Tornow, and Daniel J. Egger, "Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware," Phys. Rev. Res. **3**, 043088 (2021).

[447] Alexey Galda, Michael Cubeddu, Naoki Kanazawa, Prineha Narang, and Nathan Earnest-Noble, "Implementing a Ternary Decomposition of the Toffoli Gate on Fixed-Frequency Transmon Qutrits," (2021), arXiv:2109.00558 [quant-ph].

[448] Laurin E. Fischer, Alessandro Chiesa, Francesco Tacchino, Daniel J. Egger, Stefano Carretta, and Ivano Tavernelli, "Universal qudit gate synthesis for transmons," PRX Quantum **4**, 030327 (2023).

[449] D. J. Egger and F. K. Wilhelm, "Optimal control of a quantum measurement," Phys. Rev. A **90**, 052331 (2014).

[450] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong, "Partial compilation of variational algorithms for noisy intermediate-scale quantum machines," in *Pro-*

ceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO '52 (Association for Computing Machinery, New York, NY, USA, 2019) pp. 266–278.

[451] D. T. McClure, Hanhee Paik, L. S. Bishop, M. Steffen, Jerry M. Chow, and Jay M. Gambetta, "Rapid driven reset of a qubit readout resonator," Phys. Rev. Appl. 5, 011001 (2016).

[452] Daniel J. Egger, Chiara Capecci, Bibek Pokharel, Panagiotis Kl. Barkoutsos, Laurin E. Fischer, Leonardo Guidoni, and Ivano Tavernelli, "Pulse variational quantum eigensolver on cross-resonance-based hardware," Phys. Rev. Res. 5, 033159 (2023).

[453] Andrew Wack, Hanhee Paik, Ali Javadi-Abhari, Petar Jurcevic, Ismael Faro, Jay M. Gambetta, and Blake R. Johnson, "Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers," (2021), arXiv:2110.14108 [quant-ph].

[454] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta, "Validating quantum computers using randomized model circuits," Phys. Rev. A 100, 032328 (2019).

[455] Gopal Chandra Santra, Fred Jendrzejewski, Philipp Hauke, and Daniel J. Egger, "Squeezing and quantum approximate optimization," (2023), arXiv:2205.10383 [quant-ph].

[456] "Best practices in quantum optimization," (2023).

[457] Iain Dunning, Swati Gupta, and John Silberholz, "What works best when? a systematic evaluation of heuristics for max-cut and qubo," INFORMS J. Comput. 30, 608–624 (2018).

[458] Raj Jain, The Art of Computer Systems Performance Analysis: Techniques For Experimental Design, Measurement, Simulation, and Modeling (Wiley, 1991).

[459] Thorsten Koch, Timo Berthold, Jaap Pedersen, and Charlie Vanaret, "Progress in mathematical programming solvers from 2001 to 2020," EURO Journal on Computational Optimization 10, 100031 (2022).

[460] Thomas Bartz-Beielstein, Carola Doerr, Daan van den Berg, Jakob Bossek, Sowmya Chandrasekaran, Tome Eftimov, Andreas Fischbach, Pascal Kerschke, William La Cava, Manuel Lopez-Ibanez, Katherine M. Malan, Jason H. Moore, Boris Naujoks, Patryk Orzechowski, Vanessa Volz, Markus Wagner, and Thomas Weise, "Benchmarking in Optimization: Best Practice and Open Issues," (2020), arXiv:2007.03488 [cs.NE].

[461] Matthias Müller, Brian Whitney, Robert Henschel, and Kalyan Kumaran, "Spec benchmarks," in Encyclopedia of Parallel Computing (Springer US, Boston, MA, 2011) pp. 1886–1893.

[462] Jack J Dongarra, Cleve Barry Moler, James R Bunch, and Gilbert W Stewart, LINPACK users' guide (SIAM, 1979).

[463] "DIMACS Implementation Challenges," (2023).

[464] Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, "Sat competition 2020," Artificial Intelligence 301, 103572 (2021).

[465] Emily Yu, Armin Biere, and Keijo Heljanko, "Progress in certifying hardware model checking results," in Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20–23, 2021, Proceedings, Part II 33 (Springer, 2021) pp. 363–386.

[466] Koen Claessen, Niklas Een, Mary Sheeran, and Niklas Sorensson, "Sat-solving in practice," in 2008 9th International Workshop on Discrete Event Systems (IEEE, 2008) pp. 61–67.

[467] Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp M. Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans Mittelmann, Derya Ozyurt, Ted Ralphs, Domenico Salvagnin, and Yuji Shinano, "MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library," Mathematical Programming Computation 13, 443 – 490 (2021).

[468] Hans Mittelmann, "Decision Tree for Optimization Software," (2023).

