



Discrete Optimization

Quantum computing for discrete optimization: A highlight of three technologies

Alexey Bochkarev^a*, Raoul Heese^b, Sven Jäger^a, Philine Schiewe^c, Anita Schöbel^{a,b}^a Department of Mathematics, RPTU Kaiserslautern-Landau, Kaiserslautern, 67663, Germany^b Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, 67663, Germany^c Department of Mathematics and Systems Analysis, Aalto University, Espoo, 02150, Finland

ARTICLE INFO

Keywords:

Combinatorial optimization
Heuristics

ABSTRACT

Quantum optimization has emerged as a promising frontier of quantum computing, providing novel numerical approaches to mathematical optimization problems. The main goal of this paper is to facilitate interdisciplinary research between the Operations Research (OR) and quantum computing communities by helping OR scientists to build initial intuition for-, and offering them a hands-on gateway to quantum-powered methods in the context of discrete optimization. To this end, we consider three quantum-powered optimization approaches that make use of different types of quantum hardware available on the market. To illustrate these approaches, we solve three classical optimization problems: the Traveling Salesperson Problem, Weighted Maximum Cut, and Maximum Independent Set. With a general OR audience in mind, we attempt to provide an intuition behind each approach along with key references, describe the corresponding high-level workflow, and highlight crucial practical considerations. In particular, we emphasize the importance of problem formulations and device-specific configurations, and their impact on the amount of resources required for computation (where we focus on the number of qubits). These points are illustrated with a series of experiments on three types of quantum computers: a neutral atom machine from *QuEra*, a quantum annealer from *D-Wave*, and gate-based devices from *IBM*.

1. Introduction

Discrete optimization is central to many problems in OR, often arising in efficient organization of complex systems across various domains: logistics, supply chain management, transportation, finance, healthcare, and more. This led to the development of a vast variety of computational methods, with many state-of-the-art algorithms implemented in advanced stand-alone solvers, such as [Gurobi \(2024\)](#), [SCIP \(Bolusani et al., 2024\)](#), and others. Still, practical problems often require substantial resources, as the solution space grows exponentially with the problem size.

Quantum computing offers an alternative computation model, but both the methodology and the hardware are in early stages of development, as compared to classical¹ computing. It is unclear to what extent, if at all, OR will benefit from quantum technology in the near future. There are reasons to believe that it might yield a significant speed-up, especially for discrete optimization ([Preskill, 2018](#)), although no practical (exponential) quantum advantage has yet been demonstrated

for an optimization problem ([Hoefer et al., 2023](#)). We think that to be able to assess potential opportunities at this early stage, OR experts might want to familiarize themselves with the topic to a certain degree. However, the necessary background knowledge significantly differs from a typical discrete optimization specialist training. This is mainly because the language used in the literature naturally draws on the works from quantum physics, making it less accessible to the general mathematical readership.

Existing literature already provides overviews of the current state of research. A discussion of possible synergies between OR and quantum information science along with further research directions is presented by [Parekh \(2023\)](#). [Au-Yeung et al. \(2023\)](#) and [Klug \(2024\)](#) discuss quantum optimization; a more comprehensive overview is provided by [Abbas et al. \(2024\)](#). The current state and prospects of QCs are discussed by [Scholten et al. \(2024\)](#). There is also a significant body of literature devoted to numerical investigation and benchmarking of quantum-powered approaches, for example [Koch et al. \(2025\)](#) and [Lubinski et al. \(2024\)](#).

* Corresponding author.

E-mail address: a.bochkarev@rptu.de (A. Bochkarev).¹ We use the term “classical” throughout this paper to distinguish non-quantum technologies from quantum technologies. In this sense, a “classical computer” is a conventional computing device that uses non-quantum principles of information processing.

This paper differs fundamentally from the overviews mentioned above. Rather than providing a general overview, we focus on specific quantum devices and types of discrete optimization problems, and seek to achieve three goals: convey the underlying intuition of each approach, provide the necessary references for a deeper understanding, and discuss relevant workflows and key practical considerations. An important aspect that we take into account is the required number of qubits for optimization tasks, as it is an easy-to-understand yet important metric. Our findings highlight the fact that the number of qubits required to solve an optimization problem depends not only on the number of binary variables, but also on the problem structure, formulation, and the specific quantum device. We illustrate this aspect with a series of experiments.

As with any scientific work, this paper represents the state of the art at the time of writing. Quantum computing is a rapidly evolving field, with continuous advances in both hardware and algorithms. Although specific hardware implementations may change and improved algorithms may emerge, we believe that the basic principles discussed here will continue to be relevant for understanding the advances and challenges in quantum optimization in the near future. By analyzing current technologies, we aim to provide insights that will be valuable both now and as the field continues to evolve.

The paper is structured as follows. In Section 2, we provide a brief introduction to quantum optimization, where we focus on the motivation for applying quantum computing to discrete optimization. Subsequently, we present three distinct quantum technologies in Section 3, focusing on their optimization workflows. In Section 4, we discuss the application of these workflows to three specific discrete optimization problems, highlighting the necessary number of qubits in connection to the chosen computational approach. This discussion is further illustrated with numerical results in Section 5. Finally, we conclude with a summary and brief outlook in Section 6.

2. Quantum optimization

A quantum computer (QC) is a computational device that harnesses quantum physics to process information (Nielsen & Chuang, 2010). Quantum information processing is fundamentally different from its classical counterpart, and offers the possibility of performing certain computations beyond the classically achievable performance. This makes QCs a potentially powerful tool for solving optimization problems. However, while the potential is significant, improvements on both the algorithmic and hardware side are currently needed to realize an advantage for practical applications (Abbas et al., 2024). Despite these obstacles, we believe this somewhat early stage is a particularly good moment to study the topic from an OR perspective, in order to better understand current limitations and assess possible future opportunities. In the following, we lay the foundation for such an investigation. For this purpose, we outline how QCs work and motivate their use for optimization tasks in light of the selected quantum computing technologies.

It is important to note that quantum physics can be described using different mathematical formalisms, each based on different principles, but each equally capable of accurately predicting experimental results (Styer et al., 2002). One of the most widely used approaches in quantum computing involves the linear algebraic descriptions, where the states of a quantum system are represented by vectors in Hilbert space, and the state transformations are described in terms of linear operators (matrices). For a rigorous treatment of this framework, we refer the reader to standard textbooks, such as Mermin (2007) or Nielsen and Chuang (2010).

In the scope of the present work, we can only provide a high-level summary of the underlying concepts. Therefore, we use slightly oversimplified explanations where necessary and try to omit the mathematical background of quantum physics where possible. More in-depth information can be found in the cited references.

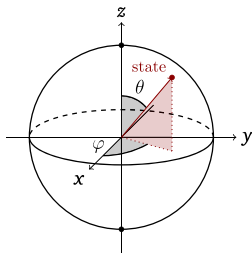
2.1. Quantum computing

A QC is typically a highly complex device with many different interoperating components, with the QPU being the core information processing unit within a QC, in analogy to the central processing unit (CPU) within a conventional computer. There are generally two modes of operation for a QPU: the *analog mode* and the *digital mode*, both of which are considered in Section 3. In analog mode, computations are realized with a continuous control of an underlying quantum system, similar to how quantum systems behave in nature due to the laws of quantum physics. This idea is closely related to classical analog computing, where the natural behavior of physical systems is exploited to store and process information (MacLennan, 2009; Ullmann, 2024; Wu et al., 2022; Zangeneh-Nejad et al., 2021). The most popular approach for this mode of operation, adiabatic quantum computing (AQC), is discussed in Section 2.4. The digital mode, on the other hand, is based on discrete controls, so-called *gates* (in analogy to classical logical gates, such as AND or XOR), and is therefore also commonly referred to as gate-based quantum computing (GQC). Gates are effectively a discretization of the underlying continuous controls, making the digital mode an additional level of abstraction over the analog mode. Notably, Nannicini (2020) attempts to provide a relatively compact and self-contained introduction to gate-based quantum computing, without relying on the ideas from physics.

Regardless of whether it is operated in analog or digital mode, a QPU works with quantum bits, or *qubits*, to process information. These qubits exhibit a much more complex behavior than classical bits. The main difference is that a classical bit has, by definition, two binary states, typically denoted by 0 and 1. A qubit, on the other hand, can attain infinitely many states because its state space is a continuous spectrum of possible configurations. Based on the underlying theory of quantum physics, each state in this continuous state space can be identified with two coordinates. Due to the state space topology, a commonly used interpretation is to identify the two coordinates as angles that address a point on the surface of a unit sphere. This interpretation allows visualizing the state space of a qubit in form of the so-called Bloch sphere, as shown in Fig. 1.

This fundamental difference between the state spaces of classical bits and qubits is crucial to understand the difference between classical and quantum information processing. In simple terms, one could argue that the information encoding of a qubit is much *denser* than of a classical bit. In fact, a single qubit could in principle store an infinite amount of information with its infinitely large state space (Nielsen & Chuang, 2010). However, this would require an infinitely precise qubit control, which is not feasible in practice. And even if we could achieve infinitely precise control over qubits, fundamental principles of quantum mechanics still limit the amount of information that can be accessed. This is formalized by the *Holevo bound*, which states that the amount of classical information that can be extracted from n qubits is at most equivalent to that of n classical bits (Nielsen & Chuang, 2010). The advantage of quantum computing therefore lies less in the efficient storage of data than in its efficient processing. However, being able to encode information in a dense way does not necessarily mean that processing such information is automatically efficient. In contrast, a major challenge for quantum computing is precisely how to exploit the large state space of qubits for efficient computations.

Performing computations is equivalent to processing data, and the question arises as to how this can be done with a qubit. Since information is encoded using the principles of quantum physics, any interaction with this information also takes place on the quantum level. In this context, there is one important feature that strongly distinguishes classical from quantum information processing: the non-deterministic nature of quantum physics. Non-determinism affects how the information that has been stored in a qubit can be read out. For a classical bit, this is a trivial task since any bit that has been written can also be read in the same way without any loss of information. The situation is different for



Every possible qubit state is represented by a point on the surface of this (three-dimensional) sphere. The axes of the underlying Cartesian coordinate system are labeled with x , y , and z . Two angles, $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi)$, are sufficient to identify any state, as shown for an example state. The north pole and south pole of the sphere represent two classical states in analogy to the two states of a classical bit.

Fig. 1. The Bloch sphere: a visualization of the state space of a single qubit.

qubits. Reading information from a qubit, also known as *measuring* the qubit, means to interact with it on a quantum level. According to the laws of quantum physics, such a measurement is always a probabilistic process that reveals a binary outcome. The two possible measurement outcomes can for example be labeled 0 or 1 in agreement with the states of a classical bit.

The chance of measuring either 0 or 1 depends on the state of the qubit before the measurement. In other words, each qubit state represents a distinct probability distribution that determines its probabilistic measurement process. Again, the Bloch sphere can be used to develop an understanding for this highly counter-intuitive behavior. The north pole and the south pole of the Bloch sphere represent the only two states with an effectively deterministic measurement process. The north pole is also called the *ground state* and the south pole the *excited state* of the qubit (for reasons we explain further below). By definition, measuring a qubit in the excited state always yields 1 and, conversely, measuring a qubit in the ground state always yields 0. Both state are therefore also referred to as *classical states* in analogy to the two states of a classical bit. For any other state on the Bloch sphere (also called *superposition states*), the probability distribution of measurement outcomes depends on its spherical distance to the north pole representing the ground state (or, conversely, the south pole representing the excited state). The closer it is to a respective pole, the more probable the corresponding outcome. In other words, the closer a state is located to one of the poles, the more biased is the measurement. All states on the northern hemisphere have a bias towards 0, all states on the southern hemisphere have a bias towards 1, and all states on the equator have an equal chance of a measurement outcome of 0 or 1, as for an ideal coin flip.

It is important to clarify that this non-deterministic behavior is not the result of technical limitations, but an intrinsic property of quantum physics (Bera et al., 2017). In addition to being non-deterministic, measurements also are inherently *destructive*. This means that when a measurement yields an outcome of 0 (1), the state of the qubit is changed from its original state into the ground (excited) state and a subsequent measurement will always result in the same measurement outcome. In other words, the information encoded in a qubit is lost after measuring it once. The significance of single measurements may therefore be limited for quantum information processing, and it is often the case that quantum computations need to be repeated many times to get meaningful measurement results and algorithmic designs that can operate on finite samples (i.e., measurement results) of the underlying joint probability distribution. We also discuss this limitation in Section 3.

To this point, we have mostly focused on a single qubit, but meaningful quantum information processing requires multiple qubits in the same sense as multiple bits are required for meaningful classical information processing. Similar to how information is stored more densely in qubits than in classical bits, the interaction between qubits is also much more complicated than the interaction between classical bits. First and foremost, multi-qubit systems can exhibit *entanglement*, a unique quantum phenomenon in which the state of one qubit cannot be described independently of the others. As a result, operations performed on one qubit may instantaneously alter the state of the entire quantum system.

This is fundamentally different from classical information processing, where each bit can always be flipped independently of all others. Entanglement is a key feature used in many quantum algorithms. It enables *quantum parallelism* (Markidis, 2024), a distinct computing paradigm for quantum computers which is not equivalent to mere parallel computation, as different branches of computation can actually interact, an effect also known as *interference*. Quantum parallelism does not necessarily lead to a computational advantage, but can be used to design quantum algorithms that may benefit from entanglement. As a prototypical example, the Deutsch-Josza algorithm (Nielsen & Chuang, 2010) determines a global property of a Boolean function using a single quantum evaluation.

In analogy to the measurement of a single qubit, the measurement of a multi-qubit system behaves non-deterministically, however, the results of measurements of the entangled qubits are correlated. Generally, the measurement results of an n -qubit system can therefore be viewed as samples drawn from a joint probability distribution with a number of parameters exponential in n . This may (or may not) address the limitations of classical systems in some cases, when an exponentially large state space leads to practical intractability of the problem at hand.

2.2. Quantum hardware

In addition to the fundamental challenges of quantum computing due to the unique computational model, an additional practical challenge lies in the technological limitations of the hardware. QCs are currently very error-prone and have extremely limited resources, which is why they are also sometimes called noisy intermediate-scale quantum (NISQ) devices. A long-term goal for hardware development is fault-tolerant quantum computing (FTQC), where the effects of hardware noise are completely eliminated using methods such as quantum error correction (QEC). While there is active research and continuous improvement (Gottesman, 2022; Katabarwa et al., 2024), fault-tolerant computations are not yet feasible at any practical scale. Therefore, current algorithms must take these hardware-induced errors and limitations into account.

The physical realization of qubits requires quantum systems that can attain at least two distinct quantum states, which are used to represent 0 and 1. In practice, these states are typically chosen as so-called *energy states* (which describe the energy of a system), since energy levels in many quantum systems naturally occur in discrete levels. For a single-qubit system, we have referred to them as the ground state and the excited state, respectively, a terminology that becomes clear in this context: the ground state corresponds to the lower energy state of the qubit (representing 0), while the excited state corresponds to a higher energy state (representing 1). These energy levels provide a practical and intuitive basis for labeling the two poles of the Bloch sphere. Computations can then be realized by shifting qubit states between different energy levels. For a multi-qubit system, while a measurement still yields a single binary digit per qubit, the system is usually configured in a problem-dependent way to rearrange its energy levels. The usual goal is for a measurement of the ground state (i.e., the minimum-energy state) not to yield a trivial bitstring of zeroes, but reveal some meaningful information about the problem. In other

words, the ground state of a multi-qubit system can be used to encode information. We will revisit this topic in Section 2.4.

There are many different quantum hardware providers competing for the best solution and offering different kinds of devices based on different kinds of technologies that may each have their unique advantages and disadvantages. Most devices can be accessed remotely via an online interface or platform, eliminating the need for physical proximity between QC and user. Practically usable QCs include superconducting devices (e.g., by IBM, D-Wave, Google, IQM, and Rigetti), photonic devices (e.g., by Xanadu), neutral atom devices (e.g., by QuEra, Pasqal, and RymaxOne), and trapped ion devices (e.g., by Quantinuum, Honeywell QS, and IonQ), just to name a few (Gyongyosi & Imre, 2019). In Section 3, we consider three quantum-powered optimization approaches running on distinct QCs to illustrate how differently operating devices can still be summarized under a unified workflow.

Throughout this work, we use the term *logical qubit* to refer to an abstract (idealized) qubit at the algorithmic level of the underlying quantum computing model, but independent of the hardware implementation.² Conversely, a *physical qubit* refers to the actual implementation on a hardware device, which may differ in its physical nature from device to device. A central question in quantum optimization is whether the quantum hardware of choice has a sufficient number of physical qubits to process a given task. In practice, determining the required number of qubits typically involves a two-step process. First, the mathematical optimization problem must be translated into an abstract quantum algorithm, which requires a specific number of logical qubits. This number typically scales with the size of the optimization problem. Second, the abstract algorithm must then be mapped to a hardware-specific realization, which requires a certain number of physical qubits. This number is often significantly higher than the number of logical qubits, as we will describe in Sections 3.1 to 3.3.

