
Quantum Informed Optimization Algorithms

Aron Kerschbaumer



Quantum Informed Optimization Algorithms

Aron Kerschbaumer

Master's Thesis in Theoretical and Mathematical Physics
Department of Informatics
Technische Universität München

Author: Aron Kerschbaumer
Supervisor: Prof. Dr. Christian Mendl
Advisor: Jernej Rudi Finžgar

München, May 2023

Summary

Quantum optimization is a promising field for practical quantum computing. Still, limitations in current quantum devices, such as the limited number of qubits and noise, restrict the applicability of these algorithms. This has led to considerable efforts in developing quantum optimization algorithms, which are feasible for noisy intermediate-scale quantum (NISQ) devices. One promising candidate for solving combinatorial optimization problems is the Quantum Approximate Optimization Algorithm (QAOA). Recently, several limitations of this algorithm have been identified. This work focuses on addressing these limitations, developing NISQ-feasible algorithm modifications to improve its performance, and developing novel approaches, such as using low-energy quantum states to inform classical algorithms. Drawing on the principles of the Recursive-QAOA, new iterative quantum-classical optimization techniques were developed. These methods were specifically developed for the Max-2-SAT problem, resulting in a new hybrid quantum-classical algorithm called Quantum Informed Branch and Bound Algorithm. However, many components of the algorithm are applicable to a wide range of combinatorial optimization problems and may serve as a blueprint for the development of related problem-tailored algorithms. Furthermore, another algorithm that utilizes QAOA to calculate correlations between variables and inform a classical cluster algorithm was developed. This so-called QAOA Informed Cluster Algorithm is used to calculate the ground state of Ising-like spin glasses. The techniques developed in this thesis can be extended beyond QAOA - other quantum protocols which prepare low-energy states could be used in a similar manner to inform classical subroutines. However, the utilization of low-depth QAOA for the proposed algorithms enables efficient classical simulations. As a result, numerical simulations in this thesis could be performed on problem instances exhibiting hundreds of variables, showcasing comparable performance between the developed hybrid algorithms and traditional classical solvers. The findings presented in this thesis are believed to be useful in a broader context of developing new heuristic quantum optimization algorithms for near-term devices.

Acknowledgements

I am enormously grateful for all the support I received while writing this thesis. Without the help of so many people, this thesis wouldn't have been possible in this form and I want to express my heartfelt thanks to all of them.

First and foremost, I want to thank my advisor Jernej Rudi Finžgar. Over the last year, his support and encouragement have fundamentally shaped my master's thesis. I genuinely cherished our often (very) lengthy, but (almost) always fruitful discussions, and I am grateful to have been able to work alongside him. He consistently made time for my questions and problems, no matter how trivial or self-inflicted they may have been. Not only did we form an unbeatable foosball team, but under his guidance, I think we formed an exceptional scientific team. I sincerely thank him for the past year, as I have learned so much from his guidance, both within and beyond the scientific realm.

I would also like to express my gratitude to my supervisor, Prof. Christian Mendl for his support throughout the thesis. His knowledge of quantum computing and beyond greatly contributed to the quality of my work. Christian always made himself available for questions and discussions and therefore definitely improved the quality of this work.

Lastly, I am tremendously thankful for the support provided by BMW. Not only did they provide financial assistance and resources, but the privilege to work alongside such an exceptional team of dedicated people was truly amazing. I would like to express my deep appreciation to Andre Luckow and Johannes Klepsch for their great leadership and for creating such an incredible working atmosphere. Their support extended beyond work-related matters, and I greatly enjoyed engaging in technical discussions as well as informal conversations with them. I want to thank the entire team at BMW who made my time so very pleasant. I would like to specifically mention Philipp Ross, Carlos Riofrío, Marvin Erdmann, and Ann Christin Rathje. Working alongside such talented and wonderful people has been a privilege, and I consider myself incredibly fortunate for that experience.

Statement of Authorship

I declare that this master's thesis is the result of my original work, and I have documented all contributions and content that are not my own accordingly.