[469] Thomas Lubinski, Sonika Johri, Paul Varosy, Jeremiah Coleman, Luning Zhao, Jason Necaise, Charles H. Baldwin, Karl Mayer, and Timothy Proctor, "Application-Oriented Performance Benchmarks for Quantum Computing," arXiv preprint (2023), arXiv:2110.03137.

[470] T. Tomesh, P. Gokhale, V. Omole, G. Ravi, K. N. Smith, J. Viszlai, X. Wu, N. Hardavellas, M. R. Martonosi, and F. T. Chong, "SupermarQ: A Scalable Quantum Benchmark Suite," in 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA) (IEEE Computer Society, Los Alamitos, CA, USA, 2022) pp. 587–603.

[471] Simon Martiel, Thomas Ayral, and Cyril Allouche, "Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way," IEEE Transactions on Quantum Engineering 2 (2021), 10.1109/TQE.2021.3090207.

[472] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer, "Defining and detecting quantum speedup," Science 345, 420–424 (2014).

[473] Michael Juenger, Elisabeth Lobe, Petra Mutzel, Gerhard Reinelt, Franz Rendl, Giovanni Rinaldi, and Tobias Stollenwerk, "Performance of a Quantum Annealer for Ising Ground State Computations on Chimera Graphs," (2019), arXiv:1904.11965.

[474] Michael Jünger, Elisabeth Lobe, Petra Mutzel, Gerhard Reinelt, Franz Rendl, Giovanni Rinaldi, and Tobias Stollenwerk, "Quantum annealing versus digital computing: An experimental comparison," ACM J. Exp. Algorithmics 26 (2021), 10.1145/3459606.

[475] Alejandro Perdomo-Ortiz, Alexander Feldman, Asier Ozaeta, Sergei V. Isakov, Zheng Zhu, Bryan O'Gorman, Helmut G. Katzgraber, Alexander Diedrich, Hartmut Neven, Johan de Kleer, Brad Lackey, and Rupak Biswas, "Readiness of quantum optimization machines for industrial applications," Phys. Rev. Appl. 12, 014004 (2019).

[476] Erica Grant, Travis S. Humble, and Benjamin Stump, "Benchmarking quantum annealing controls with portfolio optimization," Phys. Rev. Appl. 15, 014012 (2021).

[477] Byron Tasseff, Tameem Albash, Zachary Morrell, Marc Vuffray, Andrey Y. Lokhov, Sidhant Misra, and Carleton Coffrin, "On the Emerging Potential of Quantum Annealing Hardware for Combinatorial Optimization," (2022), arXiv:2210.04291 [math.OC].

[478] Thomas Lubinski, Carleton Coffrin, Catherine McGeoch, Pratik Sathe, Joshua Apanavicius, and David E. Bernal Neira, "Optimization applications as quantum performance benchmarks," (2024), arXiv:2302.02278

[quant-ph].

[479] Madita Willsch, Dennis Willsch, Fengping Jin, Hans De Raedt, and Kristel Michielsen, "Benchmarking the quantum approximate optimization algorithm," Quantum Information Processing 19 (2020), 10.1007/s11128-020-02692-8.

[480] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," Nature Reviews Physics 4, 363–379 (2022).

[481] Jernej Rudi Finžgar, Philipp Ross, Leonhard Hölscher, Johannes Klepsch, and Andre Luckow, "Quark: A framework for quantum computing application benchmarking," in 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) (2022) pp. 226–237.

[482] Gurobi Optimization, GUROBI Optimizer reference manual version 10.0, Tech. Rep. (Gurobi Optimization, LLC, 2023).

[483] Timo Berthold and Zsolt Csizmadia, "The confined primal integral: a measure to benchmark heuristic minlp solvers against global minlp solvers," Mathematical Programming 188, 523–537 (2021).

[484] Fabio Furini, Emiliano Traversi, Pietro Belotti, Antonio Frangioni, Ambros Gleixner, Nick Gould, Leo Liberti, Andrea Lodi, Ruth Misener, Hans Mittelmann, Nikolaos Sahinidis, Stefan Vigerske, , and Angelika Wiegele, "QPLIB: A library of quadratic programming instances," Mathematical Programming Computation (2018), 10.1007/s12532-018-0147-4.

[485] Holger Hoos and Thomas Stützle, "SATLIB: An online resource for research on SAT," (Kluwer Academic Publishers, 2000) pp. 283–292.

[486] Yunlong Yu, Chenfeng Cao, Xiang-Bin Wang, Nic Shannon, and Robert Joynt, "Solution of sat problems with the adaptive-bias quantum approximate optimization algorithm," Physical Review Research 5, 023147 (2023).

[487] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook, The Traveling Salesman Problem: A Computational Study (Princeton University Press, 2006).

[488] Daniel Rehfeldt, Thorsten Koch, and Yuji Shinano, "Faster exact solution of sparse maxcut and qubo problems," Mathematical Programming Computation 15, 445–470 (2023).