2.3. Quantum advantage

There is evidence that also without FTQC, quantum computations can be practically relevant for certain use cases (Kim et al., 2023). However, identifying such use cases is not trivial. Because of the different computational model, a one-to-one implementation of classical algorithms on QCs usually do not lead to performance gains (in fact, the opposite is true). Specialized solutions for carefully selected problem classes are needed to make the most of the hardware (Harrow & Montanaro, 2017). In this sense, the role of a QPU can be compared to that of a graphics processing unit (GPU). A GPU has a clear advantage over a CPU in certain specialized tasks, such as parallel numerical computation. Likewise, a QPU has the potential to offer significant benefits in solving specific quantum-suited problems. However, like a GPU, a QPU cannot be expected to be a general-purpose problem solver.

Demonstrating that a quantum computer can solve a problem significantly better than a classical computer (typically implying a superpolynomial speedup) is also known as a *quantum advantage* or *quantum supremacy* (Harrow & Montanaro, 2017). There is theoretical and experimental evidence for quantum advantage in tasks such as *random circuit sampling* (Hangleiter & Eisert, 2023), but the practical utility of these tasks remains limited. Moreover, there are quantum algorithms that have a purely theoretical advantage when run on noise-free hardware. For example, Grover's search algorithm (Grover, 1996) is mathematically proven to provide a quadratic speedup to unstructured search problems, but there has not been any experimental verification of significant scale yet. Preskill (2018) identified three main reasons why a quantum advantage could potentially be achieved with sufficient technological advancement:

1. There are examples of mathematical problems that are believed hard to solve for classical computers, and for which efficient quantum-powered algorithms exist (e.g., factoring integers and finding discrete logarithms, as studied in Shor, 1997).
2. There are quantum states that are relatively easy to prepare on a QC such that measuring them is equivalent to the sampling of random numbers from a particular probability distribution, which is impossible to achieve efficiently by classical means (Harrow & Montanaro, 2017).
3. No efficient classical algorithm is known that is able to simulate a QC efficiently at arbitrary scales, which is essentially due to the exponentially large state space of multi-qubit systems.

It is not immediately obvious how these points translate to practical applications in the OR context. In general, it can be expected that problems that can be solved efficiently by classical methods today are likely to remain in the domain of classical computing. On the other hand, problems that are currently intractable or that require an excessive amount of computational resources are promising candidates for a quantum algorithm. Among these, discrete optimization is a particularly notable application domain (Koch et al., 2025), which justifies to study this topic from an OR perspective. Despite this premise, no clear proof of a quantum advantage for optimization has been achieved yet (Abbas et al., 2024). Although obtaining a quantum advantage remains a key goal, it is not a strict requirement for achieving practical benefits. Another perspective is that quantum computers only need to perform reliable computations beyond the scope of brute-force classical computations to achieve a practical benefit, which is referred to as *quantum utility* (Kim et al., 2023).

Quantum computations do not necessarily have to be seen as isolated from classical computations. The term *hybrid quantum-classical* is used for algorithms, where quantum and classical computations are performed together, typically in an iterative fashion. The QC can then be used to solve only the specific tasks for which it is suitable. This makes hybrid quantum-classical algorithms particularly promising for near-term applications on NISQ devices. We present an example in Section 3.3. There is evidence that hybrid quantum-classical algorithms can also enable a polynomial runtime improvement. For example, Creemers and Armas (2025) focus on applying Grover's search to achieve a quadratic speedup in the context of algorithms such as hybrid quantum-classical branch-and-bound.

From a broader methodological perspective, one can identify several paradigms for designing potentially promising quantum algorithms. Their detailed description is beyond the scope of this paper, but some high-level discussion is given by Nielsen and Chuang (2010), and an in-depth overview in the context of discrete optimization is presented by Abbas et al. (2024). We take another approach here and focus on a specific example of a key idea from physics in Section 2.4, which motivates a group of actively developed quantum optimization methods discussed in more detail further.

2.4. Quantum adiabatic algorithm

We start with the outline of a central quantum computational strategy that serves as an umbrella concept for the three quantum-powered optimization approaches we consider in this paper. This strategy, the so-called QAA (Albash & Lidar, 2018; Farhi et al., 2000), is based on the principle of solving optimization problems by leveraging the natural evolution of quantum systems. QAA founds on the computational model of AQC, a form of analog quantum computing.³

³ While adiabatic and digital quantum computing are often presented as distinct, they are theoretically equivalent up to polynomial overheads. This means that each algorithm in one model can be translated into the other (Albash & Lidar, 2018). In practice, some quantum computations such as QAA are more naturally expressed in the adiabatic framework, whereas others align better with the digital framework.

² We do not use the term *logical qubit* as it is commonly used in the context of FTQC (Zhao et al., 2022).

The QAA relies on the *adiabatic theorem* of quantum physics, which, in simplified terms, guarantees that if a quantum system is in its ground state (i.e., the minimum energy state) and its “energy landscape” (comprised of all possible energy states including the ground state) is changed “slowly enough”, it will remain in the ground state. In other words, the system may slowly “evolve” from one ground state to another ground state in a different energy landscape. The necessary pace of the changes in the system depends on the size of the so-called *minimum energy gap*, the smallest energy difference between the ground state and the second lowest energy state throughout the entire evolution process.⁴ The evolution of a quantum state that complies with the adiabatic theorem is called *adiabatic evolution*.

The energy landscape of a multi-qubit quantum system can be used to encode information. The ground state, as the lowest-energy configuration within this landscape, therefore carries information as well.⁵ At the core of QAA lies the ability to store and extract information from the ground state of a quantum system by carefully shaping and controlling its energy landscape, taking into account the adiabatic theorem. The conceptual idea can (from a hardware-agnostic perspective) be outlined as follows: First, the multi-qubit system of the QC is initialized in a known and easy-to-prepare ground state. By design, this is a generic ground state, which is in particular independent of the optimization problem of interest. Then, over time, the system parameters (which are hardware-specific) are slowly tuned to achieve a controlled adiabatic evolution. The goal of this controlled evolution is to gradually shape an energy landscape that is a one-to-one representation of the optimization landscape of the problem of interest. This means that each candidate solution of the optimization problem can be associated with a quantum state and the corresponding objective value with an energy. After a successful adiabatic evolution, the system is by definition still in the ground state, but this new ground state now represents an optimal solution to the optimization problem. Hence, by measuring this state, the solution can be found. The surprising insight is that it is not necessary to know the optimal solution to be able to prepare the final ground state that contains this information. This is only possible because quantum physics itself is used to discover the solution in the sense of an analog computation.

The QAA is inherently different from classical methods because it can benefit from quantum features such as superposition and tunneling, which allows the system to explore many possible solutions “simultaneously” and to “escape” from local energy minima.⁶ As a consequence, the QAA has the potential to outperform classical methods. However, this presumption has not been demonstrated for problems of practical scale yet, and is the subject of ongoing research (Abbas et al., 2024).

Indeed, there are three significant challenges for practical applications. First, a technical requirement for an optimization problem to be solvable with this approach is that the energy landscape of the quantum system must be tuneable in such a way that it becomes a one-to-one representation of the corresponding optimization landscape of the problem. In effect, this means that only hardware-specific (or “natural”) problem classes can be solved on every quantum device. In order to solve a problem from a different problem class, it must first be mapped onto the natural problem class through suitable modeling.

⁴ The adiabatic theorem requires that the evolution runtime must be large on the timescale set by $(1/\Delta^2)$, where Δ denotes the minimum energy gap (Albash & Lidar, 2018). In practice, this means that Δ must also be non-zero, which means that energy levels must not cross (Zhang et al., 2014). Otherwise, achieving an adiabatic transition is fundamentally impossible.

⁵ It is thus not surprising that a measurement of the ground state does not necessarily yield a trivial all-zero bitstring, as such an outcome would imply no information gain.

⁶ The role of entanglement in the performance of QAA and even how best to characterize and measure it in this context remains an open area of research topic (Albash & Lidar, 2018).

If this is not possible, the problem cannot be solved with the device at hand.

Second, the minimum energy gap is usually undeterminable in practice since determining it requires knowledge of the ground state energy, which is the optimal solution to the optimization problem to be found by the computation. Therefore, an estimated minimum energy gap (e.g., based on empirical results) is typically used instead to determine the timescale of the evolution process. However, this does not necessarily ensure an adiabatic behavior.

Third, there is no guarantee that the complex quantum dynamics behind the analog computation reveal the desired result in all cases. Typically, the final state of the quantum system is a superposition state such that repeating the entire process might yield different measurement results each time, corresponding to new solution candidates. In an idealized scenario, these solution candidates would all be optimal solutions to the underlying problem (each with the same minimal energy). In practice, however, quantum fluctuations, non-optimal tunneling, an insufficiently slow evolution speed (due to an unfavorable estimate for the minimum energy gap), and hardware-related uncertainties of NISQ devices lead to a final state that is in a superposition of sub-optimal (and potentially optimal) solutions. In other words, measuring the final state may reveal sub-optimal solutions, making the QAA effectively a heuristic. It is therefore common to collect data from multiple algorithm runs, usually referred to as *shots*, to choose the best outcome.

3. Quantum optimization workflows

In this paper, we examine three different quantum-powered optimization approaches, all of which are based on the conceptual idea of AQC and, more specifically, the QAA from Section 2.4. They differ not only in the chosen hardware and algorithmic realization, but also in their applicability to specific problem classes. Definitions for these problem classes are provided in Section 4. Our goal is to illustrate how different quantum computing strategies can be used for practical OR applications. All of the algorithms we consider here are well-known in the literature, we particularly do not invent new methods to achieve better results, but focus on the performance of existing methods. We consider the three following approaches (see also Supplement F for more details).

1. **NA-OPT** (Section 3.1): We use a quantum algorithm that can be understood as a realization of a QAA close to the original concept of analog adiabatic evolution. As quantum hardware, we use *Aquila* from *QuEra* (Wurtz et al., 2023), an analog device operating with trapped neutral atoms. It can be accessed commercially online via the *AWS Braket* cloud service (Amazon Web Services, 2020). The task is to solve UD-MIS, the natural problem class of the *Aquila* device.
2. **QA-OPT** (Section 3.2): We use a quantum algorithm known as QA. In simple terms, it can be understood as a practice-oriented variant of a QAA that may potentially involve a non-adiabatic evolution. As quantum hardware, we use the *D-Wave Advantage* quantum annealer (McGeoch & Farré, 2020), an analog device operating with superconducting qubits. It is commercially accessible online via the *Leap* cloud service (D-Wave, 2024). The task is to solve QUBOs, the natural problem class of the *Advantage* device.
3. **QAOA-OPT** (Section 3.3): We use a hybrid quantum-classical algorithm known as QAOA. In simple terms, it can be understood as a digitized approximation of a QAA. As quantum hardware, we use two gate-based devices from *IBM*, *ibm_cusco* and *ibm_nazca*, which operate with superconducting qubits. *IBM* devices are commercially accessible online via the *IBM Quantum* cloud service (IBM Quantum, 2023a). The task is to solve QUBOs, although variants of QAOA are suitable for different problem classes.

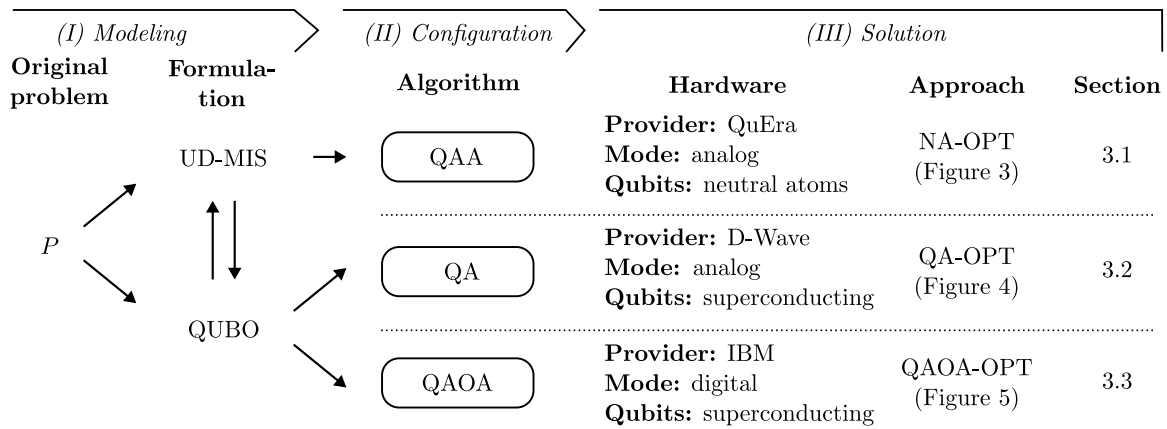


Fig. 2. Unified workflow for the three quantum-powered optimization approaches NA-OPT, QA-OPT, and QAOA-OPT that consists of three high-level steps: (I) Modeling, (II) Configuration, and (III) Solution. All three approaches utilize QAA-inspired techniques.

All three approaches share a unified workflow in three high-level steps, as outlined in Fig. 2:

- (I) The given problem P first has to be *modeled* in form of a suitable problem class.
- (II) This problem formulation is then used to specify the *configuration* of the algorithm and hardware.
- (III) Finally, the actual *solution process* takes place, usually involving a series of quantum hardware computations and additional classical computations, (e.g., pre- and post-processing).

In the following, we provide a brief outline of the three approaches in the context of these three high-level steps. We limit ourselves to the description of the fundamental principles and provide further references for more details.

3.1. NA-OPT: solving UD-MIS with an analog QC (QuEra)

Our first approach, NA-OPT, makes use of the analog device *Aquila* from QuEra (Wurtz et al., 2023) to solve UD-MIS (da Silva Coelho et al., 2022; Ebadi et al., 2022, 2021; Pichler et al., 2018) by exploiting the so-called *Rydberg blockade effect*, a physical law originating from quantum physics. A recent review of the technology is given by Wintersperger et al. (2023).

Hardware. The *Aquila* device operates with an array of rubidium atoms, which are trapped in a vacuum cell by lasers and can be arranged on a two-dimensional plane using optical tweezers. The atoms realize so-called *Rydberg qubits*, which possess two energy states: the non-Rydberg state and the Rydberg state. The Rydberg blockade effect makes it energetically favorable that atoms within a certain distance from each other, the so-called *Rydberg blockade radius*, are not simultaneously in a Rydberg state. This effect can be used to solve UD-MIS, as explained in the following.

Algorithm. We use a realization of a QAA and follow the unified workflow from Section 3. A device-specific scheme is summarized in Fig. 3. For practical reasons, the *Aquila* device can only hold up to 256 atoms in the array at once, which translates to at most 256 nodes in the UD-MIS instance. Moreover, the geometric configurations of the atoms array, and hence possible UD-MIS graphs, are restricted to a square two-dimensional lattice that also enforces a minimum distance between all nodes. If the geometry of a graph from a UD-MIS instance does not comply to these requirements, it has to be transformed first in the configuration step, which may or may not be possible depending on the instance.

To get an intuitive understanding of how the adiabatic algorithm works, we first revisit the structure of unit disk graphs. Presuming

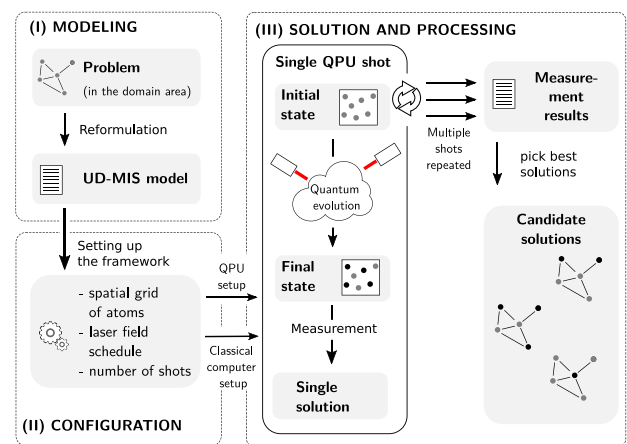


Fig. 3. Workflow of the NA-OPT approach: solving UD-MIS with QAA.

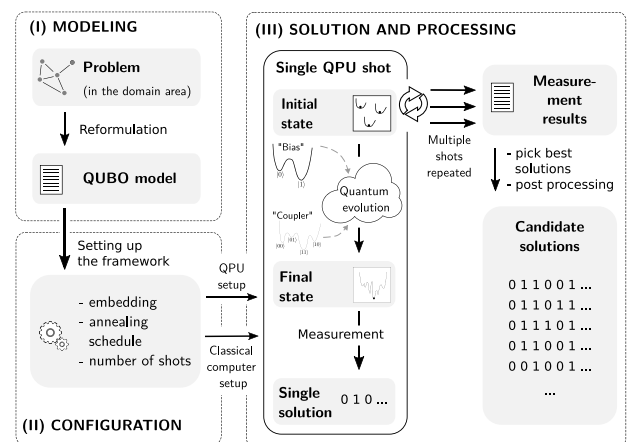


Fig. 4. Workflow of the QA-OPT approach: solving QUBOs with QA.

nodes that are localized on an Euclidean plane, a unit disk graph has an edge between two vertices if and only if the distance between them is at most one. In other words, the relative location of nodes determines their connectivity. Therefore, provided the unit disk graph of a UD-MIS instance, we can associate an atom with each node and arrange the atoms on the two-dimensional plane according to the corresponding (scaled) node coordinates to achieve a one-to-one spatial correspondence between graph nodes and atoms. The Rydberg blockade radius of

the system is tuned to match the unit disk radius such that atoms that correspond to connected nodes lie within the Rydberg blockade radius.