Contents

Zusammenfassung	v
Acknowledgements	vi
Statement of Authorship	viii
1 Theory	1
1.1 Combinatorial Optimization Problems	1
1.1.1 Ising-like Spin Glasses	2
1.1.2 The SAT Problem and its Descendents	3
1.1.3 Max-Cut and Maximum Independent Set Problem	5
1.2 Computational Complexity and Algorithmic Hardness	5
1.2.1 Classical Complexity Theory	5
1.2.2 Quantum Complexity Theory	6
1.2.3 Algorithmic Hardness - An Alternative Approach	7
1.3 Classical Algorithms	11
1.3.1 Max-SAT/Branch and Bound Solver	11
1.3.2 Metropolis Hastings and Simulated Annealing	13
1.3.3 Parallel Tempering	15
1.4 Quantum Approximate Optimization Algorithm	16
1.4.1 Overview	16
1.4.2 Locality	17
1.4.3 Limitations	18
2 Quantum Informed Algorithms	19
2.1 Recursive-QAOA	19
2.1.1 Algorithm	20
2.1.2 Results	26
2.1.3 Discussion	32
2.1.4 Outlook	32
2.2 QAOA Informed Branch and Bound Solver	34
2.2.1 Algorithm	34
2.2.2 Results	41
2.2.3 Discussion	45

2.2.4	Outlook	46
2.3	Quantum Informed Cluster Algorithm	48
2.3.1	Algorithm	48
2.3.2	Results	51
2.3.3	Discussion	56
2.3.4	Outlook	58

Chapter 1

Theory

1.1 Combinatorial Optimization Problems

In general terms, optimization is concerned with finding a solution to a problem considered "optimal" according to a certain metric. Combinatorial optimization is restricted to finding optimal solutions in a finite solution space with discrete decision variables. Combinatorial optimization plays an important role in various domains, ranging from science to numerous real-world applications. Solving combinatorial optimization problems efficiently has profound implications for cost-reduction in processes, decision-making, and optimization of resources.

This thesis will deal solely with combinatorial problems defined on an n -bit binary string $z = z_1z_2\dots z_n$ with a cost function

$$C : \{0, 1\}^n \longrightarrow \mathbb{R}. \quad (1.1)$$

Many combinatorial optimization problems can be efficiently encoded in this form. Depending on the optimization problem, the goal is to find a bitstring that maximizes or minimizes this cost function. For maximization, the problem can be defined as finding the bitstring z^* that satisfies $z^* = \arg \max_z C(z)$. An important metric to evaluate the quality of a solution z in this context is the approximation ratio, defined as $\eta = C(z)/C(z^*)$. Despite the typically simple formulation of combinatorial optimization problems, dealing with large problem sizes usually poses severe challenges. One key characteristic of these problems is the exponentially large solution space, which contributes to the inherent difficulty of these problems. Despite decades of research, efficient classical optimization algorithms often remain elusive. Quantum computing has gained significant attention as a promising approach to solving these complex optimization problems, which are deemed to be hard to solve for classical algorithms. However, there is still uncertainty if any meaningful quantum advantage can be established over their classical counterparts. It is, therefore, a vibrant field of research. In the following, a few well-known combinatorial optimization problems, which will reappear throughout this thesis in various contexts, will be introduced.

1.1.1 Ising-like Spin Glasses

Spin glasses are a critical domain in statistical and condensed matter physics which concerns disordered magnetic systems. Ising-like spin glasses, which have important connections to combinatorial optimization problems, are of particular interest for this thesis. In general, an Ising spin glass consists of a system with n spins $\sigma_i \in \{-1, +1\}$ and coupling constants $J_{ij} \in \mathbb{R}$, for $i, j \in [n]$. A system is defined via its Hamiltonian, which has the following form for a configuration $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n)$:

$$H(\boldsymbol{\sigma}) = \sum_{m=1}^n \sum_{i_1, i_2, \dots, i_m} J_{i_1, i_2, \dots, i_m} \sigma_1 \sigma_2 \dots \sigma_m. \quad (1.2)$$

Of special interest in this work are Ising Hamiltonians with interactions of maximal order 2 which have the following form:

$$H(\boldsymbol{\sigma}) = \sum_i h_i \sigma_i + \sum_{ij} J_{ij} \sigma_i \sigma_j. \quad (1.3)$$

By relating the binary variables of optimization problems to spins and the constraints to interactions of the Hamiltonian, a correspondence can be established with a wide range of combinatorial optimization problems, including Max-2-SAT, Max-Cut, and the Maximum Independent Set problem.