[489] Fred Glover, Mark Lewis, and Gary Kochenberger, "Logical and inequality implications for reducing the size and difficulty of quadratic unconstrained binary optimization problems," European Journal of Operational Research 265, 829–842 (2018).

[490] Sanroku Tsukamoto, M. Takatsu, S. Matsubara, and Hirotaka Tamura, "An accelerator architecture for combinatorial optimization problems," Fujitsu Scientific and Technical Journal 53, 8–13 (2017).

[491] Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki, Yasuhiro Watanabe, Kazuya Takemoto, and Hirotaka Tamura, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC) (2020) pp. 667–672.

[492] Chihiro Yoshimura, Masato Hayashi, Takashi Takemoto, and Masanao Yamaoka, "Cmos annealing machine: A domain-specific architecture for combinatorial optimization problem," in 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC) (2020) pp. 673–678.

[493] Hayato Goto, "Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network," Scientific Reports 6, 21686 (2016).

[494] Juntao Wang, Daniel Ebler, K. Y. Michael Wong, David Shui Wing Hui, and Jie Sun, "Bifurcation behaviors shape how continuous physical dynamics solves discrete ising optimization," Nature Communications 14, 2510 (2023).

[495] Tameem Albash and Daniel A. Lidar, "Adiabatic quantum computation," Rev. Mod. Phys. 90, 015002 (2018).

[496] Catherine C. McGeoch, "Adiabatic quantum computation," in Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice (Springer International Publishing, Cham, 2014) pp. 9–27.

[497] D. Hangleiter, J. Carolan, and K.P.Y. Thébault, Analogue Quantum Simulation: A New Instrument for Scientific Understanding (Springer International Publishing, 2022).

[498] Immanuel Bloch, Jean Dalibard, and Sylvain Nascimbène, "Quantum simulations with ultracold quantum gases," Nature Physics 8, 267–276 (2012).

[499] S. Trotzky, Y-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch, "Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional bose gas," Nature Physics 8, 325–330 (2012).

[500] J.-Y. Choi, Sebastian Hild, Johannes Zeiher, Peter Schauß, Antonio Rubio-Abadal, Tarik Yefsah, Vedika Khemani, David A. Huse, Immanuel Bloch, and Christian Gross, "Exploring the many-body localization transition in two dimensions," Science 352, 1547–1552 (2016).

[501] Maxwell F. Parsons, Anton Mazurenko, Christie S. Chiu, Geoffrey Ji, Daniel Greif, and Markus Greiner, "Site-resolved measurement of the spin-correlation function in the fermi-hubbard model," Science 353, 1253–1256 (2016).

[502] Hannes Bernien, Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin, "Probing many-body dynamics on a 51-atom quantum simulator," Nature 551, 579–584 (2017).

[503] R. Blatt and C. F. Roos, "Quantum simulations with trapped ions," Nature Physics 8, 277–284 (2012).

[504] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe, "Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator," Nature 551, 601–604 (2017).

[505] Iñigo Arrazola, Julen S. Pedernales, Lucas Lamata, and Enrique Solano, "Digital-analog quantum simulation of spin models in trapped ions," Scientific Reports 6, 30534 (2016).

[506] Andrew D. King, Sei Suzuki, Jack Raymond, Alex Zucca, Trevor Lanting, Fabio Altomare, Andrew J. Berkley, Sara Ejtemaee, Emile Hoskinson, Shuiyuan Huang, Eric Ladizinsky, Allison J. R. MacDonald, Gaelen Marsden, Travis Oh, Gabriel Poulin-Lamarre, Mauricio Reis, Chris Rich, Yuki Sato, Jed D. Whittaker, Jason Yao, Richard Harris, Daniel A. Lidar, Hidetoshi

Nishimori, and Mohammad H. Amin, "Coherent quantum annealing in a programmable 2,000 qubit Ising chain," Nature Physics **18**, 1324–1328 (2022).

[507] Andrew D. King, Jack Raymond, Trevor Lanting, Richard Harris, Alex Zucca, Fabio Altomare, Andrew J. Berkley, Kelly Boothby, Sara Ejtemaee, Colin Enderud, Emile Hoskinson, Shuiyuan Huang, Eric Ladizinsky, Allison J. R. MacDonald, Gaelen Marsden, Reza Molavi, Travis Oh, Gabriel Poulin-Lamarre, Mauricio Reis, Chris Rich, Yuki Sato, Nicholas Tsai, Mark Volkmann, Jed D. Whittaker, Jason Yao, Anders W. Sandvik, and Mohammad H. Amin, "Quantum critical dynamics in a 5,000-qubit programmable spin glass," Nature **617**, 61–66 (2023).