With this problem encoding, a QAA can be realized by evolving the system in a controlled fashion with an external laser field. To this end, we start from a problem-agnostic energy landscape, where the ground state corresponds to a quantum system in which all atoms are in a non-Rydberg state. Slowly, the controls are adjusted to arrive at a problem-specific energy landscape while the positions of the atoms remain unchanged. The problem-specific ground state corresponds to a quantum system in which as many atoms as possible are in a Rydberg state (which might be ambiguous). The Rydberg blockade effect—implemented by the atom locations—is intended to ensure the constraints of the maximum independent set problem (MIS), which require that neighboring (i.e., connected) nodes cannot both be part of the resulting node set. As a consequence, the ground state of the quantum system encodes a UD-MIS solution: the atoms in the Rydberg state indicate the resulting node set.

Serret et al. (2020) estimate the necessary size of a neutral atom array to yield an advantage over classical approximation algorithms for UD-MIS, and Wurtz et al. (2022) provide an overview of the connections between the MIS and many OR problems in this context. Note that neutral atom arrays can in principle also be used to solve other problem classes as well, which is an active field of research (Nguyen et al., 2023). For instance, a conceptional idea for a non-unit disk (non-blockade-based) framework was suggested by Goswami et al. (2024). Using a hardware feature called local detuning, *Aquila* can also be used to solve the maximum-weight independent set problem on a unit disk graph (UD-MWIS) in complete analogy to UD-MIS.⁷

Number of qubits needed. The number of logical qubits necessary to solve a UD-MIS instance is equal to the number of graph nodes, which also directly translates to the number of physical qubits. In addition to the number of nodes, a UD-MIS graph must satisfy certain constraints to be encodable as an atom array on *Aquila*. These include feasible placement of the corresponding atoms on a grid (with a size of around 16×16) and a unit disk size within a fixed range. Detailed hardware requirements are provided in Wurtz et al. (2023). We refer to any UD-MIS instance that meets these criteria as a *hardware-compliant* UD-MIS instance. In practice, choosing a suitable geometric arrangement of the atoms for an accurate problem representation might pose additional challenges. While this issue was never binding in our numerical experiments, for more connected problem instances it might become a problem. In a sense, a neutral-atoms-based QPU can provide qubit *connectivity* only up to a certain limit. Investigating this effect might constitute a potential direction for further research.

While the same UD-MIS graph can be represented by different atom geometries, there are engineering constraints on the distances between the atoms and physical implications for each specific atomic configuration. For instance, positioning atoms approximately within the blockade radius of other atoms (not too close and not too far) might result in undesirable outcomes, due to the underlying physics of the protocol. Some best practices and discussion of the details of the underlying technology can be found in Wurtz et al. (2023).

As the decision version of the UD-MIS problem is $\mathcal{NP}$ -complete (Clark et al., 1990), an arbitrary problem from $\mathcal{NP}$ can be reduced to it with a polynomial overhead in the instance size. While the reduction following from the general proof goes via non-deterministic Turing machines, more direct reductions can exist for specific problems. We exemplarily use the results from Nguyen et al. (2023), who proposed a transformation that maps a QUBO instance with N variables to an equivalent maximum-weight independent set problem (MWIS) instance over a unit disk graph with at most $4N^2$ nodes, as summarized in Lemma 1.

⁷ At the time of performing our hardware experiments, this feature was not fully supported yet and UD-MWIS is therefore not considered in our numerical evaluations.

Lemma 1 (Nguyen et al., 2023, Section V.C). A hardware-compliant UD-MIS instance with N nodes can be solved on a neutral-atom-based QC with N qubits. An arbitrary QUBO instance with N binary variables can be encoded as a UD-MWIS instance with at most $4N^2$ nodes, and hence, assuming the geometric hardware constraints are met,⁸ requires a neutral-atom-based QC with $4N^2$ physical qubits.

3.2. QA-OPT: solving QUBOs with a quantum annealer (D-Wave)

Our second approach, QA-OPT, makes use of the *Advantage* quantum annealer from D-Wave (McGeoch & Farré, 2020) to solve quadratic unconstrained binary optimization problems (QUBOs). We review the specific formulation later, in Section 4. *Advantage* is a special-purpose device designed for QA (Das & Chakrabarti, 2005; Finnila et al., 1994; Hauke et al., 2020; Morita & Nishimori, 2008).

Hardware. The *Advantage* device is based on superconducting technology in which superconducting loops are used to physically realize qubits. The quantum system can be controlled via so-called *couplers* and *biases*. The bias of each qubit is a control which allows making it more energetically favorable for this qubit to end up in a specific state (either the ground, or the excited state). A coupler controls the interaction between two qubits. It allows increasing or decrease the energy contribution of qubit correlations. For technical reasons, couplers are only implemented between physically adjacent qubits, which means that there is a limit to the interactions that can be realized. We will revisit this topic further below.

Algorithm. Since NISQ hardware is often too noisy to support sufficiently slow and stable adiabatic evolution and the minimum energy gap required for adiabaticity is usually unknown, QA can be considered as a more practical variant of QAA, which does not strictly rely on adiabatic evolution. It is based on the same premise as QAA by starting from the ground state of a problem-agnostic energy landscape and ending in the ground state of a problem-specific landscape. However, it is usually implemented under conditions where adiabaticity may not be preserved due to noise or the finite runtime of the hardware. Consequently, it is clearly a heuristic method that relies on sufficiently many shots to produce reliable results.

Here, we use a realization of a QA and follow the unified workflow from Section 3. A device-specific scheme is summarized in Fig. 4. Specifically, to solve a given QUBO instance, we first encode the problem on the device by associating each binary optimization variable with a qubit. The initial energy landscape is chosen in a problem-agnostic way such that its ground state is a balanced superposition state of every qubit, for which an immediate measurement would return a uniformly random candidate solution. Then, using a predefined annealing schedule, the system is driven to a problem-specific energy landscape with a ground state that represents a candidate solution. To this end, the biases and couplers are tuned to encode the QUBO coefficients. Finally, a measurement reveals a solution candidate. This process is typically repeated many times to generate an ensemble of solutions.

The biases and couplers of the *Advantage* device have a limited effective resolution to encode the QUBO coefficients, which means that a QUBO can only be expected to be solved efficiently if its coefficients have a limited dynamic range (ratio between largest and smallest

⁸ Generally speaking, not any UD-MWIS instance will be hardware-compliant. This comes from the fact that the grid necessary to place the $4N^2$ atoms might in principle require more space than that available due to the hardware constraints stemming from the optical properties of the hardware. For example, a dense QUBO instance with 5 variables requires 100 atoms on a graph that spans almost the full 16×16 grid of *Aquila* (Nguyen et al., 2023). However, in our dataset, such constraints were never binding, and it was the number of qubits that constituted a limiting factor to our instance sizes.

values; see Mücke et al., 2025). Furthermore, the *Advantage* device has 5760 qubits, which represents the largest possible QUBO instance that can be encoded. However, choosing a one-to-one correspondence between optimization variables and qubits for the encoding, as described above, is in fact not possible for all instances. Due to the limited availability of couplers, the effective number of required qubits can be much larger than the number of variables. This is an important practical issue, which is explained in more detail in the following.

Number of qubits needed. For a given QUBO instance, the number of logical qubits corresponds to the number of binary optimization variables. The logical qubits need to be encoded with the physical qubits of the quantum annealer, which means that each logical qubit has to be assigned to a physical qubit and each correlation between logical qubits has to be assigned to a coupler. As already mentioned above, the *Advantage* device only supports couplers between specific physical qubits (in which case they are called *connected*). Therefore, a straightforward encoding of a sufficiently large and dense QUBO instance may in fact require non-existing couplers. To overcome this limitation, a logical qubit can also be encoded as a *chain* comprising several physical qubits (Venegas-Andraca et al., 2018) that are strongly coupled with each other. The encoding task then corresponds to the graph-theoretic problem of finding a *minor embedding* (Choi, 2008) of the *problem graph* (where nodes are QUBO variables and edges correspond to non-zero QUBO coefficients) into the *topology graph* of the device (where nodes are physical qubits and edges exist for every coupler). In this context, the chains are usually called the *branch sets* associated to the nodes of the problem graph. If possible, chains are to be avoided because *chain breaks* are an important type of errors occurring in quantum annealers, where physical qubits representing the same logical qubit attain different states. Some *post-processing* techniques to recover consistent solutions have been proposed (Pelofske et al., 2020). Gilbert et al. (2024) conducted experiments on crafted QUBO instances whose problem graphs are subgraphs of the hardware topology, allowing for embedding without additional overhead.

The *Advantage* device is designed according to the so-called *Pegasus* topology (Boothby et al., 2020; Dattani et al., 2019), and the aforementioned works imply a polynomial-time algorithm for embedding a complete graph into this topology graph. We extend this result with the following lemma, formulated in terms of the number of physical qubits required for a given QUBO instance. It is a direct consequence of the formal description of the topology by Boothby et al. (2020), see Supplement H for further details.

Lemma 2. *The problem graph G_Q of a QUBO instance with N variables can be embedded into a Pegasus graph G_T with*

$$N_{QA}(N) := 24 \left\lceil \frac{N+10}{12} \right\rceil \left\lceil \frac{N-2}{12} \right\rceil \leq \frac{(N+21)(N+9)}{6}$$

nodes such that the corresponding QUBO instance can be solved on a quantum annealer with Pegasus topology using $N_{QA}(N)$ physical qubits. If G_Q is a subgraph of the Pegasus graph, N qubits suffice.

In practice, certain physical qubits of a quantum annealer might turn out to be “broken” as a result of an imperfect construction process, which means that these qubits cannot be used for computations. This is also the case for the *Advantage* device. As a consequence, the effective topology reduces to a subgraph of the Pegasus topology graph, in which the nodes corresponding to broken qubits are removed. Deciding whether a given problem graph is a minor of such an irregular topology graph is $\mathcal{NP}$ -complete (Lobe & Lutz, 2024), and the embedding problem is usually solved heuristically (Cai et al., 2014; Zbinden et al., 2020). Such heuristics typically aim to find embeddings that use few physical qubits while still maintaining some redundancy to strengthen the coupling of the chains and reduce the probability of errors (Pelofske, 2024). Some research works study the probability that a random problem graph can be successfully embedded into a fixed

topology graph (corresponding to a fixed hardware configuration). This is known as the *embedding probability* (Sugie et al., 2021). For further information and recent research results, we refer to Gomez-Tejedor et al. (2025), Sinno et al. (2025), Sugie et al. (2021) and Zbinden et al. (2020).

3.3. QAOA-OPT: solving QUBOs with gate-based QCs (IBM)

Our third approach, QAOA-OPT, makes use of the gate-based QCs from IBM to solve QUBOs with QAOA (Blekos et al., 2024; Farhi et al., 2014; Grange et al., 2023). For our numerical experiments, we have made use of two of the 127-qubit devices *ibm_cusco* and *ibm_nazca*.

Hardware. The IBM devices are based on superconducting technology. As universal gate-based QCs, they can in principle run any gate-based quantum algorithm, which can be seen as a three-step process. First, all qubits are prepared in the ground state. Then, a sequence of gates is applied, each corresponding to a discrete control that alters the quantum state of the multi-qubit system. Finally, a measurement of the resulting state yields an outcome. A shot involves performing all these steps to obtain a single bitstring. A sequence of gates acting on a set of qubits is referred to as a *quantum circuit* and provides all necessary instructions to operate a gate-based QC. In other words, a quantum circuit controls the evolution from an initial quantum state to a final one, which is then measured.

Algorithm. We consider QAOA to solve QUBOs,⁹ which is a hybrid quantum–classical algorithm that can be understood as an attempt to digitize adiabatic evolution in the sense of a *digital QAA*. As a special implementation of a variational quantum algorithm (VQA) (Blekos et al., 2024; Cerezo et al., 2021; Grange et al., 2023), the key concept of QAOA is that of a *parameterized circuit*, a quantum circuit with gates that depend on real-valued parameters. Depending on the choice of parameters, executing a parameterized circuit on a QC will provide different measurement outcomes. The goal of QAOA is to iteratively adjust the circuit parameters to guide the quantum system towards states that represent optimal or near-optimal solutions of the underlying optimization problem. This is realized along the lines of a typical QAA by starting from a problem-agnostic ground state, a balanced superposition state of every qubit, and then applying a sequence of (parameterized) gates to arrive at the problem-specific state that encodes the solution. In contrast to an analog QAA, QAOA only mimics the adiabatic transition (which may or may not yield an approximation) using a discrete sequence of gates instead of a continuous control. We follow the unified workflow from Section 3, where a device-specific scheme is summarized in Fig. 5. Note that in the solution step, we perform not a single series of shots as in the two previous approaches, but a hybrid quantum–classical optimization loop.

In the following, we explain QAOA in more detail. For a given QUBO instance, each qubit encodes one optimization variable. The quantum circuit is then chosen in such a way that it represents a parameterized approximation of the annealing process. The specific circuit choice is also known as *ansatz*, its *depth* is a fixed parameter that denotes the number of repeating circuit elements (i.e., single gates or certain sequences of gates) in the *ansatz*. On the one hand, choosing a more complex *ansatz* (e.g., with greater depth) enables the circuit to explore more possibilities to mimic the adiabatic schedule. On the other hand, this exploration also requires tuning more parameters, which becomes computationally challenging and typically requires much more efforts to converge to a solution. These two opposing factors must be weighed against each other by the user to select a suitable *ansatz*.

⁹ In addition to solving QUBOs, variants of QAOA can also take into account constraints (Bucher et al., 2025; Fuchs & Bassa, 2024; Fuchs et al., 2022; Hadfield, 2018; Hadfield et al., 2019). Furthermore, QAOA can also be used to solve polynomial unconstrained binary optimization problems (PUBOs) (Grange et al., 2023).

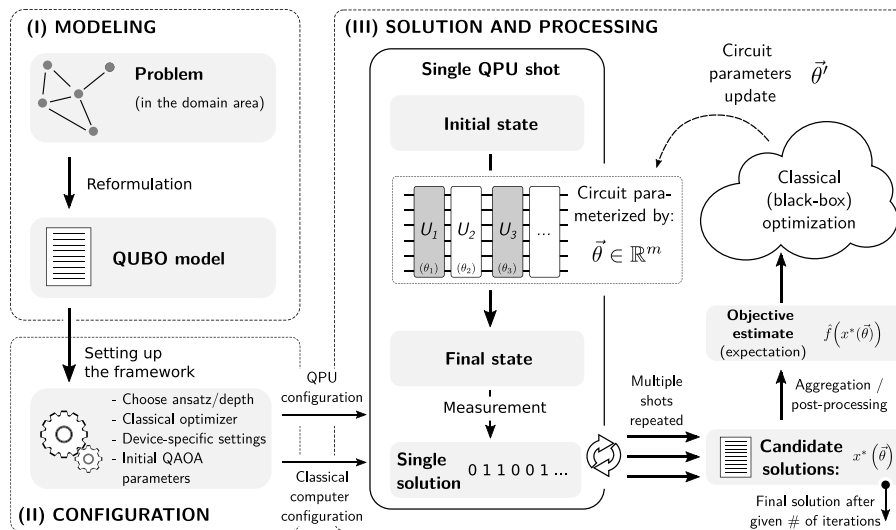


Fig. 5. Workflow of the QAOA-OPT approach: solving QUBOs with QAOA.

The role of the QC in the hybrid QAOA setup is to run parameterized circuits and provide the measurement results, which can then be used to estimate the expectation value of the system's energy. The classical computer, on the other hand, optimizes the circuit parameters based on the quantum measurements and proposes updated parameters that aim to reduce the expected energy.

Summarized, the task of the classical computer is to effectively perform a black-box optimization under uncertainties. The objective function to be minimized is the expected energy of the quantum system and the optimization variables are the real-valued circuit parameters. In principle, this problem can be solved with any suitable classical optimization algorithm. (E.g., gradient descent methods, simultaneous perturbation stochastic approximation (SPSA), and so on. See Section D by Cerezo et al. (2021) for a compact overview.) It is a black-box optimization because the problem structure is effectively unknown. The classical optimizer can query the QC to provide the expected energy for a given choice of circuit parameters. Gradients of the objective function can be obtained using finite differences or specialized techniques (Crooks, 2019). Uncertainties arise because of two reasons: first, the expected energy is only estimated with finite samples (each representing a series of shots) and, second, NISQ devices suffer from hardware-related uncertainties. Since evaluations on a QC are costly, it is typically desired to find sufficiently good circuit parameters with as few iterations (or shots) as possible. Once the optimal parameters have been found, the QC can be used to sample solution candidates.