Moreover, these types of Ising Hamiltonians can be linked to an interaction graph illustrated in Fig. 1.1. A graph (more precisely, an undirected graph) is defined as an ordered pair $G = (V, E)$, where V represents a set of vertices (also referred to as nodes) and $E \subseteq \{\{i, j\} | i, j \in V\}$ is the set of edges. In the following, some terminology will be introduced that will be used throughout this thesis: The degree of a vertex i is defined as the number of edges that are connected to it. A bounded degree graph subsequently is a graph for which the degree of any vertex in the graph can be bounded by some fixed constant natural number. In a regular graph, each vertex has the same degree k , where $k \in [|V| - 1]$. A complete graph is a regular graph with $k = |V| - 1$. A path is a sequence of distinct edges e_k , where two consecutive edges share a joint vertex i . The distance $\text{dist}(i, j)$ between two nodes i, j of a graph corresponds to the minimal number of edges needed to connect the two nodes via a path. A loop is a path that connects a node in the graph with itself. A graph without loops is called a tree. A further very important concept in this thesis is random graphs. A random graph is characterized by the fact that its properties (like the number of vertices and edges) follow a certain probability distribution.

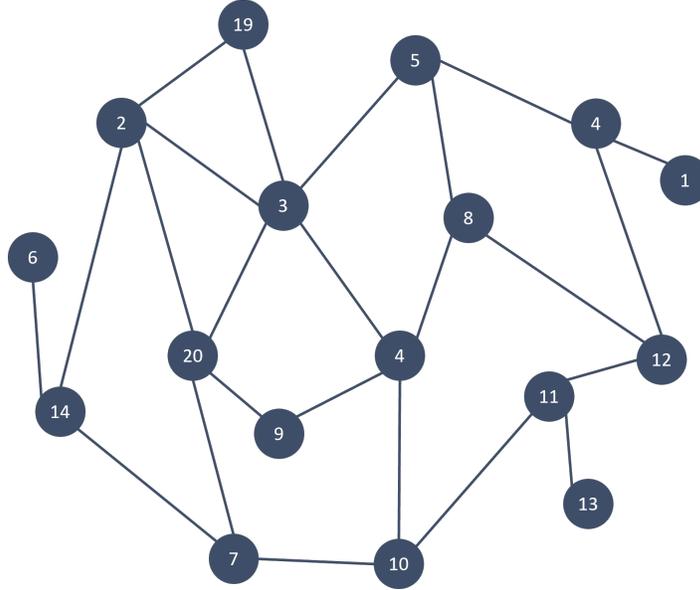


Figure 1.1: Example of a graph. Nodes with their corresponding indices are represented as dark circles. Edges are represented lines connecting these nodes.

1.1.2 The SAT Problem and its Descendants

The satisfiability problem (SAT) is one of the most studied combinatorial optimization problems and was the first problem proven to be NP-complete [19]. Given a propositional logic formula over n variables, the SAT problem examines whether or not an assignment to the boolean variables exists, which satisfies the formula ϕ . The variables x_i in the formula can take the value 0 (false) or 1 (true). A literal l_i is either the variable x_i (positive polarity) or the corresponding negation \bar{x}_i (negative polarity). Writing the formula in conjunctive normal form, one can define the K -SAT problem, which is a SAT problem with K variables per clause. The Max- K -SAT problem asks for the maximal number of satisfiable clauses in an K -SAT formula. A K -clause is of the form

$$c_i = \bigvee_{j=1}^K l_{i_j}, \quad (1.4)$$

where the index j represents the position inside the clause. If m is the number of clauses in a SAT formula, the clause to variable ratio α is defined as:

$$\alpha = m/n \quad (1.5)$$

and the approximation ratio η of an assignment is defined as the ratio of satisfied clauses to the maximum number of satisfiable clauses. Eventually, a SAT formula can be written in the following form:

$$\phi = \bigwedge_{i=1}^m c_i. \quad (1.6)$$

For many algorithms, it is useful or even necessary to encode SAT or Max-SAT problems in terms of a cost Hamiltonian. Finding an assignment to a formula which corresponds to the maximal number of satisfied clauses then corresponds to finding a ground state of the Hamiltonian:

$$H(\phi) = \sum_{i=1}^m \frac{1}{2^{K_i}} \prod_{j=1}^{K_i} (1 - l_{i_j}). \quad (1.7)$$

This expression can be evaluated for a certain assignment of variables when inserting the value of the variable x_{i_j} into l_{i_j} and $1 - x_{i_j}$ if the literal has negative polarity. An assignment with no violated clauses corresponds to a ground state of the Hamiltonian in Eq. (1.8) with zero energy. Therefore, the SAT problem in the Hamiltonian formulation is reduced to the question of the existence of a ground state with zero or higher energy.

Another variation of the satisfiability problem is the so-called weighted Max-SAT problem. A real number, which is called weight, is assigned to each of the clauses. Then, an objective function, which is simply the sum of the weights of all satisfied clauses, is defined. The goal is to find an assignment that maximizes this objective function. The Hamiltonian can easily be rewritten as

$$H(\phi) = \sum_{i=1}^m w_i \frac{1}{2^{K_i}} \prod_{j=1}^{K_i} (1 - l_{i_j}), \quad (1.8)$$

where the w_i is the weight of the corresponding clause.