[508] Sergio Boixo, Troels F Rønnow, Sergei V Isakov, Zhihui Wang, David Wecker, Daniel A Lidar, John M Martinis, and Matthias Troyer, "Evidence for quantum annealing with more than one hundred qubits," Nat. Phys. **10**, 218–224 (2014).

[509] Vasil S Denchev, Sergio Boixo, Sergei V Isakov, Nan Ding, Ryan Babbush, Vadim Smelyanskiy, John Martinis, and Hartmut Neven, "What is the computational value of finite-range tunneling?" Phys. Rev. X **6**, 031015 (2016).

[510] Hidetoshi Nishimori and Kabuki Takada, "Exponential Enhancement of the Efficiency of Quantum Annealing by Non-Stoquastic Hamiltonians," Frontiers in ICT **4** (2017).

[511] Edson L. Padoin, Daniel A.G. de Oliveira, Pedro Velho, and Philippe O.A. Navaux, "Time-to-solution and energy-to-solution: A comparison between arm and xeon," in *2012 Third Workshop on Applications for Multi-Core Architecture* (2012).

[512] Thomas Lubinski, Carleton Coffrin, Catherine McGeoch, Pratik Sathe, Joshua Apanavicius, and David E. Bernal Neira, "Optimization Applications as Quantum Performance Benchmarks," arXiv preprint (2023), arXiv:2302.02278.

[513] Timo Berthold, "Measuring the impact of primal heuristics," Operations Research Letters **41**, 611–614 (2013).

[514] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2nd ed. (Springer, New York, NY, USA, 2006).

[515] J.F. Bonnans, J.C. Gilbert, C. Lemarechal, and C.A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Universitext (Springer Berlin Heidelberg, 2006).

[516] Masayuki Yamamoto, Masayuki Ohzeki, and Kazuyuki Tanaka, "Fair sampling by simulated annealing on quantum annealer," Journal of the Physical Society of Japan **89**, 025002 (2020).

[517] Mario S. Könz, Guglielmo Mazzola, Andrew J. Ochoa, Helmut G. Katzgraber, and Matthias Troyer, "Uncertain fate of fair sampling in quantum annealing," Phys. Rev. A **100**, 030303 (2019).

[518] Jon Nelson, Marc Vuffray, Andrey Y. Lokhov, Tameem Albash, and Carleton Coffrin, "High-Quality Thermal Gibbs Sampling with Quantum Annealing Hardware," Physical Review Applied **17**, 044046 (2022).

[519] John Golden, Andreas Bärtschi, Daniel O'Malley, and Stephan Eidenbenz, "Fair Sampling Error Analysis on NISQ Devices," ACM Transactions on Quantum Computing **3**, 8:1–8:23 (2022).

[520] Elijah Pelofske, John Golden, Andreas Bärtschi, Daniel

O'Malley, and Stephan Eidenbenz, "Sampling on NISQ Devices: "Who's the Fairest One of All?"," in *IEEE International Conference on Quantum Computing and Engineering QCE'21* (2021) pp. 207–217.

[521] Laurence Wolsey, *Integer Programming: 2nd Edition* (John Wiley & Sons, Ltd, 2020).

[522] Jo Devriendt, Stephan Gocht, Emir Demirović, Jakob Nordström, and Peter J. Stuckey, "Cutting to the core of pseudo-boolean optimization: Combining core-guided search with cutting planes reasoning," Proceedings of the AAAI Conference on Artificial Intelligence **35**, 3750–3758 (2021).

[523] Mukul R. Prasad, Armin Biere, and Aarti Gupta, "A survey of recent advances in SAT-based formal verification," International Journal on Software Tools for Technology Transfer **7**, 156–173 (2005).

[524] Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack, "Minizinc: Towards a standard cp modelling language," in *Principles and Practice of Constraint Programming – CP 2007*, edited by Christian Bessière (Springer Berlin Heidelberg, 2007) pp. 529–543.

[525] "Hexaly," (2023), localSolver.

[526] Ryo Kuroiwa and J. Christopher Beck, "Solving domain-independent dynamic programming problems with anytime heuristic search," Proceedings of the International Conference on Automated Planning and Scheduling **33**, 245–253 (2023).

[527] N. J. A. Sloane, "Challenge problems: Independent sets in graphs," (2000).

[528] Brent N. Clark, Charles J. Colbourn, and David S. Johnson, "Unit disk graphs," Discrete Mathematics **86**, 165–177 (1990).

[529] Ruben S. Andrist, Martin J. A. Schuetz, Pierre Minssen, Romina Yalovetzky, Shouvanik Chakrabarti, Dylan Herman, Niraj Kumar, Grant Salton, Ruslan Shaydulin, Yue Sun, Marco Pistoia, and Helmut G. Katzgraber, "Hardness of the maximum-independent-set problem on unit-disk graphs and prospects for quantum speedups," Physical Review Research **5** (2023), 10.1103/physrevresearch.5.043277.