For our QAOA-OPT approach, we only consider a foundational implementation of QAOA to explore the basic performance. However, from a practical perspective, fine-tuning the algorithm is an important but challenging aspect. For example, there are separate works on circuit parameter optimization (Zhou et al., 2020) and initial parameter values (Sack et al., 2023; Sack & Serbyn, 2021). Improvements to the original algorithm include extensions of the aggregation function (Barkoutsos et al., 2020), counterdiabatic driving (Chandarana et al., 2022), and warmstarts (Egger et al., 2021).

In addition, there are two general technical challenges when running quantum circuits on IBM devices that also apply to all QC evaluations within QAOA:

1. Limited gates: Only specific one-qubit and two-qubit gates, the so-called *basis gates*, can be executed. The set of basis gates is universal in the sense that any other gate can be decomposed into a sequence of basis gates.
2. Limited connectivity: Two-qubit gates can only be executed for specific pairs of physical qubits according to the prescribed

hardware connectivity. These gaps can be bridged by the use of additional gates.

To overcome these obstacles, a classical preprocessing step called *transpilation* is necessary to transform a given quantum circuit into a hardware-compliant quantum circuit. In fact, there are infinitely many hardware-compliant quantum circuits that lead to the same measurement outcomes on an idealized (i.e., noise-free) QC. On actual IBM devices, however, some mathematically equivalent circuits will perform better than others. For example, circuits with fewer basis gates typically perform better. Moreover, each physical qubit exhibits an individual level of noise, which might also change over time (Baheri et al., 2022). Therefore, the practical performance of a quantum circuit may also depend on which physical qubits are used for its implementation. Thus, the effective goal of the transpilation is to find a hardware-compliant and well-performing representation of a given quantum circuit (Hua et al., 2023; Karuppasamy et al., 2025; Li et al., 2019; Nation & Treinish, 2023; Waring et al., 2024; Wilson et al., 2020). Essentially a classical optimization problem, transpilation within QAOA-OPT is solved in the IBM software package *Qiskit* (IBM Quantum, 2023b; Javadi-Abhari et al., 2024) with multi-step heuristics that include mapping logical qubits to physical qubits, decomposing all gates into basis gates, routing qubits with additional gates according to hardware connectivity, and an overall circuit optimization to improve the performance. Note that similarly to the previous two approaches, qubit connectivity manifests as a separate issue here.

Number of qubits needed. In analogy to QA-OPT, the number of logical qubits for QAOA-OPT corresponds to the number of binary optimization variables for a given QUBO instance, which can then also be translated one-to-one into physical qubits, as summarized in Remark 1.

Remark 1. An arbitrary QUBO instance with N variables that is solved with QAOA requires N physical qubits. While transpilation might affect the output quality, it does not necessarily require additional physical qubits.

4. Application to selected problems

In this section, we focus on three classes of optimization problems: UD-MIS, the MaxCut, and the TSP. All three are graph-based problems, meaning that each instance is defined on an undirected graph $G := (V, E)$, where V is the set of nodes and $E \subseteq \binom{V}{2}$ is the set of edges (with no duplicates or self-loops). For each class, we present a formal problem definition as a QUBO and study the required number of physical qubits

it takes to solve them with the three quantum-powered optimization approaches from Fig. 2. Solving the same QUBO instance may require a different number of physical qubits for each approach, depending on the problem structure and device implementation details.

We emphasize that we do not present novel ways of modeling the problems, nor do we provide a comprehensive overview of possible quantum-aware modeling approaches for optimization problems. In fact, finding suitable formulations of classical optimization problems that allow an effective treatment with quantum optimization is a task known as *problem encoding* and constitutes a research direction of its own. For more general discussions of different formulations for a wide range of optimization problems in the context of quantum computing and QUBOs, see for example Dominguez et al. (2023), Glover et al. (2022) and Lucas (2014). Furthermore, Codognet (2024), Gonzalez-Bermejo et al. (2022), Ruan et al. (2020), and Salehi et al. (2022) focus on TSP, while Hadfield (2021) considers more general classes of functions and aims to provide a “design toolkit of quantum optimization”. Hadfield et al. (2017) discuss encoding constraints without penalty terms, specifically in the context of QAOA. Padmasola et al. (2025) focus on TSP, but compare and contrast a few other quantum technologies. Possible problem encoding strategies are diverse, and we limit ourselves here to common formulations suitable in the context of the considered quantum-powered optimization approaches.

We start by reviewing QUBOs. An instance is defined by a symmetric matrix $Q \in \mathbb{R}^{N \times N}$ as:

$$\min_x x^T Q x, \quad x \in \{0, 1\}^N \quad (1)$$

with N binary decision variables $x := (x_1, \dots, x_N)$. Note that the requirement to represent problems in this form is in fact not too restrictive. Lucas (2014) discusses so-called *Ising formulations*, which are very close to QUBO, for many combinatorial optimization problems, including the 21 $\mathcal{NP}$ -hard problems from Karp's (1975) list. A tutorial by Glover et al. (2022) focuses on formulating combinatorial optimization problems as QUBOs.

4.1. UD-MIS

Given an undirected graph $G := (V, E)$, the MIS problem consists of finding a subset of nodes of maximum cardinality such that no two nodes in the subset are adjacent. The problem can be modeled as an integer linear program (ILP) using one binary variable x_i for each node $i \in V$, indicating whether it constitutes a part of the solution:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i \\ \text{s. t.} \quad & x_i + x_j \leq 1 \quad \text{for all } \{i, j\} \in E, \\ & x_i \in \{0, 1\} \quad \text{for all } i \in V. \end{aligned} \quad (2)$$

If the graph G is a unit disk graph, the problem is also called UD-MIS. In a unit disk graph, each node corresponds to a point in the plane, and an edge exists between two nodes if and only if the Euclidean distance between their corresponding points is at most one. In general, given a graph from an arbitrary MIS instance, it is $\mathcal{NP}$ -hard to decide whether it can be represented as a unit disk graph or not (Breu & Kirkpatrick, 1998). Heuristics such as a force-based approach (da Silva Coelho et al., 2022) can be used to find a unit disk representation. Alternatively, MIS on an arbitrarily connected graph can also be transformed into a UD-MWIS by increasing the graph size (Bombieri et al., 2025; Nguyen et al., 2023). The MWIS problem is a generalization of MIS, which takes into account a non-negative weight $w_i \in \mathbb{R}_{\geq 0}$ for each node $i \in V$. A MWIS instance is therefore defined by a weighted graph $G := (V, E, w)$ and reduces to a MIS instance for units weights.¹⁰ In analogy

¹⁰ As before, if the graph G is a unit disk graph, the problem is called UD-MWIS.

to formulation (2), the MWIS problem can be formulated as:

$$\begin{aligned} \max \quad & \sum_{i \in V} w_i x_i \\ \text{s. t.} \quad & x_i + x_j \leq 1 \quad \text{for all } \{i, j\} \in E, \\ & x_i \in \{0, 1\} \quad \text{for all } i \in V. \end{aligned} \quad (3)$$

In the following, we only consider the unweighted problem class. To obtain an equivalent QUBO for formulation (2), we can introduce constraints as quadratic penalty terms corresponding to all edges $\{i, j\} \in E$ and switch to a minimization, which results in:

$$\begin{aligned} \max_{x \in \{0, 1\}^{|V|}} \quad & \sum_{i \in V} x_i - M \sum_{\{i, j\} \in E} x_i x_j = - \min_{x \in \{0, 1\}^{|V|}} x^T Q x, \quad \text{where} \\ Q_{ij} = \begin{cases} -1 & \text{if } i = j, \\ M/2 & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (4)$$

In the definition of Q_{ij} , we consider all ordered pairs i, j to obtain a symmetric matrix. This formulation is equivalent to (2) for large enough M (e.g., $M = |V| + 1$, see Supplement B). This yields the necessary number of binary variables, as stated in the following lemma.

Lemma 3. For a graph $G = (V, E)$, the UD-MIS instance given by formulation (2) can be solved on a neutral-atom-based machine with $N := |V|$ physical qubits. The respective QUBO formulation (4) requires N logical qubits. Therefore, it can be solved on a quantum annealer with Pegasus topology using at most $24 \lceil \frac{N+10}{12} \rceil \lceil \frac{N-2}{12} \rceil$ physical qubits or on a general gate-based QC with N physical qubits.

Proof. Follows from Lemmas 1 and 2 and Remark 1. $\square$

4.2. MaxCut

Given an undirected graph $G = (V, E, w)$, where $w_{ij} = w_{ji} \in \mathbb{R}_{\geq 0}$ denotes the weight of the edge $\{i, j\} \in E$, the MaxCut problem consists of finding a partition of the node set V into two disjoint subsets that maximizes the total weight of the edges connecting nodes in different subsets. This can be reformulated as the following ILP: follows.

$$\max \quad \sum_{\{i, j\} \in E} e_{ij} w_{ij}, \quad (5)$$

$$\text{s. t.} \quad e_{ij} \leq x_i + x_j \quad \text{for all } \{i, j\} \in E, \quad (6)$$

$$e_{ij} \leq 2 - (x_i + x_j) \quad \text{for all } \{i, j\} \in E, \quad (7)$$

$$x_i, e_{ij} \in \{0, 1\} \quad \text{for all } \{i, j\} \in E.$$

Here, variables x_j denote which set in the partition vertex j belongs to, and the constraints (6) and (7) ensure that in an optimal solution, the variable $e_{ij} = 1$ if and only if $\{i, j\}$ is in the cut (i.e., exactly one of x_i and x_j equals 1), and 0 otherwise. We assume that $i < j$ for all double-indexed variables e_{ij} .

Based on the formulation given, e.g., in the textbook chapter of Laurent (1997), the problem (5) can be posed as a QUBO:

$$\begin{aligned} \max_{x \in \{0, 1\}^n} \quad & \sum_{\{i, j\} \in E} w_{ij} (x_i + x_j - 2x_i x_j) = - \min_{x \in \{0, 1\}^n} x^T Q x, \quad \text{where} \\ Q_{ij} = \begin{cases} \sum_{k: \{i, k\} \in E} (-w_{ik}) & \text{if } i = j, \\ w_{ij} & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (8)$$

Whenever $x_i = 1$ and $x_j = 0$ (or vice versa), the term $w_{ij}(x_i + x_j - 2x_i x_j)$ contributes w_{ij} to the objective. Otherwise, if $x_i = x_j$, the term's contribution will be zero.

Lemma 4. For a graph $G = (V, E, w)$, the QUBO formulation of the MaxCut problem (8), requires $N := |V|$ logical qubits. Therefore, it can be solved on a quantum annealer with Pegasus topology using at most $24 \lceil \frac{N+10}{12} \rceil \lceil \frac{N-2}{12} \rceil$ physical qubits, on a neutral-atom-based device using $4N^2$ physical qubits, or on a general gate-based device using N physical qubits.

Proof. Follows from Lemmas 1 and 2, and Remark 1. $\square$

Note that for MaxCut, the number of binary variables of the QUBO is only linear in the number of nodes, while for the ILP it may be quadratic, making this problem particularly suitable for QCs.

4.3. TSP

Given a complete undirected graph $G = (V, E, w)$, where $w_{ij} = w_{ji} \in \mathbb{R}_{\geq 0}$ denotes the weight of the edge $\{i, j\} \in E$ and $N := |V|$ the number of nodes, the TSP seeks for a shortest possible route in terms of the cumulative edge weight that visits each vertex exactly once and returns to the start node. On classical computers, the TSP is typically solved by sophisticated ILP techniques, combined with heuristics to find a good initial solution (Applegate et al., 2006). Different ILP formulations have been developed for this problem, which makes it convenient to use it as an example to compare and contrast a few alternative formulations in the context of quantum computing.

Dantzig–Fulkerson–Johnson formulation. The classical formulation (Dantzig et al., 1954) used in most OR textbooks implies binary variables x_{ij} , $i < j$, for all edges $\{i, j\} \in E$, indicating that an edge is traversed by the tour:

$$\min \sum_{i < j} c_{ij} x_{ij}, \quad (\text{DFJ})$$

$$\text{s. t. } \sum_{j: j > i} x_{ij} + \sum_{j: j < i} x_{ji} = 2 \quad \text{for all } i = 1, \dots, N, \quad (9)$$

$$\sum_{\{i, j\} \in E(S)} x_{ij} \leq |S| - 1 \quad \text{for all } S \subseteq V : 3 \leq |S| \leq \frac{N}{2}, \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i < j,$$

where $E(S) := \{\{i, j\} | i, j \in S, i < j\}$ denotes a set of all edges between nodes in $S \subseteq V$. There are N degree constraints (9), and if $N > 5$, we also have subtour elimination constraints (10). Note that inequalities (10) for subsets S of size 2 is equivalent to the upper bound of the variables. In the original formulation by Dantzig et al. (1954) given above, we get redundant constraints if N is even, because for every set S with $|S| = \frac{N}{2}$ we have a constraint corresponding to S and a constraint corresponding to $V \setminus S$. If we remove the redundant constraints, e.g., by only taking subsets of size $\frac{N}{2}$ containing node 1, we are left with $2^{N-1} - 1 - N - \frac{N(N-1)}{2}$ subtour elimination constraints.

To create an equivalent QUBO formulation, one needs to incorporate constraints into the objective function as penalty terms, involving binary variables only. As before, the QUBO formulation (14) will be equivalent to (DFJ) for large enough M , e.g., $M = \max c_{ij} + 1$ (see Supplement B for details). Each constraint from the family (9) induces a term of the form $M(\sum_{j: j > i} x_{ij} + \sum_{j: j < i} x_{ji} - 2)^2$, but no new variables. The subtour elimination constraints (10) require slack variables in binary representation. A constraint corresponding to subset size $|S|$ requires $\log_2 |S| \leq \log_2 N$ new binary variables to represent the possible integer slack. Therefore, the family of constraints (10) requires at most $(2^{N-1} - 1 - N - \frac{N(N-1)}{2}) \log_2 N$ additional binary variables. More careful calculation presented in Supplement C yields the following result.

Lemma 5. The Dantzig–Fulkerson–Johnson ILP can be reformulated as a QUBO having

$$n^{\text{DFJ}} := \frac{N(N-1)}{2} + \sum_{k=3}^{\lceil N/2 \rceil - 1} \binom{N}{k} \cdot \lceil \log_2(k-1) + 1 \rceil$$

$$+ \frac{1 + (-1)^N}{4} \cdot \binom{N}{\lfloor N/2 \rfloor} \cdot \lceil \log_2(N/2 - 1) + 1 \rceil$$

binary variables. Therefore, it requires n^{DFJ} logical qubits and can be solved on a neutral-atom-based QC with $4(n^{\text{DFJ}})^2$ physical qubits, a quantum annealer with $24 \lceil \frac{n^{\text{DFJ}}+10}{12} \rceil \lceil \frac{n^{\text{DFJ}}-2}{12} \rceil$ physical qubits, or on a general gate-based QC with n^{DFJ} physical qubits.

Finally, we remark that formulation (DFJ) is rarely passed completely to a solver even as an integer program, but is instead used as basis for row generation approaches. It might be possible to apply a similar approach, generating penalty terms and slack variables on the fly, in an iterative quantum–classical hybrid algorithm, which might constitute a potential direction for further research.

Miller–Tucker–Zemlin formulation. An alternative approach (Miller et al., 1960) introduces variables $u_j \in \mathbb{N}$ denoting the number of each node $j \in V$ in a tour (e.g., $u_5 = 7$ means that node 5 is the seventh city in the tour). We also use variables x_{ij} indicating whether edge $\{i, j\}$ is traversed. This time, however, we also have to keep track of the traversal direction, i.e., we have two different variables x_{ij} and x_{ji} for all $i, j \in V$ with $i \neq j$. Here, $x_{ij} = 1$ indicates that j is visited directly after i . Since we assume that a tour starts at node 1, for all $i \in V$, $j \in V \setminus \{1\}$, we must have $u_j \geq u_i + 1$ if $x_{ij} = 1$. Note that such constraint would be violated by any subtour not passing through node 1. This yields:

$$\min \sum_{i=1}^N \sum_{j: j \neq i} c_{ij} x_{ij}, \quad (\text{MTZ})$$

$$\begin{aligned} \text{s. t. } & \sum_{j: j \neq i} x_{ij} = 1 && \text{for all } i = 1, \dots, N, \\ & \sum_{i: i \neq j} x_{ij} = 1 && \text{for all } j = 1, \dots, N, \\ & u_i - u_j + (N-1)x_{ij} \leq N-2 && \text{for all } 2 \leq i, j \leq N, i \neq j, \\ & x_{ij} \in \{0, 1\} && \text{for all } 1 \leq i, j \leq N, i \neq j, \\ & u_j \geq 2 && \text{for all } j = 2, \dots, N, \end{aligned} \quad (11)$$

where we assume the distance matrix to be symmetric: $c_{ij} = c_{ji}$. We have $N(N-1)$ binary variables and $(N-1)$ integer variables. However, now there is only a polynomial number of constraints, namely, $2N$ degree constraints and $(N-1)(N-2)$ constraints (11) involving the new variables. In the above formulation we did not demand integrality of the u_j variables because it is not necessary and for classical hardware, omitting this constraint may accelerate the solver. In the following transformation to a QUBO, we will however assume that the $u_j \in \mathbb{N}$ and, again, that M is large enough, e.g., $M = 2 \max c_{ij} + 1$ (see Supplement B).