Formally one can assign the weight of certain clauses to ∞ . The problem then refers to partial weighted Max-SAT. The clauses with weight ∞ resemble hard constraints. A valid solution must satisfy all hard constraints and should be optimized according to the soft constraints (clauses with finite weight). This problem class is very interesting for real-world problems since, in reality, one is often only interested in particular solutions and many assignments might not make sense in a real-world setting.

Another important problem class, that will reappear in many of the numerical studies in this thesis, is the Random-Max- K -SAT problem. In this case, each variable that appears in the SAT formula is drawn with uniform probability and negated with probability $1/2$.

A quick note on the computational complexity of the different satisfiability problems in regard to Sec. 1.2.1. The 2-SAT problem is solvable in linear time [5], whereas k -SAT for $k \geq 3$ is NP-complete [31]. The other mentioned problems (also Max-2-SAT) are all NP-Hard (non-decision version). The decision version of MAX-Sat is also NP-Complete [31, 35].

1.1.3 Max-Cut and Maximum Independent Set Problem

We will not tackle the Max-Cut or Maximum Independent Set with our developed algorithms. However they will appear in certain references throughout the paper, therefore we want quickly mention them. Both of them can be natively defined on a graph like in 1.1. For a given graph the Max-Cut problem is concerned with finding a partition of the nodes into two sets such that the number of edges between the nodes of these two sets is maximal.

An independent set is a set of nodes such that no two nodes of this set are adjacent. The Maximum Independent Set problem aims to find the largest possible independent set in a given graph.

Both problems can be formulated as finding the ground state of a Hamiltonian of the form denoted in (1.10). Whereas the Max-Cut Hamiltonian exhibits no one-point correlations and therefore has a \mathbb{Z}_2 -symmetry.

1.2 Computational Complexity and Algorithmic Hardness

The following chapter contains many parts from [29] in a summarized form, especially Sec. 1.2.3 relies solely on this paper. Put simply, an important goal of combinatorial optimization is to develop algorithms that are able to solve certain problems in a time and resource-efficient way. Depending on the type of the problem and the requirements for the solution, the question of how to measure the hardness of optimization problems emerges. In the following, different approaches in this context are being discussed. First, we will informally introduce a few important classical and quantum complexity classes which will appear throughout the thesis. Subsequently, modern approaches focused on problems with random input (classes of optimization problems where the instances for a particular problem class are drawn according to some probability distribution, e.g. random Max-SAT) are discussed. A new approach to classifying these problems depending on the topological structure of the solution space was developed by David Gamarnik and was dubbed "Overlap Gap Property" (OGP). It has recently been shown that OGP is not only a useful property to examine the algorithmic hardness of classical optimization algorithms, but was also used to identify shortcomings of quantum optimization algorithms [24]. The importance of this framework in connection to quantum algorithms will be presented in Sec. 1.4.

1.2.1 Classical Complexity Theory

The resource we are focusing on in designing combinatorial optimization algorithms is the running time [20]. One can define the different complexity classes in terms of a Turing machine, but for this thesis the complexity is defined more informally as the number of basic operations (defined via fundamental logical gates) acting on a n-bit binary string (input). The scaling behaviour of the running time will then be stated dependent on the number of input variables. Furthermore, when talking about decision problems, we will refer to computational problems which are defined on a n-bit binary string and can be posed as a yes or no question (0 or 1). This said a few complexity classes which will be used throughout the thesis are introduced in the following:

- Polynomial time: The set of decision problems, which can be solved in polynomial running time.
- Exponential time: The set of decision problems, which can be solved in exponential running time.
- NP (nondeterministic polynomial time): The set of decision problems, in which the correctness of a solution to the problem can be verified in polynomial time.
- NP-hard (non-deterministic polynomial-time hardness): Put simply, the set of computational problems which are at least as hard as NP problems in NP, in terms of their running time. This means that a problem in NP can be reduced to a problem in NP-hard in polynomial time.
- Bounded error probabilistic problems: The set of decision problems, in which a solution can be found in polynomial time with probability $p > 2/3$. Through repeated execution of the algorithm, the probability can be increased to be close to one.

From this point on, algorithms that have a polynomial running time will also be referred to as fast algorithms and algorithms that have an exponential running time will be referred to as slow algorithms. It is widely accepted that no fast algorithms exist for a wide range of NP problems. Therefore, considerable efforts are being made to develop polynomial time heuristic algorithms (algorithms that have no guarantee of finding the optimal solution) for these difficult problems.