[530] Hannes Pichler, Sheng-Tao Wang, Leo Zhou, Soonwon Choi, and Mikhail D. Lukin, "Quantum optimization for maximum independent set using rydberg atom arrays," (2018), arXiv:1808.10816 [quant-ph].

[531] Xiaoling Wu, Xinhui Liang, Yaoqi Tian, Fan Yang, Cheng Chen, Yong-Chun Liu, Meng Khoon Tey, and Li You, "A concise review of Rydberg atom based quantum computation and quantum simulation*," Chinese Physics B **30**, 020305 (2021).

[532] Minh-Thi Nguyen, Jin-Guo Liu, Jonathan Wurtz, Mikhail D. Lukin, Sheng-Tao Wang, and Hannes Pichler, "Quantum optimization with arbitrary connectivity using Rydberg atom arrays," PRX Quantum **4**, 010316 (2023).

[533] David Pisinger, "The quadratic knapsack problem—a survey," Discrete Applied Mathematics **155**, 623–648 (2007).

[534] G. Cornuéjols and M. Dawande, "A class of hard small 01 programs," INFORMS Journal on Computing **11**, 205–210 (1999), also published in LNCS 1412 doi: 10.1007/3-540-69346-7_22.

[535] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl, "Qaplib –a quadratic assignment problem library," Jour-

nal of Global Optimization **10**, 391–403 (1997).

[536] David Van Bulck and Dries Goossens, "The international timetabling competition on sports timetabling (itc2021)," European Journal of Operational Research **308**, 1249–1267 (2023).

[537] D. de Werra, "Scheduling in sports," in *Annals of Discrete Mathematics (11)*, North-Holland Mathematics Studies, Vol. 59, edited by P. Hansen (North-Holland, 1981) pp. 381–395.

[538] David Van Bulck, Dries Goossens, Jan-Patrick Clarner, Angelos Dimitsas, George H. G. Fonseca, Carlos Lamas-Fernandez, Martin Mariusz Lester, Jaap Pedersen, Antony E. Phillips, and Roberto Maria Rosati, "Which algorithm to select in sports timetabling?" (2023), arXiv:2309.03229 [cs.AI].

[539] Tameem Albash and Daniel A. Lidar, "Demonstration of a scaling advantage for a quantum annealer over simulated annealing," Phys. Rev. X **8**, 031016 (2018).

[540] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi, "Optimization by simulated annealing," Science **220**, 671–680 (1983).

[541] Giuseppe E Santoro, Roman Martoňák, Erio Tosatti, and Roberto Car, "Theory of quantum annealing of an Ising spin glass," Science **295**, 2427–2430 (2002).

[542] Francesco Slongo, Philipp Hauke, Pietro Faccioli, and Cristian Micheletti, "Quantum-inspired encoding enhances stochastic sampling of soft matter systems," Science Advances **9** (2023).

[543] Abraham P. Punnen, ed., *The Quadratic Unconstrained Binary Optimization Problem* (Springer, 2022).

[544] Ruslan Shaydulin and Marco Pistoia, "QAOA with $N \cdot p \geq 200$," in *IEEE International Conference on Quantum Computing and Engineering QCE'23* (2023) pp. 1074–1077.

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

[546] Filip B Maciejewski, Jacob Biamonte, Stuart Hadfield, and Davide Venturelli, "Improving quantum approximate optimization by noise-directed adaptive remapping," arXiv preprint arXiv:2404.01412 (2024).

[547] Elijah Pelofske, Andreas Bärtschi, Lukasz Cincio, John Golden, and Stephan Eidenbenz, "Scaling Whole-Chip QAOA for Higher-Order Ising Spin Glass Models on Heavy-Hex Graphs," (2023), arXiv:2312.00997 [quant-ph].

[548] Qiskit contributors, "Qiskit: An Open-source Framework for Quantum Computing," (2023).

[549] Daniel J. Egger, Claudio. Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain, "Quantum computing for finance: state of the art and future prospects," IEEE Transactions on Quantum Engineering **1**, 1–24 (2020).

[550] Roman Orus, Samuel Mugel, and Enrique Lizaso, "Quantum computing for finance: Overview and prospects," Rev. Phys. **4**, 100028 (2019).

[551] Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alexeev, "Quantum computing for finance," Nature Reviews Physics **5**, 450–465 (2023).

[552] Casey Berger, Agustin Di Paolo, Tracey Forrest, Stuart Hadfield, Nicolas Sawaya, Michał Stęchły, and Karl Thibault, "Quantum technologies for climate change: Preliminary assessment," (2021), arXiv:2107.05362 [quant-ph].

[553] Yifan Zhou, Zefan Tang, Nima Nikmehr, Pouya Babahajiani, Fei Feng, Tzu-Chieh Wei, Honghao Zheng, and Peng Zhang, "Quantum computing in power systems," iEnergy **1**, 170–187 (2022).