Lemma 6. The Miller–Tucker–Zemlin integer program can be reformulated as a QUBO with

$$n^{\text{MTZ}} := (N-1)((N-1) \cdot \lceil \log_2(N-2) \rceil + 2N-3)$$

binary variables. So, it requires n^{MTZ} logical qubits and can be solved on a neutral-atom-based QC with $4(n^{\text{MTZ}})^2$ physical qubits, a quantum annealer with $24 \lceil \frac{n^{\text{MTZ}}+10}{12} \rceil \lceil \frac{n^{\text{MTZ}}-2}{12} \rceil$ physical qubits, or a general gate-based QC with n^{MTZ} physical qubits.

Proof. The first two families of constraints give us $2N$ penalty terms in the objective. Each integer variable u_j must be represented by binary variables. In this case we can replace u_j by $2 + \sum_{r=0}^{\lceil \log_2(N-2) \rceil} 2^r b_{j,r}$, introducing $\lceil \log_2(N-2) \rceil + 1$ binary variables $b_{j,r}$. The slack in constraint (11) can take values from 0 (if j is the successor of i) to $2(N-2)$ (if $u_i = 2$, $u_j = N$). To accommodate this range, we need $\lceil \log_2(N-2) \rceil + 2$ binary variables. Since we have this for every pair of $i, j \in \{2, \dots, N\}$ with $i \neq j$, the total number of such variables is $(N-1) \cdot (N-2) \cdot (\lceil \log_2(N-2) \rceil + 2)$.

Therefore, the original ILP can be transformed into a QUBO model with $2N + (N - 1)(N - 2)$ penalty terms, involving

$$(N - 1) \cdot ([\log_2(N - 2)] + 1) + (N - 1) \cdot (N - 2) \cdot ([\log_2(N - 2)] + 2) \\ = (N - 1)((N - 1)[\log_2(N - 2)] + 2N - 3)$$

binary variables. $\square$

Quadratic assignment formulation. While these ILP approaches are very successful on classical computers, a direct translation of the formulations to QUBOs results in a large number of variables and thus of physical qubits required. In quantum computing, it makes sense to move away from ILP-based approaches and consider alternative strategies. This is well illustrated by the TSP, which actually has a very natural and compact formulation as a quadratic program, more specifically a quadratic assignment problem (Garfinkel, 1985; Lawler, 1963). Let us introduce binary variables y_{ik} , which equal one if and only if node i is visited during step k in the tour. Since we are interested only in tours that return to the original location, assume without loss of generality that $y_{11} = 1$. These ideas yield the following formulation:

$$\min \sum_{j=2}^N c_{1j} y_{j2} + \sum_{i=2}^N \left(\sum_{j=2}^N c_{ij} \sum_{k=2}^{N-1} y_{ik} y_{j(k+1)} + c_{1i} y_{iN} \right), \quad (\text{QAP})$$

$$\text{s. t. } \sum_{k=2}^N y_{ik} = 1 \quad \text{for all } i = 2, \dots, N. \quad (12)$$

$$\sum_{i=2}^N y_{ik} = 1 \quad \text{for all } k = 2, \dots, N. \quad (13)$$

$$y_{ik} \in \{0, 1\}, \quad \text{for all } i = 2, \dots, N \text{ and } k = 2, \dots, N$$

Each coefficient c_{ij} encodes the cost of edge $\{i, j\} \in E$. Whenever nodes i and j appear as consecutive steps in the tour, $y_{ik} = y_{j(k+1)} = 1$ for some moment k , which contributes c_{ij} to the objective. The sums in the first and the last terms in (QAP) implement the same logic for the first step in the tour (from node 1 to j), and the last one (returning from node i to 1), respectively. Constraints (12) and (13) ensure that a solution represents a permutation of cities. It yields the following QUBO:

$$\min \sum_{j=2}^N c_{1j} y_{j2} + \sum_{i=2}^N \left(\sum_{j=2}^N c_{ij} \sum_{k=2}^{N-1} y_{ik} y_{j(k+1)} + c_{1i} y_{iN} \right) \\ + M \sum_{i=2}^N \left(\sum_{k=2}^N y_{ik} - 1 \right)^2 \\ + M \sum_{k=2}^N \left(\sum_{i=2}^N y_{ik} - 1 \right)^2 \quad (14) \\ y_{ik} \in \{0, 1\}, \quad i = 2, \dots, N \text{ and } k = 2, \dots, N.$$

As before, this QUBO problem is equivalent to the original formulation in the sense that optimal solutions coincide, if the penalty coefficient M is chosen to be large enough, e.g., $M = \frac{N(N-1)}{2} \max c_{ij} + 1$ (see Supplement B). Formulation (QAP) involves $(N - 1)^2$ binary variables, $2(N - 1)$ linear equality constraints, and a quadratic objective with $(N - 1)((N - 2)^2 + 2)$ terms, which we summarize in the following result.

Lemma 7. *TSP over a complete graph $G = (V, E)$ can be formulated as a QUBO with $(N - 1)^2$ binary variables. Therefore, it requires $(N - 1)^2$ logical qubits and can be solved on a neutral-atom-based QC with $4(N - 1)^4$ physical qubits, a quantum annealer with $24 \lceil \frac{(N-1)^2+10}{12} \rceil \lceil \frac{(N-1)^2-2}{12} \rceil$ physical qubits, or a general gate-based QC with $(N - 1)^2$ physical qubits.*

Although less widespread in OR literature, these three formulations yield QUBO instances of significantly different sizes for the same problem, leading to different qubit requirements in a quantum computing context. A summary is provided in Table 1, where we list the general expressions in terms of the number of nodes $N = |V|$, and illustrative

numerical values for $N = 10$. For example, a naive, full implementation of the formulation (DFJ) requires the fewest binary variables, but the exponential number of inequality constraints contributes to an exponential number of variables in the QUBO instance, which translates to more than a thousand for $N = 10$ nodes. Furthermore, the formulation (MTZ) reduces the number of inequality constraints and consequently brings the number of variables in the QUBO instance down to around 400. Finally, the formulation (QAP) does require slack variables at all, resulting in a relatively compact QUBO instance with only 81 variables.

4.4. Summary

The problem classes we consider in Sections 4.1 to 4.2 and 4.3 scale differently in terms of the number of binary variables with respect to the graph size. Depending on the chosen quantum-powered optimization approach from Section 3, the number of binary variables can be associated with a number of physical qubits that are necessary to run the problem. Upper bounds on the numbers of physical qubits are provided by Lemmas 1 and 2 and Remark 1. These bounds are not necessarily tight and depend on properties of the QUBO matrix Q , such as sparsity. A more detailed discussion is provided in Supplement A. An overview of the qubit requirements is presented in Table 2.

In summary, the qubit requirements are significant for all three problem classes, considering the capacities of currently available quantum devices. Especially for TSP, the quadratic scaling leads to very demanding requirements. The situation is better for MaxCut and UD-MIS, which both scale only linearly. For NA-OPT, both TSP and MaxCut lead to prohibitively high qubit requirements. On the other hand, we were able to solve fairly large UD-MIS instances on the neutral-atom-based device, as we will present in the next section.

Overall, given the necessities of the device-dependent setups and the current state of the technology, we see QA-OPT as the most practice-ready approach of the three. Similarly, QAOA-OPT represents a highly promising research direction with great versatility, which poses relatively modest qubit requirements. However, the available hardware devices can only be used to solve problems of a very limited scale. While NA-OPT is a feasible approach for UD-MIS, its applicability to other problem classes requires a large number of physical qubits.

Beyond the mere qubit requirements, there are other factors that have to be taken into account when choosing a suitable quantum-powered optimization approach. In the next section, we will therefore conduct numerical experiments to evaluate some practical aspects of quantum optimization.

5. Numerical experiments

The primary goal of our numerical experiments is not to compare the performance of state-of-the-art classical and quantum optimizers, but rather to illustrate the “out-of-the-box experience” with various quantum optimization strategies, and to discuss the associated challenges and opportunities. Informally, this can be seen as a comparison of the user experience assuming default parameters across the different approaches. We begin by describing the experimental setup in Section 5.1, followed by a presentation of the results in Section 5.2. Since the number of considered instances is relatively small (particularly for QAOA-OPT), our findings should be considered illustrative.

5.1. Experimental setup

We compare five optimization approaches across three problem classes, as summarized in Table 3.

Table 1

Number of variables for a TSP instance with N nodes, depending on the chosen QUBO formulation. The corresponding number of required qubits is provided by Lemma 7.

	Dantzig–Fulkerson–Johnson Eq. (DFJ)	Miller–Tucker–Zemlin Eq. (MTZ)	Quadratic assignment Eq. (QAP)
Constraints:			
Equality	N	$2N$	$2(N-1)$
Inequality	$2^{N-1} - 1 - \frac{N(N+1)}{2}$	$(N-1)(N-2)$	–
Variables:			
Binary	$N(N-1)/2$	$N(N-1)$	$(N-1)^2$
Integer	–	$N-1$	–
QUBO: the number of variables required			
Per binary	1	1	1
Per integer ^a	$\leq \log_2 N$	$\leq \log_2 N$	–
Total	$O(2^N \log_2 N)$	$O(N^2 \log_2 N)$	$O(N^2)$
Exact formula	See Lemma 5	See Lemma 6	See Lemma 7
Example: a TSP on $N = 10$ nodes.			
Constraints:			
Equality	10	20	18
Inequality	456	72	–
Variables:			
Binary	45	90	81
Integer	456	9	–
QUBO: the number of variables required			
Per binary	1	1	1
Per integer	≤ 4	≤ 4	–
Total	$\sim 1100^b$	~ 400	81

^a This includes slack variables (one per inequality constraint).

^b The value of $456 \times 4 + 45 = 1869$ is an upper bound, but we used a refined estimate based on Lemma 5.

Table 2

Number of the binary variables and number of physical qubits necessary, depending on the quantum-powered approach and problem class.

Number of variables		Required number of physical qubits		
Original formulation	QUBO	NA-OPT (Section 3.1)	QA-OPT (Section 3.2)	QAOA-OPT (Section 3.3)
		UD-MIS representation, (Nguyen et al., 2023)	Clique embedding (Lemma 2)	No reformulation ^a
MIS (Section 4.1) over a graph with N vertices (examples are given for $N = 10$), ILP formulation Eq. (2):				
• N binary variables (e.g.: 10) • up to $N(N-1)/2$ inequality constraints (e.g.: 45)	N binary variables, see Lemma 3. (e.g.: 10)	Arbitrary graph: $4N^2$ (e.g.: ~ 400 qubits) Unit disk graph: N (e.g.: 10 qubits)	$24 \lceil \frac{N+10}{12} \rceil \lceil \frac{N-2}{12} \rceil$ (e.g.: ~ 50 qubits)	N (e.g.: 10 qubits)
MaxCut (Section 4.2) over a complete graph with N vertices (examples are given for $N = 10$), ILP formulation Eq. (5):				
• $N(N+1)/2$ binary variables (e.g.: 55) • $N(N-1)$ inequality constraints (e.g.: 90)	N binary variables, see Lemma 4. (e.g.: 10)	$4N^2$ (e.g.: ~ 400 qubits)	$24 \lceil \frac{N+10}{12} \rceil \lceil \frac{N-2}{12} \rceil$ (e.g.: ~ 50 qubits)	N (e.g.: 10 qubits)
TSP (Section 4.3) over a complete graph with N vertices (examples are given for $N = 10$). Quadratic assignment formulation Eq. (QAP):				
• $(N-1)^2$ binary variables (e.g.: 81) • $2(N-1)$ equality constraints (e.g.: 18)	$(N-1)^2$ binary variables, constraints as penalties, see Lemma 7. (e.g.: 81)	$4(N-1)^4$ (e.g.: $\sim 26\,000$ qubits)	$24 \lceil \frac{(N-1)^2+10}{12} \rceil \lceil \frac{(N-1)^2-2}{12} \rceil$ (e.g.: ~ 1350 qubits)	$(N-1)^2$ (e.g.: 81 qubits)
For reference: number of qubits for largest QC available		256	5000+	1000+ ^b

^a IBM gate-based QC requires device dependent *transpilation* to ensure the necessary qubit connectivity, which does not require additional qubits.

^b Castelvécchi (2023).

Problem instances. We consider the following randomized procedures to generate instances for the three problem classes. More details can be found in Supplement A.

- **UD-MIS:** To generate hardware-compliant instances for *Aquila*, we construct unit disk graphs effectively by sampling the nodes from a grid randomly. (Graph connectivity is then determined by unit disk radius.)
- **MaxCut:** Instances are generated using the Erdős-Rényi random graph model (Erdős & Rényi, 1959) with randomly selected edges and uniformly random edge weights within specified bounds.

- **TSP:** We select instances from the TSPLIB (Reinelt, 1991) and generate smaller TSP instances from them by randomly sampling subsets of nodes while preserving edge weights. Each sampled subset forms a complete subgraph, retaining the original distance structure.

Optimization approaches. We consider three types of optimization approaches:

- **Quantum-powered:** The three quantum-powered approaches discussed in Section 3.

Table 3

Each optimization approach is used to solve a selection of instances across problem classes.

Approach	Type	Considered problem classes and formulations
NA-OPT	Quantum-powered	UD-MIS (analog/native)
QA-OPT	Quantum-powered	UD-MIS (4), MaxCut (8), TSP (14)
QAOA-OPT	Quantum-powered	UD-MIS (4), MaxCut (8), TSP (14)
SIM-OPT	Quantum-simulated	UD-MIS (4), MaxCut (8), TSP (14)
Gurobi	Classical	UD-MIS (2), MaxCut (8), TSP (MTZ)

- **Quantum-simulated:** In addition to the QAOA-OPT approach, we also consider a simulated variant running entirely on classical hardware. To that end, the *IBM* QPU is replaced by the cloud-based *QASM Simulator* from *IBM* (Javadi-Abhari et al., 2024). We configure the simulator in such a way that it performs a noise-free, idealized simulation of a 32-qubit quantum device.¹¹ We refer to this approach as SIM-OPT in the following.
- **Baseline:** We use the commercial solver by Gurobi (2024) with default parameters. The time limit is set to 5 min per instance, or beyond that time until the optimality gap of 5% is reached, but no more than 20 min. Our choice of models was driven by the aim to take a small integer model that would seem as a natural first choice from an OR perspective. We took this approach for UD-MIS and TSP, but not for the MaxCut problem, because the preliminary experiments proved that for this problem the quadratic formulation could be solved faster by Gurobi than the simple ILP. Further, note that we are choosing an exact solution approach as a baseline, because our goal is not to compare the state-of-the-art heuristics, but to capture some simple characteristic of instance “hardness” and highlight a group of instances that may warrant further investigation. We briefly revisit this issue further. See Supplement I for more details.

For the quantum-powered optimization approaches, we make use of the QCs by using the cloud-based services of the respective hardware providers. To that end, for each use of the QC, the corresponding quantum computing task (comprising high-level device instructions) is sent to a service platform, where it is placed in a scheduler queue. This scheduler is necessary because we rely on publicly available services, which may be accessed by multiple users concurrently. As a result, significant delays may occur in completing a quantum computing task, depending on hardware utilization at the time of the request. Especially for QAOA-OPT, these delays can become significant due to the hybrid nature of the approach, which involves multiple iterative quantum computing tasks. Although the scheduler can account for hybrid algorithms by adjusting task prioritization, this is only partially effective. No guaranteed response times were provided on any of the platforms during our experiments.

5.2. Results

The evaluation of our numerical experiments is focused on two key characteristics for the end user: end-to-end algorithm runtimes and the resulting solution quality. In the following, we refer to each attempt to solve a single instance with a specific approach as a *run*. Note that for NA-OPT and QA-OPT, a run implies aggregation over a sequence of shots obtained from the respective QCs, which yields a single objective value (using the best value across all samples). Furthermore, a run for QAOA-OPT implies a sequence of quantum–classical iterations, where each quantum iteration involves a sequence of shots, which are used for intermediate calculations and the resulting objective value (again,

using the best value across all final samples). The runtime includes both the necessary configuration steps performed on a classical computer and the actual QC runtime (total of all shots), but does not include the classical procedure of the problem reformulation to UD-MIS (for NA-OPT) or to QUBO (for QA-OPT and QAOA-OPT).