1.2.2 Quantum Complexity Theory

In essence, to achieve universal quantum computing capabilities, we should be able to perform arbitrary unitary operations on our qubits. Fortunately, any unitary operation U can be broken down into a sequence of gates from a small, fundamental set of gates known as the universal set of gates. Analogous to classical complexity theory, one measures how many basic operations with respect to some fundamental set of gates (consists normally of one- and two-qubit unitary gates) are applied [63]. This will be the quantity of interest for our quantum algorithms to establish a reasonable comparison in terms of running time between classical algorithms and quantum algorithms. We are not really concerned with the different quantum complexity classes since we solely rely on heuristic quantum algorithms in this thesis but for reasons of completeness a few important ones are mentioned here:

- Bounded error quantum polynomial time (BQP): The set of decision problems, in which the solution can be found in polynomial time on a quantum computer with probability $p > 2/3$. This can be seen as the counterpart to the classical BPP problems.
- BQP-hard: The set of problems to which any BQP problem can be reduced in polynomial time.

- BQP-complete: The set of problems that are BQP-hard, but are also in BQP.

It is widely believed that there are no fast quantum algorithms that can solve NP-complete problems. Nevertheless, there are quantum algorithms for which - on certain problems - a better scaling behavior in terms of computational complexity was proven when compared to the best-known classical algorithms [61, 34]. However, this thesis solely deals with heuristic quantum algorithms.

1.2.3 Algorithmic Hardness - An Alternative Approach

Topological Phenomena in the Solution Space

The classical P-NP complexity classes only convey the worst-case performance of finding the optimal solution to a certain problem. For an NP-hard problem, we know that (under reasonable assumptions) we will not find a classical polynomial algorithm that finds the optimal solution for all instances. However, it is worth considering whether there exist other frameworks better suited to tackle specific questions regarding algorithmic complexity. For example, there are optimization problems for which fast algorithms may be able to achieve a certain approximation ratio but seem unable to improve on these ratios. When do random optimization problems which depend on certain parameters become hard to solve on average?

An important strategy for addressing the described problems is based on the idea of investigating the solution space geometry of optimization problems. This approach, initially inspired by research in statistical physics, is nowadays widely used in various areas beyond statistical physics. The basic idea is that the computational complexity of a problem should be reflected in the intricate geometry of the solution space. One of the earliest problems which were investigated by this approach was K -SAT. It connects the algorithmic hardness with the closeness to the satisfiability phase transition threshold. [43, 51]. In the following, we quickly summarize the discussion about the algorithmic hardness of the random K -SAT problem in [29]. Despite the fact that we will not deal with this problem specifically (but with random Max-2-SAT), it will provide a good intuition of how the solution space geometry of many optimization problems in difficult regimes may look like. This knowledge greatly inspired us in the design of our two non-local hybrid quantum-classical algorithms, especially the Quantum Informed Cluster Algorithm, introduced in Sec. 2.3.

It is believed that for all K , there exists a critical clause to variable ratio α_{SAT} . In the thermodynamic limit (number of variables $n \rightarrow \infty$) there is a phase transition, such that random formulas with clause-to-variable-ratios below this critical value are satisfiable, while there exists no satisfying assignment beyond this value with high probability [27]. Algorithms seem to stall at a critical value $\alpha_{\text{ALG}} < \alpha_{\text{SAT}}$ [18]. Work on replica symmetry methods has helped determine the values of α_{SAT} for different K [55, 46]. For example for Max-2-SAT $\alpha_{\text{SAT}} = 1$ [8]. Approaches that focus on the solution space geometry of the problem seem to be helpful descriptions for algorithmic hardness. This includes the so-called weak clustering property which appears at α_{Clust} . Especially for large K this value is in the vicinity of α_{ALG} . Hence it was suggested that this property is intertwined