[554] Hari P. Paudel, Madhava Syamlal, Scott E. Crawford, Yueh-Lin Lee, Roman A. Shugayev, Ping Lu, Paul R. Ohodnicki, Darren Mollot, and Yuhua Duan, "Quantum computing and simulations for energy applications: Review and perspective," ACS Engineering **2**, 151–196 (2022).

[555] Stavros Zenios, *Practical Financial Optimization. Decision making for financial engineers* (Wiley-Blackwell, 2007).

[556] Shanker Ramamurthy, John J Duigenan, Hans Tesselaar, Hector Arias, and Paolo Sironi, "Embedded finance: Creating the everywhere, everyday bank," (2023).

[557] L. Bachelier, "Theory of speculation," (1900).

[558] F. A. Hayek, "The use of knowledge in society," The American Economic Review **35**, 519–530 (1945).

[559] Raphael A Espinoza, Miguel A. Segoviano, Ji Yan, "Systemic risk modeling: How theory can meet statistics," (2020).

[560] European Central Bank, "Financial stability review," (2023).

[561] International Monetary Fund, "Global financial stability report: Financial and climate policies for a high-interest-rate era," (2023).

[562] European Comission, "Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts," (2023).

[563] Basel Committee on Banking Supervision, Bank of International Settlements, "Climate-related risk drivers and their transmission channels," (2021).

[564] Tom Donilon, Catherine Kress, and Stephanie Lee Jackson Spivey, "Geopolitical risk dashboard," (2023).

[565] Fishcer Black and Myron Scholes, "The pricing of options and corporate liabilities," Journal of political economy **81**, 637 (1973).

[566] Rama Cont, "Volatility clustering in financial markets: Empirical facts and agent-based models," in *Long Memory in Economics* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007) pp. 289–309.

[567] Tim Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of Econometrics **31**, 307–327 (1986).

[568] Steven L. Heston, "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options," The Review of Financial Studies **6**, 327–343 (1993).

[569] J. Muzy, J. Delour, and E. Bacry, "Modelling fluctuations of financial time series: from cascade process to stochastic volatility model," Eur. Phys. J. B 17 , 537–548 (2000).

[570] Thibault Jaisson Jim Gatheral and Mathieu Rosenbaum, "Volatility is rough," Quantitative Finance **18**, 933–949 (2018).

[571] J. Donier P. Blanc and J.-P. Bouchaud, "Quadratic

70

hawkes processes for financial prices," Quantitative Finance **17**, 171–188 (2017).

[572] Aditi Dandapani, Paul Jusselin, and Mathieu Rosenbaum, "From quadratic Hawkes processes to super-Heston rough volatility models with Zumbach effect," (2021), arXiv:1907.06151 [q-fin.ST].

[573] Rama Cont, "Empirical properties of asset returns: stylized facts and statistical issues," Quantitative Finance **1**, 223 (2001).

[574] Jayakrishnan Nair, Adam Wierman, and Bert Zwart, *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*, Cambridge Series in Statistical and Probabilistic Mathematics (Cambridge University Press, 2022).

[575] Organisation for Economic Co-operation and Development, "Environmental social and governance (esg) investing," (2023).

[576] Sebastian Brandhofer, Daniel Braun, Vanessa Dehn, Gerhard Hellstern, Matthias Hüls, Yanjun Ji, Ilia Polian, Amandeep Singh Bhatia, and Thomas Wellens, "Benchmarking the performance of portfolio optimization with QAOA," Quantum Information Processing **22**, 25 (2022).

[577] Mark Hodson, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman, "Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz," (2019), arXiv:1911.05296 [quant-ph].

[578] Samuel Mugel, Carlos Kuchkovsky, Escolástico Sánchez, Samuel Fernández-Lorenzo, Jorge Luis-Hita, Enrique Lizaso, and Román Orús, "Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks," Physical Review Research **4**, 013006 (2022).

[579] Davide Venturelli and Alexei Kondratyev, "Reverse quantum annealing approach to portfolio optimization problems," Quantum Machine Intelligence **1**, 17–30 (2019).

[580] Megan C. Giron, Georgios Korpas, Waqas Parvaiz, Prashant Malik, and Johannes Aspman, "Approaching Collateral Optimization for NISQ and Quantum-Inspired Computing," IEEE Transactions on Quantum Engineering **4**, 1–18 (2023).

[581] Stefan Woerner and Daniel J. Egger, "Quantum risk analysis," npj Quantum Inf. **5**, 15 (2019).

[582] Daniel J. Egger, Ricardo G. Gutierrez, Jordi Cahue Mestre, and Stefan Woerner, "Credit risk analysis using quantum computers," IEEE Trans. Comput. **70**, 1–1 (5555).