For the evaluation presented here, we consider a set of runs where each solution approach attempts to solve each problem instance across all three problem classes exactly once (see also Supplement F).

Scope. The scope of the numerical experiments is visualized in Fig. 6. The height of each bar indicates the number of instances of a certain size (horizontal axis) of the respective problem class (in columns) that is to be solved with a specific optimization approach (in rows). In other words, each counted instance represents a run. The three colors indicate the nature of the optimization result for each run: A feasible, though not necessarily optimal, solution (“success”), an infeasible solution (“infeasible”), or no result at all due to a technical reason (“fail”), for example because of an early termination due to a timeout or a network error. In contrast, some runs were not started at all, because the waiting time in the queue exceeded the time frame planned for this study. These non-started runs are not further considered and do not contribute to Fig. 6. Selected example for solutions from the quantum-powered approaches are presented in Supplement G.

Our original aim was to consider a similar number of instances per problem size across approaches and problem classes. However, the results obtained are not fully balanced in this regard for various reasons. First, we did not attempt to solve MaxCut and TSP instances with the NA-OPT approach due to the infeasible scaling of their UD-MIS reformulations. Second, SIM-OPT supports only instances with up to 32 binary variables. Finally, some minor irregularities in the number of solved instances are attributable to technical and organizational constraints, such as varying queue times and limited computational resources available in the scope of this study. This has primarily an effect on QAOA-OPT because of the hybrid quantum–classical algorithm, which requires a reoccurring availability of the quantum hardware over sufficiently many iterations, an elaborate overall task. For some instances, the waiting time in the queue exceeded the time frame planned for this study.

We find that the success rates vary significantly between the different approaches. For the NA-OPT approach, we only consider UD-MIS instances, but achieve a total success rate of 100%. For the QA-OPT approach, the total success rate is 75%. Smaller instances have a higher success rate than larger instances, where minor embeddings become a significant bottleneck. We will revisit this topic further below. For the QAOA-OPT approach, the total success rate is only 21%.¹² Among the key practical reasons for so few instances solved are network connectivity issues and significantly longer queuing times in comparison with the other approaches. Most of the instances solved using the QAOA-OPT approach belong to the UD-MIS problem class, which is known to have sparser QUBO matrices, see Supplement A. Finally, for the SIM-OPT approach, the total success rate is 65%. The reason why this success rate is higher than for QAOA-OPT can be partially attributed to convergence issues of the classical optimization loop in presence of noise (the simulator is chosen to be noise-free).

Baseline results. Gurobi, as the classical baseline, found a feasible solution for all instances and is therefore not included in Fig. 6. Specifically, all TSP instances were all solved to optimality with the largest runtime of about 15 min. Furthermore, the UD-MIS instances were all solved to optimality in under 1 second. Larger MaxCut instances were more

¹¹ The simulator does not model physical noise effects, but it does simulate *shot noise* by performing 1000 shots per iteration, thereby introducing a certain level of (pseudo-)randomness.

¹² These results do not necessarily characterize the theoretical performance of variational algorithms on *IBM* hardware. However, we believe it is a good illustration of the point that QAOA-based approaches require significantly more work beyond a naive implementation in order to achieve meaningful results.

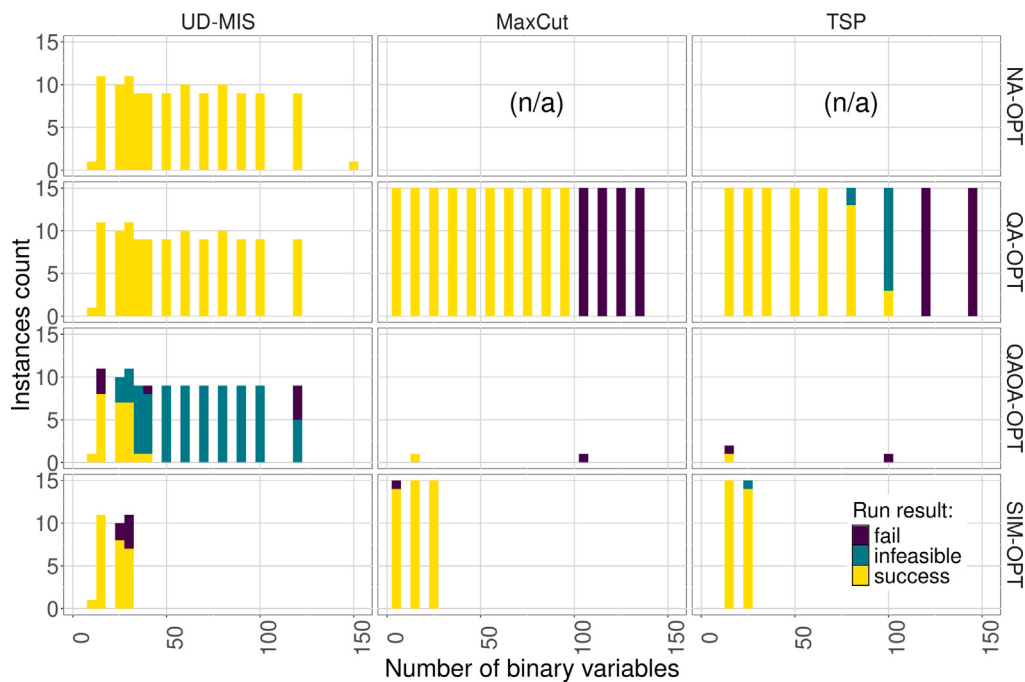


Fig. 6. Number of instances across different problem sizes and classes, which constitute the runs of our numerical experiments. The colors classify the outcomes of each run: “success” (a feasible, but not necessarily optimal solution was found), “infeasible” (an infeasible solution was found), or “fail” (no solution was obtained due to technical issues). Runs that did not start at all are not counted here, leading to empty spots in the plots. The distribution of runs is unbalanced for various reasons (also leading to blank spots): insufficient qubits to realize UD-MIS reformulation on NA-OPT or certain QUBO instances on SIM-OPT, as well as limited computational resources and varying queue times. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

time-consuming to solve. In total, 84% of MaxCut instances were solved with optimality gap of at most 5% (more than 99% with a gap of at most 10%). Note that there are specialized classical methods that might yield better runtimes (e.g., Rehfeldt et al., 2023 for MaxCut). Moreover, just imposing a time limit on the classical solver already yields a heuristic that takes only a fraction of time of the exact method, while ensuring comparable quality of solutions: For our selection of instances, *Gurobi* was very successful in finding good solutions early on, and spent significant amount of time on improving the bounds. (See also Supplement I.)

Performance metrics. To evaluate the performance of each run, we consider the *relative objective value deviation from the baseline* R_f and the *relative runtime deviation from the baseline* R_t ,

$$R_f := \frac{f_q - f_c}{f_c} \quad \text{and} \quad R_t := \frac{t_q - t_c}{t_c}, \quad (15)$$

respectively. Here, $f_q \in \mathbb{R}$ denotes the resulting objective value from the run of interest and $t_q \in \mathbb{R}_{>0}$ its runtime. Analogously, $f_c \in \mathbb{R}$ denotes the best objective value found by *Gurobi* for the same instance (which is not necessarily the optimal solution for some MaxCut instances), and $t_c \in \mathbb{R}_{>0}$ denotes the corresponding runtime. We presume here that $f_c > 0$ for all considered instances.

Runtimes. A runtime comparison of the considered approaches to *Gurobi* is presented in Fig. 7(a). Each point represents a single run, and the tilted line in the top left corner reflects the situation where the compared runtimes are equal. For the comparably small problem instances of interest, the runtimes for quantum-powered approaches (if feasible) were mostly longer than the classical baseline (points below the tilted line). However, there is a small set of problem instances for which the runtime of *Gurobi* is longer than for the QA-OPT approach (points above the tilted line). For this subset of “hard” MaxCut instances, *Gurobi* required more than 20 s for a solution. The “hard” instances are all larger MaxCut instances, and investigating their structure in the context of OR applications might constitute a promising direction for further quantum optimization research. A detailed view of the “hard” instances is shown in Fig. 7(b).

Objective values. A comparison of the objective values from the quantum-powered approaches with the objective values from *Gurobi* is presented in Table 4. No quantum-powered run has a resulting objective value better than the corresponding *Gurobi* result. We observe that for both MaxCut and TSP, the quantum-powered approaches found many solutions within 25% of the *Gurobi* solutions. For the subset of “hard” MaxCut instances, Fig. 7(b), where *Gurobi* required more than 20 s for a solution, the QA-OPT approach was able to find some solutions within 10%–15% from the baseline objective in only half of the respective runtime.¹³

While the proportion of instances that were solved by QA-OPT no worse than *Gurobi* is comparable across problem classes, the share of instances with a gap within 5% or 10% for the UD-MIS problem is noticeably lower than that for MaxCut. This might be surprising at first glance, given that the QUBO matrices for UD-MIS instances are significantly less dense. Understanding this effect might constitute a possible direction for further research. One possible reason could be the different structure of constraints and their effects of the penalty terms. In particular, relatively small deviations of the solution in terms of the Hamming distances (number of bitflips) might lead to large relative changes in the objective value. Analysis of the solution landscapes in terms of Hamming distances and effects of coefficients normalization might therefore constitute reasonable first steps towards this direction.

Further, each independent set constraint corresponding to a single edge, when formulated as a penalty term, is still relatively local and does not require many interaction terms in the objective. Therefore, the embedding process in the context of QA-OPT approach turns out less problematic. Formulating and falsifying such hypotheses might constitute a viable line of further research. A more careful approach to benchmarking involving a quantum annealer is discussed by Lubinski et al. (2024).

¹³ The set of instances where *Gurobi* took more than 20 s coincided with the set of instances where QA-OPT took no more time than *Gurobi*. We refer to such instances as “hard”.

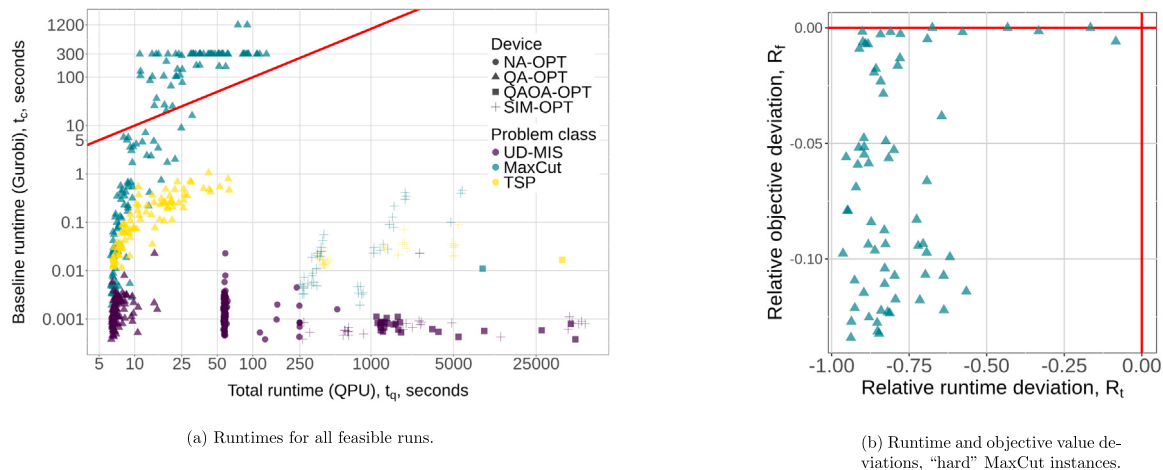


Fig. 7. (Color online) Performance evaluation of the quantum-powered approaches with *Gurobi* as a baseline. We observe in (a) that there are “hard” MaxCut instances for which the runtime of *Gurobi* is longer than the QA-OPT approach and takes more than 20 s to solve. These instances are shown in (b).

Table 4

Comparison of the objective values from the quantum-powered approaches f_q with the values from *Gurobi* f_c , as per Eq. (15). “Total” is the total number of feasible runs for the given problem class and solution approach, “within P%” shows the number of feasible runs with $|R_f| \leq P/100$ and its proportion of the “Total.”^a The last column describes the number of feasible runs no worse than the *Gurobi* objective, i.e., with $R_f \geq 0$ for MaxCut and UD-MIS, and $R_f \leq 0$ for TSP.

Problem class	Approach	Number of instance runs with relative objective deviations								
		Total	Within 25%		Within 10%		Within 5%		No worse	
UD-MIS	NA-OPT	117	93	79.5%	48	41.0%	34	29.1%	30	25.6%
	QA-OPT	116	116	100.0%	77	66.4%	54	46.6%	46	39.7%
	QAOA-OPT	25	9	36.0%	2	8.0%	2	8.0%	2	8.0%
	SIM-OPT	27	8	29.6%	3	11.1%	3	11.1%	3	11.1%
MaxCut	QA-OPT	149	149	100.0%	128	85.9%	105	70.5%	50	33.6%
	QAOA-OPT	1	1	–	0	–	0	–	0	–
	SIM-OPT	44	44	100.0%	44	100.0%	44	100.0%	20	45.5%
TSP	QA-OPT	91	61	67.0%	49	53.8%	46	50.5%	37	40.7%
	QAOA-OPT	1	1	–	1	–	0	–	0	–
	SIM-OPT	29	26	89.7%	22	75.9%	20	69.0%	17	58.6%

^a For example, with the approach NA-OPT, we obtain feasible solutions for 117 UD-MIS instances. Out of these 117 runs, 93 (i.e., 79.5% of the total of 117) achieve a 25% deviation from the *Gurobi* objective, and so on.

For the NA-OPT approach, the structure of constraints might also play a role for the performance. As discussed in Section 3.1, the implementation of UD-MIS instances on neutral-atom devices requires not only a hardware-compliant problem graph, but also a suitable geometric arrangement of the atoms. These geometric considerations, which directly affect the realizability and accuracy of the problem encoding, remain an important practical aspect of the NA-OPT approach (Wurtz et al., 2023). The atom placement in our experiments was directly driven by the instance generation process. A more sophisticated exploration of alternative placement strategies encoding the same UD-MIS instance may serve as a strategy to further enhance performance. While such a study lies beyond our intention to provide an “out-of-the-box experience”, it could serve as a starting point for future research.

In general, studying which characteristics of problem instances make them more or less suitable for quantum-powered optimization constitutes a possible further research direction (Koch et al., 2025), and the set of problem instances presented here could serve as a starting point for such an analysis.

Minor embeddings. To conclude, we report two key aspects of the necessary embedding process within the QA-OPT approach. First, the runtime of a classical computer to find the embedding and, second, the number of physical qubits necessary to realize it. As explained in Section 3.2, the number of required physical qubits can exceed the number of variables of the underlying QUBO due to the introduction of chains. See Supplement H for an illustration of a toy MaxCut instance embedding.

We found in our experiments that the bounds on the necessary numbers of qubits presented in Lemma 2 and Table 2 were not always tight, i.e., the heuristic algorithm used to find embeddings (Cai et al., 2014) performed better than the upper bound given in Lemma 2, with the specific results depending on the qubit connectivity required by the problem instances at hand. We quantified this effect by assessing the observed relation between the numbers of logical qubits N (from the problem instance) and physical qubits N_e (from the embedding) for the QA-OPT approach. Our statistical model, ordinary least squares regression, yields estimates for $N_e \in O(N^{\hat{\beta}})$, where $\hat{\beta} \approx 1.8$ for MaxCut and TSP instances, and $\hat{\beta} \approx 1.1$ for UD-MIS instances in our dataset. (See Supplement D for further details.) This agrees to our expectation that the share of nonzero entries in the QUBO matrix scales differently for different problem classes (see Supplement A), which affects the embedding overhead.

The embedding process also entails significant (classical) runtime costs. In our case, it dominates the solution time for large enough TSP and MaxCut instances, as illustrated in Fig. 8. For sparser UD-MIS instances with comparable numbers of binary variables, finding embeddings was relatively easy. While the embedding time varies with the number of binary variables in general, we did not scale the allotted annealing time. Therefore, for large and dense instances, up to 90% of the total runtime was spent trying to find an embedding, before starting the actual computation on the QC.