with the algorithmic hardness of Random- K -SAT [2, 49]. The weak clustering property manifests itself in the following way: One assumes that all but exponentially few (in terms of the total number of variables n) satisfying assignments can be partitioned into subsets, so-called clusters. In these clusters, satisfying assignments exist at a distance of at most $O(1)$ to each other. The subsets are separated by a distance of $O(n)$. In this context we always understand distance in terms of Hamming distance. If all satisfying assignments can be assigned to different clusters one speaks of the strong clustering property. This property occurs for large K . Another topological barrier that was later introduced is the so-called condensation phase [44], for which transition is believed to happen at $\alpha_{\text{ALG}} < \alpha_{\text{COND}} < \alpha_{\text{SAT}}$ for large K . In the condensation phase, the number of solution clusters that cover the majority of the satisfying assignments reduces from an exponential number to just a constant number of clusters. Despite this reduction, the largest cluster still encompasses a non-negligible fraction of all possible assignments for clause-to-variable ratios before α_{COND} . In Fig. 1.2 the different phase transitions and their manifestation in the solution space, depending on the clause-to-variable ratio, can be seen. Frozen variables are another interesting feature of the solution space which emerges for α before reaching α_{COND} in non-vanishing fractions in each cluster, for large K . It has also been theorized that the appearance of frozen variables may be the reason for algorithmic hardness [47]. However, this framework also has shortcomings in describing the algorithmic hardness, for example in the case when K is large.

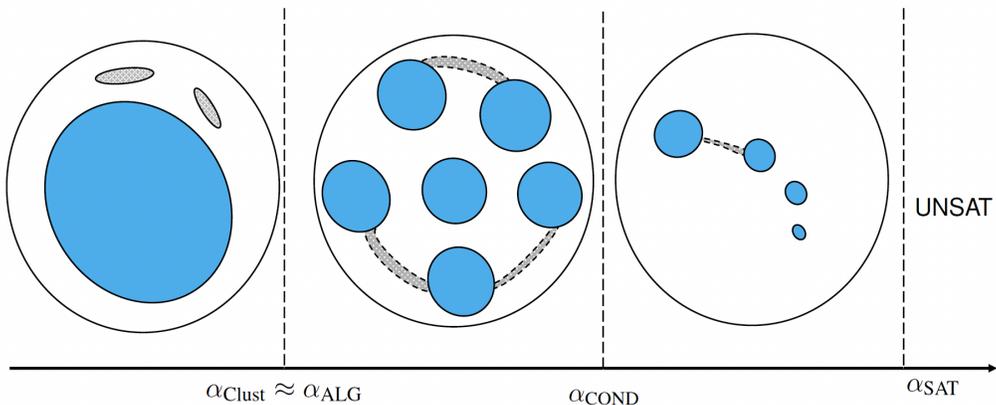


Figure 1.2: Visualization of the solution space geometry of the SAT problem dependent on the clause to variable ratio α . Blue circles illustrate satisfying solution clusters. Grey areas are satisfying assignment, not contained in the clusters. Taken from [29].

The Overlap Gap Property

Recently, a new approach for evaluating the algorithmic hardness of optimization problems named the "Overlap Gap Property" (OGP) has been developed. [30, 2, 49]. In the following, we will stick close to the notation and findings of [29]. OGP shares profound similarities with the clustering property, as both are based on the geometry of the solution space. Let us define an optimization problem over n variables, with σ an element in the discrete solution space Σ_n . A single instance is drawn from some random

structure Ξ_n (for example a random graph) and is denoted as ξ . Then one can define the optimization problem as $\min_{\sigma} \mathcal{L}(\sigma, \Xi_n)$. Subsequently, a metric $\rho(\sigma, \tau)$ is defined which measures the distance (in our case Hamming distance) between two solutions σ and τ . Using this notation the OGP is defined as follows: The optimization problem $\min_{\sigma} \mathcal{L}(\sigma, \Xi_n)$ exhibits the Overlap Gap Property with the values $\mu > 0, 0 \leq \nu_1 < \nu_2$, if for two arbitrary solutions σ, τ which satisfy $\mathcal{L}(\sigma, \xi) \leq c^* + \mu$ and $\mathcal{L}(\tau, \xi) \leq c^* + \mu$ either $\rho_n(\sigma, \tau) \leq \nu_1$ or $\rho_n(\sigma, \tau) \geq \nu_2$, where $c^* := \min_{\omega} \mathcal{L}(\omega, \xi_n)$ is the optimum. Intuitively, OGP tells us that two solutions which are close to optimality are either close to each other or far apart from each other, with no nearly optimal solutions between them. For problems with random structures (e.g. random graphs), one says that the problem exhibits OGP if the drawn instance exhibits OGP with high probability. It should be emphasized that OGP is always formulated for certain problem-dependent parameters μ, ν_1, ν_2 . It is evident that OGP presents challenges when attempting to solve optimization problems. However, this is not only due to the existence of local minima but also due to the specific geometric separation in the solution space (again dependent on the parameters). The manifestation of the OGP in the landscape of the objective function can be seen in 1.3.