[583] Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner, "Option pricing using quantum computers," Quantum **4**, 291 (2020).

[584] Cristian S. Calude, Michael J. Dinneen, and Richard Hua, "Qubo formulations for the graph isomorphism problem and related problems," Theoretical Computer Science **701**, 54–69 (2017).

[585] Petter N. Kolm, Reha Tütüncü, and Frank J. Fabozzi, "60 years of portfolio optimization: Practical challenges and current trends," European Journal of Operational Research **234**, 356–371 (2014), 60 years following Harry Markowitz's contribution to portfolio theory and operations research.

[586] Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alex-

eev, "Quantum computing for finance," Nature Reviews Physics **5**, 450–465 (2023).

[587] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão, "Quantum algorithms: A survey of applications and end-to-end complexities," (2023), arXiv:2310.03011 [quant-ph].

[588] Harry Markowitz, "Portfolio selection," The Journal of Finance **7**, 77–91 (1952).

[589] Aharon Ben-Tal and Arkadi Nemirovski, *Lectures on Modern Convex Optimization* (Society for Industrial and Applied Mathematics, 2001).

[590] Robert C. Merton, "An analytic derivation of the efficient portfolio frontier," The Journal of Financial and Quantitative Analysis **7**, 1851–1872 (1972).

[591] Nicholas I. M. Gould, Mary E. Hribar, and Jorge Nocedal, "On the solution of equality constrained quadratic programming problems arising in optimization," SIAM Journal on Scientific Computing **23**, 1376–1395 (2001).

[592] William Karush, *Minima of Functions of Several Variables with Inequalities as Side Conditions*, Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA (1939).

[593] Fred Glover, Gary Kochenberger, and Yu Du, "Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models," 4OR **17**, 335–371 (2019).

[594] Richard O. Michaud and Tongshu Ma, "Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation." The Review of Financial Studies **14**, 901–904 (2015).

[595] Fischer Black and Robert Litterman, "Global portfolio optimization," Financial Analysts Journal **48**, 28–43 (1992).

[596] Doron Avramov and Guofu Zhou, "Bayesian portfolio analysis," Annual Review of Financial Economics **2**, 25–47 (2010).

[597] Gérard Cornuéjols, Javier Peña, and Reha Tütüncü, *Optimization Methods in Finance*, 2nd ed. (Cambridge University Press, 2018).

[598] M. L. Mehta, "Random matrices," (2004).

[599] Eugene P. Wigner, "Characteristic vectors of bordered matrices with infinite dimensions," Annals of Mathematics **62**, 548–564 (1955).

[600] John von Neumann and H. H. Goldstine, "Numerical inverting of matrices of high order," Bulletin of the American Mathematical Society **53**, 1021 – 1099 (1947).

[601] Olivier Ledoit and Sandrine Péché, "Eigenvectors of some large sample covariance matrix ensembles," Probability Theory and Related Fields **151**, 233–264 (2011).

[602] Joël Bun, Romain Allez, Jean-Philippe Bouchaud, and Marc Potters, "Rotational invariant estimator for general noisy matrices," IEEE Transactions on Information Theory **62**, 7475–7490 (2016).

[603] Florent Benaych-Georges, Jean-Philippe Bouchaud, and Marc Potters, "Optimal cleaning for singular values of cross-covariance matrices," (2021), arXiv:1901.05543 [math.ST].

[604] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters, "Cleaning large correlation matrices: Tools from random matrix theory," Physics Reports **666**, 1–109 (2017), cleaning large correlation matrices: tools from random matrix theory.

[605] Olivier Ledoit and Michael Wolf, "Nonlinear Shrinkage of the Covariance Matrix for Portfolio Selection: Markowitz Meets Goldilocks," The Review of Financial Studies **30**, 4349–4388 (2017).

[606] Gary Chamberlain and Michael Rothschild, "Arbitrage, factor structure, and mean-variance analysis on large asset markets," Econometrica **51**, 1281–1304 (1983).

[607] Richard O. Michaud and Robert O. Michaud, *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*, OUP Catalogue No. 9780195331912 (Oxford University Press, 2008).

[608] Marcos Prado, "A robust estimator of the efficient frontier," SSRN Electronic Journal (2019), 10.2139/ssrn.3469961.

[609] Pierre-Alain Reigneron, Romain Allez, and Jean-Philippe Bouchaud, "Principal regression analysis and the index leverage effect," Physica A: Statistical Mechanics and its Applications **390**, 3026–3035 (2011).

[610] Armine Karami, Raphael Benichou, Michael Benzaquen, and Jean-Philippe Bouchaud, "Conditional correlations and principal regression analysis for futures," Wilmott **2021**, 63–73 (2021).

[611] Vaclav Smil, "Distributed generation and megacities: Are renewables the answer?" IEEE Power and Energy Magazine **17**, 37–41 (2019).