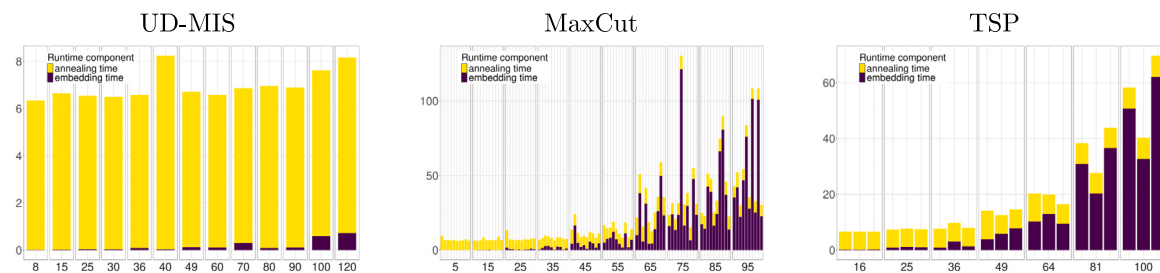


Fig. 8. Runtime histograms for the QA-OPT approach, per problem class. Each bar represents a randomly selected instance with a number of variables as indicated on the horizontal axis. Bar height represents the total runtime in seconds, as indicated on the vertical axis (different scale for each problem class). The total runtime constitutes the computation of a minor embedding (classical computer; dark color) and the QA computation (QC; light color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusion

It is difficult to assess which of the currently available quantum computing architectures is superior for OR applications. Different quantum algorithms may have different hardware requirements, and device characteristics may affect the details of algorithm implementations. As a result, the development of hardware and the development of algorithms for practical applications go together. For this reason, rather than attempting to benchmark a broad collection of methods, we highlight three specific quantum-powered approaches to solve discrete optimization problems. Specifically, we consider two analog approaches, NA-OPT (solving UD-MIS with an analog QC based on neutral atom technology) and QA-OPT (solving QUBOs with a quantum annealer based on superconducting technology), as well as a digital approach, QAOA-OPT (solving QUBOs with a gate-based QC based on superconducting technology). The paper offers three groups of key contributions.

First, we provide a high-level overview of the relevant workflows, aiming at a general OR audience. To the best of our knowledge, this is the first attempt to describe several different quantum approaches in a uniform framework as sketched in Fig. 2. We believe this effort will help motivate further involvement of the OR community into quantum optimization research. Fine-tuning of quantum algorithms and careful modeling of optimization problems may allow alternative quantum-powered solution strategies and therefore pose novel research opportunities (Blekos et al., 2024).

Second, we show that the required number of qubits for an optimization problem is not just a characteristic of the problem itself, but can be significantly affected by the chosen quantum technology, problem type, and the specific formulation, as listed in Tables 1 and 2. As discussed in Section 4, presenting new problem encodings is beyond the scope of this work, but we would like to emphasize the relevance of this issue in the context of quantum computing.

Third, we highlight the emergence of auxiliary optimization problems that arise in the course of applied quantum optimization in order to properly configure the respective quantum devices. Typically, such auxiliary problems are solved on a classical computer and could be a source for interdisciplinary collaborations between the quantum computing and OR communities. As a prototypical example, we discuss the embedding process within the QA-OPT approach and analyze the associated runtime and required number of physical qubits. We did not further explore the atom encoding step in the NA-OPT or the transpilation step in the QAOA-OPT, both of which represent distinct research directions. In general, improving heuristic methods for generating effective device configurations could enhance the practicality of all three quantum-powered approaches considered here.

Our numerical experiments have not shown a clear quantum advantage. The quality of the obtained numerical solutions depends on the problem class, but in general our classical baseline (solving an integer programming model using a state of the art commercial solver) outperformed the quantum-powered approaches, given enough time.

However, even with all the restrictions of the available quantum devices, we were able to generate a collection of MaxCut instances that were hard enough for the classical solver, and for which we were able to obtain heuristic solutions from the quantum device with reasonable quality faster than using a classical solver, although it is not difficult to design classical heuristics that would outperform the quantum-powered approaches we discuss over our selection of instances. Our experiments highlight the obvious fact that the sparsity of the QUBO matrix affects the problem complexity, both for the classical baseline and the quantum-powered approaches.

We have found that the quantum-powered optimization approaches discussed in this paper provide very different user experiences. While they can be described in a largely unified framework, each presents its own unique strengths and challenges. For example, the QA-OPT approach typically produced feasible solutions for instances up to a certain size, provided suitable embeddings could be identified. However, finding embeddings for highly connected problem instances can become very difficult. The NA-OPT approach was even more robust in terms of results. On the other hand, it was more limited in terms of the number of qubits and required a representation of the problem as a UD-MIS, which quickly leads to infeasible resource requirements. Finally, the QAOA-OPT approach, while the most flexible for algorithm development, required the most effort to establish a reliable computational workflow for technical reasons. These differences highlight the value of continued exploration across all quantum computing architectures. Rather than prematurely favoring one technology over another, it is important to recognize that each may be well suited to different types of OR problems. Improving our understanding of their computational capabilities and identifying promising application domains remain important directions for future research.

CRedit authorship contribution statement

Alexey Bochkarev: Writing – review & editing, Writing – original draft, Software, Methodology, Visualization, Project administration, Investigation, Conceptualization, Data curation. **Raoul Heese:** Writing – original draft, Software, Methodology, Data curation, Writing – review & editing, Visualization, Resources, Investigation, Conceptualization. **Sven Jäger:** Writing – review & editing, Visualization, Methodology, Conceptualization, Writing – original draft, Software, Investigation. **Philine Schiewe:** Writing – original draft, Methodology, Writing – review & editing, Visualization, Conceptualization. **Anita Schöbel:** Writing – review & editing, Project administration, Conceptualization, Supervision, Methodology.

Acknowledgments

The work was partially funded by the *Research Initiative Quantum Computing for Artificial Intelligence (QC-AI)*. Access to quantum hardware was funded by the *Fraunhofer Quantum Now* program.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.ejor.2025.07.063>.

References

- Abbas, A., Ambainis, A., Augustino, B., Bärttschi, A., Buhrman, H., Coffrin, C., Cortiana, G., Dunjko, V., Egger, D. J., Elmegreen, B. G., Franco, N., Fratini, F., Fuller, B., Gao, J., Gonciulea, C., Gribbling, S., Gupta, S., Hadfield, S., Heese, R., ... Zoufal, C. (2024). Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, 6(12), 718–735. <http://dx.doi.org/10.1038/s42254-024-00770-9>.
- Albash, T., & Lidar, D. A. (2018). Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1), Article 015002. <http://dx.doi.org/10.1103/revmodphys.90.015002>.
- Amazon Web Services (2020). Amazon braket. URL: <https://aws.amazon.com/braket/>.
- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2006). The traveling salesman problem: A computational study. *Princeton series in applied mathematics*: vol. 17, Princeton: Princeton University Press, <http://dx.doi.org/10.1515/9781400841103>.
- Au-Yeung, R., Chancellor, N., & Halffmann, P. (2023). NP-hard but no longer hard to solve? Using quantum computing to tackle optimization problems. *Frontiers in Quantum Science and Technology*, 2, Article 1128576. <http://dx.doi.org/10.3389/frqst.2023.1128576>.
- Baheri, B., Guan, Q., Chaudhary, V., & Li, A. (2022). Quantum noise in the flow of time: A temporal study of the noise in quantum computers. In *IEEE 28th int. symp. on-line testing and robust system design* (pp. 1–5). <http://dx.doi.org/10.1109/IOLTS56730.2022.9897404>.
- Barkoutsos, P. K., Nannicini, G., Robert, A., Tavernelli, I., & Woerner, S. (2020). Improving variational quantum optimization using CVaR. *Quantum*, 4, Article 256. <http://dx.doi.org/10.22331/q-2020-04-20-256>.
- Bera, M. N., Acín, A., Kuś, M., Mitchell, M. W., & Lewenstein, M. (2017). Randomness in quantum mechanics: philosophy, physics and technology. *Reports on Progress in Physics*, 80(12), Article 124001. <http://dx.doi.org/10.1088/1361-6633/aa8731>.
- Blekos, K., Brand, D., Ceschini, A., Chou, C.-H., Li, R.-H., Pandya, K., & Summer, A. (2024). A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 1068, 1–66. <http://dx.doi.org/10.1016/j.physrep.2024.03.002>.
- Bolusani, S., Besançon, M., Bestuzheva, K., Chmiela, A., Dionísio, J., Donkiewicz, T., van Doormalen, J., Eifler, L., Ghannam, M., Gleixner, A., Graczyk, C., Halbig, K., Hedtke, I., Hoen, A., Hojny, C., van der Hulst, R., Kamp, D., Koch, T., Kofler, K., ... Xu, L. (2024). The SCIP optimization suite 9.0. <http://dx.doi.org/10.48550/arXiv.2402.17702>, Preprint.
- Bombieri, L., Zeng, Z., Tricarico, R., Lin, R., Notarnicola, S., Cain, M., Lukin, M. D., & Pichler, H. (2025). Quantum adiabatic optimization with Rydberg arrays: Localization phenomena and encoding strategies. *PRX Quantum*, 6(2), Article 020306. <http://dx.doi.org/10.1103/prxquantum.6.020306>.
- Boothby, K., Bunyk, P., Raymond, J., & Roy, A. (2020). Next-generation topology of D-Wave quantum processors. <http://dx.doi.org/10.48550/arXiv.2003.00133>, Preprint.
- Breu, H., & Kirkpatrick, D. G. (1998). Unit disk graph recognition is NP-hard. *Computational Geometry*, 9(1), 3–24. [http://dx.doi.org/10.1016/S0925-7721\(97\)00014-X](http://dx.doi.org/10.1016/S0925-7721(97)00014-X).
- Bucher, D., Stein, J., Feld, S., & Linnhoff-Popien, C. (2025). IF-QAOA: A penalty-free approach to accelerating constrained quantum optimization. <http://dx.doi.org/10.48550/arXiv.2504.08663>, Preprint.
- Cai, J., Macready, W. G., & Roy, A. (2014). A practical heuristic for finding graph minors. <http://dx.doi.org/10.48550/arXiv.1406.2741>, Preprint.
- Castelvecchi, D. (2023). IBM releases first-ever 1,000-qubit quantum chip. *Nature*, 624(7991), 238. <http://dx.doi.org/10.1038/d41586-023-03854-1>.
- Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3(9), 625–644. <http://dx.doi.org/10.1038/s42254-021-00348-9>.
- Chandarana, P., Hegade, N. N., Paul, K., Albarrán-Arriagada, F., Solano, E., del Campo, A., & Chen, X. (2022). Digitized-counterdiabatic quantum approximate optimization algorithm. *Physical Review Research*, 4(1), Article 013141. <http://dx.doi.org/10.1103/physrevresearch.4.013141>.
- Choi, V. (2008). Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing*, 7, 193–209. <http://dx.doi.org/10.1007/s11128-008-0082-9>.
- Clark, B. N., Colbourn, C. J., & Johnson, D. S. (1990). Unit disk graphs. *Discrete Mathematics*, 86(1), 165–177. [http://dx.doi.org/10.1016/0012-365X\(90\)90358-O](http://dx.doi.org/10.1016/0012-365X(90)90358-O).
- Codognet, P. (2024). Comparing integer encodings in QUBO for quantum and digital annealing: The travelling salesman problem. In M. Sevaux, A.-L. Olteanu, E. G. Pardo, A. Sifaleras, & S. Makboul (Eds.), *Metaheuristics* (pp. 262–267). Springer Nature Switzerland, http://dx.doi.org/10.1007/978-3-031-62912-9_25.
- Creemers, S., & Armas, L. F. P. (2025). Discrete optimization: A quantum revolution? *European Journal of Operational Research*, 323(2), 378–408. <http://dx.doi.org/10.1016/j.ejor.2024.12.016>.
- Crooks, G. E. (2019). Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. <http://dx.doi.org/10.48550/arXiv.1905.13311>, Preprint.
- D-Wave (2024). D-Wave leap. URL: <https://cloud.dwavesys.com>.
- da Silva Coelho, W., D'Arcangelo, M., & Henry, L.-P. (2022). Efficient protocol for solving combinatorial graph problems on neutral-atom quantum processors. <http://dx.doi.org/10.48550/arXiv.2207.13030>, Preprint.
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393–410. <http://dx.doi.org/10.1287/opre.2.4.393>.
- Das, A., & Chakrabarti, B. K. (2005). *Quantum annealing and related optimization methods*. Springer Science & Business Media, <http://dx.doi.org/10.1007/11526216>.
- Dattani, N., Szalay, S., & Chancellor, N. (2019). Pegasus: The second connectivity graph for large-scale quantum annealing hardware. <http://dx.doi.org/10.48550/arxiv.1901.07636>, Preprint.
- Dominguez, F., Unger, J., Traube, M., Mant, B., Ertler, C., & Lechner, W. (2023). Encoding-independent optimization problem formulation for quantum computing. *Frontiers in Quantum Science and Technology*, 2, <http://dx.doi.org/10.3389/frqst.2023.1229471>.
- Ebadi, S., Keesling, A., Cain, M., Wang, T. T., Levine, H., Bluvstein, D., Semeghini, G., Omran, A., Liu, J., Samajdar, R., Luo, X.-Z., Nash, B., Gao, X., Barak, B., Farhi, E., Sachdev, S., Gemelke, N., Zhou, L., Choi, S., ... Lukin, M. D. (2022). Quantum optimization of maximum independent set using Rydberg atom arrays. *Science*, 376(6598), 1209–1215. <http://dx.doi.org/10.1126/science.abo6587>.
- Ebadi, S., Wang, T. T., Levine, H., Keesling, A., Semeghini, G., Omran, A., Bluvstein, D., Samajdar, R., Pichler, H., Ho, W. W., Choi, S., Sachdev, S., Greiner, M., Vuletić, V., & Lukin, M. D. (2021). Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866), 227–232. <http://dx.doi.org/10.1038/s41586-021-03582-4>.
- Egger, D. J., Mareček, J., & Woerner, S. (2021). Warm-starting quantum optimization. *Quantum*, 5, Article 479. <http://dx.doi.org/10.22331/q-2021-06-17-479>.
- Erdős, P., & Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae Debrecen*, 6(3–4), 290–297. <http://dx.doi.org/10.5486/pmd.1959.6.3-4.12>.
- Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. <http://dx.doi.org/10.48550/arXiv.1411.4028>, Preprint.
- Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. <http://dx.doi.org/10.48550/arXiv.quant-ph/0001106>, Preprint.
- Finnila, A. B., Gomez, M. A., Sebenik, C., Stenson, C., & Doll, J. D. (1994). Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219(5), 343–348. [http://dx.doi.org/10.1016/0009-2614\(94\)00117-0](http://dx.doi.org/10.1016/0009-2614(94)00117-0).
- Fuchs, F. G., & Bassa, R. P. (2024). LX-mixers for QAOA: Optimal mixers restricted to subspaces and the stabilizer formalism. *Quantum*, 8, 1535. <http://dx.doi.org/10.22331/q-2024-11-25-1535>.
- Fuchs, F. G., Lye, K. O., Møll Nilsen, H., Stasik, A. J., & Sartor, G. (2022). Constraint preserving mixers for the quantum approximate optimization algorithm. *Algorithms*, 15(6), Article 202. <http://dx.doi.org/10.3390/a15060202>.
- Garfinkel, R. S. (1985). Motivation and modeling. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, & D. B. Shmoys (Eds.), *Wiley series in discrete mathematics & optimization, The traveling salesman problem* (pp. 17–36). Chichester, UK: Wiley, chapter 2.
- Gilbert, V., Rodriguez, J., & Louise, S. (2024). Benchmarking quantum annealers with near-optimal minor-embedded instances. In *2024 IEEE int. conf. quantum computing and engineering* (pp. 531–537). IEEE, <http://dx.doi.org/10.1109/QCE60285.2024.00068>.
- Glover, F., Kochenberger, G., Hennig, R., & Du, Y. (2022). Quantum bridge analytics I: a tutorial on formulating and using QUBO models. *Annals of Operations Research*, 314(1), 141–183. <http://dx.doi.org/10.1007/s10479-022-04634-2>.
- Gomez-Tejedor, A., Osaba, E., & Villar-Rodriguez, E. (2025). Addressing the minor-embedding problem in quantum annealing and evaluating state-of-the-art algorithm performance. <http://dx.doi.org/10.48550/arXiv.2504.13376>, Preprint.
- Gonzalez-Bermejo, S., Alonso-Linaje, G., & Atchade-Adelomou, P. (2022). GPS: A new TSP formulation for its generalizations type QUBO. *Mathematics*, 10(33), Article 416. <http://dx.doi.org/10.3390/math10030416>.
- Goswami, K., Mukherjee, R., Ott, H., & Schmelcher, P. (2024). Solving optimization problems with local light shift encoding on Rydberg quantum annealers. *Physical Review Research*, 6(2), Article 023031. <http://dx.doi.org/10.1103/physrevresearch.6.023031>.
- Gottesman, D. (2022). Opportunities and challenges in fault-tolerant quantum computation. <http://dx.doi.org/10.48550/arXiv.2210.15844>, Preprint.
- Grange, C., Poss, M., & Bourreau, E. (2023). An introduction to variational quantum algorithms for combinatorial optimization problems. *4OR. A Quarterly Journal of Operations Research*, 21(3), 363–403. <http://dx.doi.org/10.1007/s10288-023-00549-1>.
- Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proc. 28th annu. ACM symp. theory of computing* (pp. 212–219). ACM, <http://dx.doi.org/10.1145/237814.237866>.
- Gurobi (2024). *Gurobi optimizer reference manual*. Gurobi Optimization, LLC, URL: <https://www.gurobi.com>.