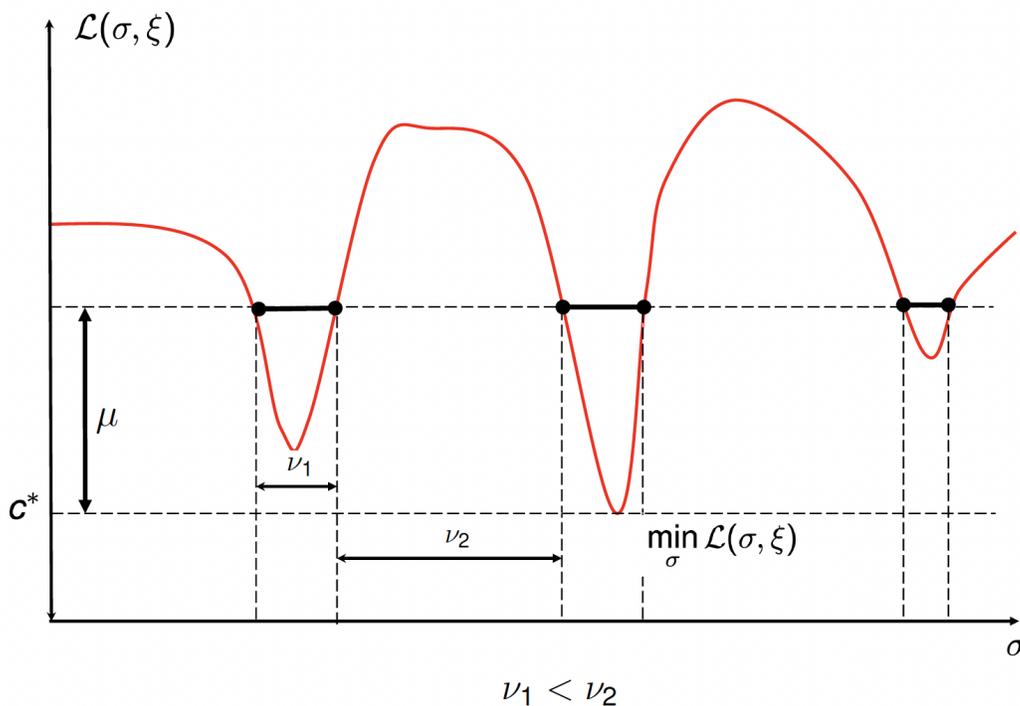


Figure 1.3: The landscape of the objective function $\mathcal{L}(\sigma, \xi)$, which needs to be minimized in the optimization problem, is depicted. Two near optimal solutions are either close (maximal distance ν_1) or far part from each other (minimal distance ν_2). Taken from [29].

Implications of topological barriers on Algorithms

Algorithms have been specially developed in order to overcome the challenges posed due to the emergence of the discussed topological phenomena in the solutions space. One famous example is the survey propagation algorithm [10]. It is an algorithm constructed to solve the random satisfiability problem, with attention to the emergence of clusters. The algorithm iteratively consists of two parts. First, it passes messages which can be thought of as surveys of clusters, while the second part uses the knowledge of these surveys to simplify the problem. The algorithm demonstrates outstanding performance in the clustering phase. Another recently introduced algorithm, which deals with complex energy functions of combinatorial optimization problems (explicitly, the emergence of frozen variables) is a quantum-inspired nonequilibrium Monte Carlo algorithm [50]. This algorithm learns important geometrical features of the cost function on a per-instance basis. This information is used to create spatially-inhomogeneous thermal fluctuations, that are intended to unfreeze variables at different length scales. The algorithm is applied to random K -SAT and Quadratic Assignment Problems and outperforms state-of-the-art solvers for these problems.

Both of the aforementioned algorithms contain many interesting concepts that have the potential to enhance other algorithms, preventing them from suffering critical slowing down when dealing with complex energy landscapes. Especially the second algorithm shares significant similarities with quantum algorithms. Such algorithms could be utilized to inspire the creation of non-local subroutines for incorporation into quantum algorithms. By doing so, some of the limitations faced by local quantum algorithms could be overcome.

OGP is very helpful on the one hand many optimization problems exhibit this property, like the Maximum Independent Set (MIS) problem, the Number Partitioning problem, or finding the ground state of p -spin models. On the other hand, a large variety of local algorithms (classical and quantum) can be ruled out due to the concrete predictions of OGP. Again speaking very naively OGP can rule out many local algorithms because they have difficulties bridging the gap between clusters of near-optimal solutions. The implication of OGP on the local quantum algorithm QAOA will be discussed in Sec. 1.4.3.