[612] Naser Hossein Motlagh, Mahsa Mohammadrezaei, Julian Hunt, and Behnam Zakeri, "Internet of things (IoT) and the energy sector," Energies **13** (2020), 10.3390/en13020494.

[613] Guneet Bedi, Ganesh Kumar Venayagamoorthy, Rajendra Singh, Richard R. Brooks, and Kuang-Ching Wang, "Review of internet of things (iot) in electric power and energy systems," IEEE Internet of Things Journal **5**, 847–870 (2018).

[614] N.P. Padhy, "Unit commitment-a bibliographical survey," IEEE Transactions on Power Systems **19**, 1196–1205 (2004).

[615] Samantha Koretsky, Pranav Gokhale, Jonathan M. Baker, Joshua Viszlai, Honghao Zheng, Niroj Gurung, Ryan Burg, Esa Aleksi Paaso, Amin Khodaei, Rozhin Eskandarpour, and Frederic T. Chong, "Adapting quantum approximation optimization algorithm (qaoa) for unit commitment," in *Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2021) pp. 181–187.

[616] Masaya Takahashi, Hiroaki Nishioka, Masahiro Hirai, and Hidetaka Takano, "A study of the optimization problem on the combination of sectionalizing switches in power grid with quantum annealing," (2023), arXiv:2305.07899 [quant-ph].

[617] Pascal Halffmann, Patrick Holzer, Kai Plociennik, and Michael Trebing, "A quantum computing approach for the unit commitment problem," in *International Conference on Operations Research*, edited by O. Grothe, S. Nickel, S. Rebennack, and O. Stein (Springer, 2022) pp. 113–120.

[618] M. C. Braun, T. Decker, N. Hegemann, S. F. Kerstan, and F. Lorenz, "Towards optimization under uncertainty for fundamental models in energy markets using quantum computers," (2023), arXiv:2301.01108 [quant-ph].

[619] Reza Mahroo and Amin Kargarian, "Hybrid quantum-classical unit commitment," in *2022 IEEE Texas Power and Energy Conference (TPEC)* (IEEE, 2022).

[620] Giuseppe Colucci, Stan van der Linde, and Frank Phillipson, "Power network optimization: A quantum approach," IEEE Access **11**, 98926–98938 (2023).

[621] Thomas Morstyn, "Annealing-based quantum computing for combinatorial optimal power flow," IEEE Transactions on Smart Grid **14**, 1093–1102 (2023).

[622] Philipp Schworm, Xiangqian Wu, Marcel Wagner, Svenja Ehmsen, Moritz Glatt, and Jan C. Aurich, "Energy supply scheduling in manufacturing systems using quantum annealing," Manufacturing Letters **38**, 47–51 (2023).

[623] European Comission, "Quarterly report on European electricity markets. Volume 15. Issue 4. ," (2023).

[624] Financial Times, "France sets course for a nuclear renaissance," (2023).

[625] INTERNATIONAL ENERGY AGENCY, "Global EV Outlook 2023," (2023).

[626] Renewable Energy World, "Aggreko delivers 2 MW/3.8 MWh energy storage system for National Grid US," (2023).

[627] Stergios Statharas, Yannis Moysoglou, Pelopidas Siskos, Georgios Zazias, and Pantelis Capros, "Factors influencing electric vehicle penetration in the EU by 2030: A model-based policy assessment," Energies **12** (2019), 10.3390/en12142739.

[628] Henrik Lund and Willett Kempton, "Integration of renewable energy into the transport and electricity sectors through V2G," Energy Policy **36**, 3578–3587 (2008).

[629] Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny H.K. Tsang, "Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging," Proc. ACM Meas. Anal. Comput. Syst. **4** (2021), 10.1145/3428336.

[630] Bo Sun, Lin Yang, Mohammad Hajiesmaili, Adam Wierman, John C. S. Lui, Don Towsley, and Danny H.K. Tsang, "The online knapsack problem with departures," Proc. ACM Meas. Anal. Comput. Syst. **6** (2022), 10.1145/3570618.

[631] Bernard Knueven, James Ostrowski, and Jean-Paul Watson, "On mixed-integer programming formulations for the unit commitment problem," INFORMS Journal on Computing **32**, 857–876 (2020).

[632] Marika Federer, Daniel Müssig, Stefan Klaiber, Jörg Lässig, Peter Bretschneider, and Steve Lenk, "Application benchmark for quantum optimization on electromobility use case," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)* (2022) pp. 1–6.

[633] Arkadi Nemirovski and Alexander Shapiro, "Convex approximations of chance constrained programs," SIAM Journal on Optimization **17**, 969–996 (2007).

[634] Oded Goldreich, "On promise problems (a survey in memory of shimon even [1935-2004])," ECCC, TR05-018 **127**, 128 (2005).