- Gyongyosi, L., & Imre, S. (2019). A survey on quantum computing technology. *Computer Science Review*, 31, 51–71. <http://dx.doi.org/10.1016/j.cosrev.2018.11.002>.
- Hadfield, S. A. (2018). *Quantum algorithms for scientific computing and approximate optimization* (Ph.D. thesis), Columbia University, <http://dx.doi.org/10.7916/D8X650C9>.
- Hadfield, S. (2021). On the representation of Boolean and real functions as Hamiltonians for quantum computing. *ACM Transactions on Quantum Computing*, 2(4), Article 18. <http://dx.doi.org/10.1145/3478519>.
- Hadfield, S., Wang, Z., O’Gorman, B., Rieffel, E. G., Venturelli, D., & Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2), Article 34. <http://dx.doi.org/10.3390/a12020034>.
- Hadfield, S., Wang, Z., Rieffel, E. G., O’Gorman, B., Venturelli, D., & Biswas, R. (2017). Quantum Approximate Optimization with hard and soft constraints. In *Proc. 2nd int. workshop post Moores era supercomputing* (pp. 15–21). ACM, <http://dx.doi.org/10.1145/3149526.3149530>.
- Hangleiter, D., & Eisert, J. (2023). Computational advantage of quantum random sampling. *Reviews of Modern Physics*, 95(3), Article 035001. <http://dx.doi.org/10.1103/revmodphys.95.035001>.
- Harrow, A. W., & Montanaro, A. (2017). Quantum computational supremacy. *Nature*, 549(7671), 203–209. <http://dx.doi.org/10.1038/nature23458>.
- Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., & Oliver, W. D. (2020). Perspectives of quantum annealing: methods and implementations. *Reports on Progress in Physics*, 83(5), Article 054401. <http://dx.doi.org/10.1088/1361-6633/ab85b8>.
- Hoefler, T., Häner, T., & Troyer, M. (2023). Disentangling hype from practicality: On realistically achieving quantum advantage. *Communications of the ACM*, 66(5), 82–87. <http://dx.doi.org/10.1145/3571725>.
- Hua, F., Wang, M., Li, G., Peng, B., Liu, C., Zheng, M., Stein, S., Ding, Y., Zhang, E. Z., Humble, T., & Li, A. (2023). QASMTans: A QASM quantum transpiler framework for NISQ devices. In *SC’23, Proc. SC’23 workshops of int. conf. high performance computing, network, storage, and analysis* (pp. 1468–1477). ACM, <http://dx.doi.org/10.1145/3624062.3624222>.
- IBM Quantum (2023a). URL: <https://quantum.ibm.com/>.
- IBM Quantum (2023b). *IBM quantum documentation: Transpiler*. IBM, URL: <https://docs.quantum.ibm.com/api/qiskit/transpiler>.
- Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C. J., Lishman, J., Gacon, J., Martiel, S., Nation, P. D., Bishop, L. S., Cross, A. W., Johnson, B. R., & Gambetta, J. M. (2024). Quantum computing with qiskit. <http://dx.doi.org/10.48550/arXiv.2405.08810>, Preprint.
- Karp, R. M. (1975). On the computational complexity of combinatorial problems. *Networks*, 5(1), 45–68. <http://dx.doi.org/10.1002/net.1975.5.1.45>.
- Karuppusamy, K., Puram, V., Johnson, S., & Thomas, J. P. (2025). A comprehensive review of quantum circuit optimization: Current trends and future directions. *Quantum Reports*, 7(1), Article 2. <http://dx.doi.org/10.3390/quantum7010002>.
- Katabarwa, A., Gratsea, K., Caesura, A., & Johnson, P. D. (2024). Early fault-tolerant quantum computing. *PRX Quantum*, 5, Article 020101. <http://dx.doi.org/10.1103/PRXQuantum.5.020101>.
- Kim, Y., Eddins, A., Anand, S., Wei, K. X., Van Den Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K., et al. (2023). Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965), 500–505. <http://dx.doi.org/10.1038/s41586-023-06096-3>.
- Klug, F. (2024). Quantum optimization algorithms in operations research: methods, applications, and implications. <http://dx.doi.org/10.48550/arXiv.2312.13636>, Preprint.
- Koch, T., Neira, D. E. B., Chen, Y., Cortiana, G., Egger, D. J., Heese, R., Hegade, N. N., Cadavid, A. G., Huang, R., Itoko, T., Kleinert, T., Xavier, P. M., Mohseni, N., Montanez-Barrera, J. A., Nakano, K., Nannicini, G., O’Meara, C., Pauckert, J., Proissl, M., ... Zoufal, C. (2025). Quantum optimization benchmark library – The intractable decathlon. <http://dx.doi.org/10.48550/arXiv.2504.03832>, Preprint.
- Laurent, M. (1997). Max-Cut problem. In M. Dell’Amico, F. Maffioli, & S. Martello (Eds.), *Annotated bibliographies in combinatorial optimization* (pp. 241–259). Wiley, URL: <https://www.wiley.com/en-ie/Annotated+Bibliographies+in+Combinatorial+Optimization-p-9780470860700>.
- Lawler, E. L. (1963). The quadratic assignment problem. *Management Science*, 9(4), 586–599. <http://dx.doi.org/10.1287/mnsc.9.4.586>, URL: <https://www.jstor.org/stable/2627364>.
- Li, G., Ding, Y., & Xie, Y. (2019). Tackling the qubit mapping problem for NISQ-era quantum devices. In *ASPLOS’19, Proc. 24th int. conf. architectural support for programming languages and operating systems* (pp. 1001–1014). ACM, <http://dx.doi.org/10.1145/3297858.3304023>.
- Lobe, E., & Lutz, A. (2024). Minor embedding in broken chimera and derived graphs is NP-complete. *Theoretical Computer Science*, 989, Article 114369. <http://dx.doi.org/10.1016/j.tcs.2023.114369>.
- Lubinski, T., Coffrin, C., McGeoch, C., Sathe, P., Apanavicius, J., & Bernal Neira, D. (2024). Optimization applications as quantum performance benchmarks. *ACM Transactions on Quantum Computing*, 5(3), Article 18. <http://dx.doi.org/10.1145/3678184>.
- Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics*, 2, Article 5. <http://dx.doi.org/10.3389/fphy.2014.00005>.
- MacLennan, B. J. (2009). Analog computation. In R. A. Meyers (Ed.), *Encyclopedia of complexity and systems science* (pp. 271–294). New York, NY: Springer New York, <http://dx.doi.org/10.1007/978-0-387-30440-3.19>.
- Markidis, S. (2024). What is quantum parallelism, anyhow?. <http://dx.doi.org/10.48550/arXiv.2405.07222>, Preprint.
- McGeoch, C., & Farré, P. (2020). *The D-Wave advantage system: An Overview: Technical report 14-1049A-A*, D-Wave Systems Inc., URL: <https://www.dwavesys.com/resources/white-paper/the-d-wave-advantage-system-an-overview/>.
- Mermin, N. D. (2007). *Quantum computer science*. Cambridge: Cambridge University Press, <http://dx.doi.org/10.1017/CBO9780511813870>.
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4), 326–329. <http://dx.doi.org/10.1145/321043.321046>.
- Morita, S., & Nishimori, H. (2008). Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12), Article 125210. <http://dx.doi.org/10.1063/1.2995837>.
- Mücke, S., Gerlach, T., & Piatkowski, N. (2025). Optimum-preserving QUBO parameter compression. *Quantum Machine Intelligence*, 7(1), Article 1. <http://dx.doi.org/10.1007/s42484-024-00219-3>.
- Nannicini, G. (2020). An introduction to quantum computing, without the physics. *SIAM Review*, 62(4), 936–981. <http://dx.doi.org/10.1137/18M1170650>.
- Nation, P. D., & Treinish, M. (2023). Suppressing quantum circuit errors due to system variability. *PRX Quantum*, 4, Article 010327. <http://dx.doi.org/10.1103/PRXQuantum.4.010327>.
- Nguyen, M.-T., Liu, J.-G., Wurtz, J., Lukin, M. D., Wang, S.-T., & Pichler, H. (2023). Quantum optimization with arbitrary connectivity using Rydberg atom arrays. *PRX Quantum*, 4(1), Article 010316. <http://dx.doi.org/10.1103/PRXQuantum.4.010316>.
- Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th Anniversary). Cambridge, New York: Cambridge University Press, <http://dx.doi.org/10.1017/CBO9780511976667>.
- Padmasola, V., Li, Z., Chatterjee, R., & Dyk, W. (2025). Solving the traveling salesman problem via different quantum computing architectures. <http://dx.doi.org/10.48550/arXiv.2502.17725>, Preprint.
- Parekh, O. (2023). Synergies between operations research and quantum information science. *INFORMS Journal on Computing*, 35(2), 266–273. <http://dx.doi.org/10.1287/ijoc.2023.1268>.
- Pelofske, E. (2024). 4-clique network minor embedding for quantum annealers. *Physical Review Applied*, 21(3), Article 034023. <http://dx.doi.org/10.1103/PhysRevApplied.21.034023>.
- Pelofske, E., Hahn, G., & Djidjev, H. (2020). Advanced unembedding techniques for quantum annealers. In *Int. conf. rebooting computing* (pp. 34–41). IEEE, <http://dx.doi.org/10.1109/ICRC2020.2020.00001>.
- Pichler, H., Wang, S.-T., Zhou, L., Choi, S., & Lukin, M. D. (2018). Quantum optimization for maximum independent set using Rydberg atom arrays. <http://dx.doi.org/10.48550/arXiv.1808.10816>, Preprint.
- Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79. <http://dx.doi.org/10.22331/q-2018-08-06-79>.
- Rehfeldt, D., Koch, T., & Shinano, Y. (2023). Faster exact solution of sparse MaxCut and QUBO problems. *Mathematical Programming Computation*, 15(3), 445–470. <http://dx.doi.org/10.1007/s12532-023-00236-6>.
- Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing*, 3(4), 376–384. <http://dx.doi.org/10.1287/ijoc.3.4.376>.
- Ruan, Y., Marsh, S., Xue, X., Liu, Z., & Wang, J. (2020). The quantum approximate algorithm for solving traveling salesman problem. *Computers, Materials and Continua*, 63(3), 1237–1247. <http://dx.doi.org/10.32604/cmc.2020.010001>.
- Sack, S. H., Medina, R. A., Kueng, R., & Serbyn, M. (2023). Recursive greedy initialization of the quantum approximate optimization algorithm with guaranteed improvement. *Physical Review A*, 107, Article 062404. <http://dx.doi.org/10.1103/PhysRevA.107.062404>.
- Sack, S. H., & Serbyn, M. (2021). Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum*, 5, Article 491. <http://dx.doi.org/10.22331/q-2021-07-01-491>.
- Salehi, Ö., Glos, A., & Miszczak, J. A. (2022). Unconstrained binary models of the travelling salesman problem variants for quantum optimization. *Quantum Information Processing*, 21(2), Article 67. <http://dx.doi.org/10.1007/s1128-021-03405-5>.
- Scholten, T. L., Williams, C. J., Moody, D., Mosca, M., Hurley, W., Zeng, W. J., Troyer, M., & Gambetta, J. M. (2024). Assessing the benefits and risks of quantum computers. <http://dx.doi.org/10.48550/arXiv.2401.16317>, Preprint.
- Serret, M. F., Marchand, B., & Ayrat, T. (2020). Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks. *Physical Review A*, 102(5), Article 052617. <http://dx.doi.org/10.1103/PhysRevA.102.052617>.
- Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5), 1484–1509. <http://dx.doi.org/10.1137/S0097539795293172>.
- Sinno, S., Groß, T., Chancellor, N., Bhalgamiya, B., & Sahoo, A. (2025). Optimized quantum embedding: A universal minor-embedding framework for large complete bipartite graph. <http://dx.doi.org/10.48550/arXiv.2504.21112>, Preprint.

- Styer, D. F., Balkin, M. S., Becker, K. M., Burns, M. R., Dudley, C. E., Forth, S. T., Gaumer, J. S., Kramer, M. A., Oertel, D. C., Park, L. H., Rinkoski, M. T., Smith, C. T., & Wotherspoon, T. D. (2002). Nine formulations of quantum mechanics. *American Journal of Physics*, 70(3), 288–297. <http://dx.doi.org/10.1119/1.1445404>.
- Sugie, Y., Yoshida, Y., Mertig, N., Takemoto, T., Teramoto, H., Nakamura, A., Takigawa, I., Minato, S.-i., Yamaoka, M., & Komatsuzaki, T. (2021). Minor-embedding heuristics for large-scale annealing processors with sparse hardware graphs of up to 102,400 nodes. *Soft Computing*, 25(3), 1731–1749. <http://dx.doi.org/10.1007/s00500-020-05502-6>.
- Ulmann, B. (2024). Beyond zeros and ones – analog computing in the twenty-first century. *International Journal of Parallel, Emergent and Distributed Systems*, 39(2), 139–151. <http://dx.doi.org/10.1080/17445760.2023.2296672>.
- Venegas-Andraca, S. E., Cruz-Santos, W., McGeoch, C., & Lanzagorta, M. (2018). A cross-disciplinary introduction to quantum annealing-based algorithms. *Contemporary Physics*, 59(2), 174–197. <http://dx.doi.org/10.1080/00107514.2018.1450720>.
- Waring, J., Pere, C., & Le Beux, S. (2024). Noise aware utility optimization of NISQ devices. In *22nd IEEE interregional NEWCAS conf., NEWCAS 2024, Sherbrooke, QC, Canada, June 16-19, 2024* (pp. 168–172). IEEE, <http://dx.doi.org/10.1109/NewCAS58973.2024.10666318>, <https://doi.org/10.1109/NewCAS58973.2024.10666318>, arXiv:402.08226.
- Wilson, E., Singh, S., & Mueller, F. (2020). Just-in-time quantum circuit transpilation reduces noise. In *2020 IEEE int. conf. quantum computing and engineering* (pp. 345–355). IEEE, <http://dx.doi.org/10.1109/QCE49297.2020.00050>.
- Wintersperger, K., Dommert, F., Ehmer, T., Hoursanov, A., Klepsch, J., Mauerer, W., Reuber, G., Strohm, T., Yin, M., & Luber, S. (2023). Neutral atom quantum computing hardware: Performance and end-user perspective. *EPJ Quantum Technology*, 10(1), Article 32. <http://dx.doi.org/10.1140/epjqt/s40507-023-00190-1>.
- Wu, J., Lin, X., Guo, Y., Liu, J., Fang, L., Jiao, S., & Dai, Q. (2022). Analog optical computing for artificial intelligence. *Engineering*, 10, 133–145. <http://dx.doi.org/10.1016/j.eng.2021.06.021>.
- Wurtz, J., Bylinskii, A., Braverman, B., Amato-Grill, J., Cantu, S. H., Huber, F., Lukin, A., Liu, F., Weinberg, P., Long, J., Wang, S.-T., Gemelke, N., & Keesling, A. (2023). Aquila: QuEra's 256-qubit neutral-atom quantum computer. <http://dx.doi.org/10.48550/arXiv.2306.11727>, Preprint.
- Wurtz, J., Lopes, P. L. S., Gorgulla, C., Gemelke, N., Keesling, A., & Wang, S. (2022). Industry applications of neutral-atom quantum computing solving independent set problems. <http://dx.doi.org/10.48550/arXiv.2205.08500>, Preprint.
- Zangeneh-Nejad, F., Sounas, D. L., Alù, A., & Fleury, R. (2021). Analogue computing with metamaterials. *Nature Reviews Materials*, 6(3), 207–225. <http://dx.doi.org/10.1038/s41578-020-00243-2>.
- Zbinden, S., Bäertschi, A., Djidjev, H., & Eidenbenz, S. (2020). Embedding algorithms for quantum annealers with Chimera and Pegasus connection topologies. In P. Sadayappan, B. L. Chamberlain, G. Juckeland, & H. Ltaief (Eds.), *High performance computing* (pp. 187–206). Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-50743-5_10.
- Zhang, D.-J., Yu, X.-D., & Tong, D. M. (2014). Theorem on the existence of a nonzero energy gap in adiabatic quantum computation. *Physical Review A*, 90(4), Article 042321. <http://dx.doi.org/10.1103/physreva.90.042321>.
- Zhao, Y., Ye, Y., Huang, H.-L., Zhang, Y., Wu, D., Guan, H., Zhu, Q., Wei, Z., He, T., Cao, S., Chen, F., Chung, T.-H., Deng, H., Fan, D., Gong, M., Guo, C., Guo, S., Han, L., Li, N., Pan, J.-W. (2022). Realization of an error-correcting surface code with superconducting qubits. *Physical Review Letters*, 129, Article 030501. <http://dx.doi.org/10.1103/PhysRevLett.129.030501>.
- Zhou, L., Wang, S.-T., Choi, S., Pichler, H., & Lukin, M. D. (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2), Article 021067. <http://dx.doi.org/10.1103/PhysRevX.10.021067>.