1.3 Classical Algorithms

In this section, several classical algorithms are presented which will continue to reappear during the course of this work. The Branch and Bound solver and the well-known simulated annealing method in particular will be very important. This is because our newly developed quantum algorithms are inspired by these classical solvers and incorporate similar methods and subroutines. Nevertheless, a few other algorithms are explained. Either because they also seem to be well-suited to be incorporated into a similar framework as our quantum-classical algorithms, or they may be used as benchmarks or for purposes of illustration.

1.3.1 Max-SAT/Branch and Bound Solver

For many Max-SAT algorithms, there are extensions that are applicable to weighted Max-SAT and sometimes also partial weighted Max-SAT. However, we will solely focus on the Max-SAT problems and make on rare occasions remarks if our developed techniques could be straightforwardly extended. Max-SAT solvers can be broadly classified into two categories. The first category comprises complete/exact solvers that offer optimal solutions. The second category encompasses approximation and heuristic algorithms. Approximation algorithms yield near-optimal solutions and offer guarantees about the solution's optimality, while heuristic algorithms are typically faster but do not provide such guarantees. Heuristic solvers are mostly based on local search algorithms. Two prominent representatives are WalkSAT and GSAT. For Max-2-SAT (the problem class we are mostly interested in) the algorithm with the best approximation ratio guarantee was introduced by Goemans and Williamson [32] with an approximation ratio of 0.878 times the optimal value. For Max-SAT, it has been shown that under the very believable assumption that $P \neq NP$ the best achievable approximation ratio is $7/8$ [42]. Complete Max-SAT solvers again can be roughly divided into branch and bound (BnB) solvers [1], SAT-based algorithms [28] and integer linear programming (ILP) algorithms [13]. From here on we will solely focus on the branch and bound solver, since this concept will be essential for our quantum-classical algorithm. We based our notation on the notation in Sec. 1.1.2 and made some slight changes. The used notation is very commonly employed across the BnB solvers community. We define our formula with n variables and m clauses as $\phi = \{c_1, c_2, \dots, c_m\}$. Each clause is defined as set $c = \{l_1, l_2, \dots, l_k\}$. Furthermore, we define an Assignment A as a set of literals. Each literal can only be contained once in the assignment. In the event that variable x_j is assigned a true (false) value then the variable x_i (\hat{x}_i) is added to the assignment. When values are assigned to certain variables, the SAT formula ϕ has to be updated accordingly. Say \hat{x}_i (x_i) is added to A , then all clauses in ϕ where \hat{x}_i (x_i) appears are removed and for all clauses in which the negation appears, only x_i (\hat{x}_i) is removed from the clause. If a clause becomes empty, it is kept in the formula. The empty clauses are unsatisfied and our goal is to end up with an assignment which produces the smallest possible number of empty clauses in our SAT formula. We denote a formula ϕ that is updated according to some partial (or full) assignment as ϕ_A .

Starting from an empty assignment A the branch and bound solver explores the whole solution space through a combination of assignment extensions and backtracking procedures. Unfavourable parts of the solution space are dismissed by a bounding function. The workflow of a branch and bound solver can be visualized in a search tree, where the nodes represent variables of the assignment and the edges their corresponding value. Prior to having a look at the pseudocode of a typical BnB solver, it is necessary to establish a few definitions:

- Inference rules: They are applied at each node and help extend the current partial assignment by fixing the values of certain variables. Therefore, fewer parts of the search tree have to be explored. Additionally, the chosen rules guarantee that the optimal assignment of the current problem and the optimal assignment of the simplified problem provide the same number of unsatisfied clauses.
- Branching: The branching step always happens subsequently to the application of the inference rules. Typically, some heuristic is applied to choose the next variable and the corresponding value to fix. Branching is crucial for reducing the search space and achieving lower runtime by making informed decisions about the variable assignments.
- Lower bound: The lower bound consists of the current number of empty clauses in a partial assignment ϕ_A plus a lower bounded estimation of the minimal number of clauses that have to become empty if the current partial assignment is completed.
- Upper bound: The number of unsatisfied clauses produced by the best assignment found so far.

Inference rules, branching heuristic and the lower bound computation are applied repeatedly and have a big influence on the runtime of the algorithm.

In Alg. 1 the pseudocode of a typical branch and bound solver can be seen.

Algorithm 1 Branch and Bound Solver

Input: formula ϕ with n, m , *termination_condition*
Output optimal assignment A_{best}, UB

```

1:  $A \leftarrow \{\}$ 
2:  $\phi_A \leftarrow \phi$  ▷ initialize empty solution
3:  $UB \leftarrow m$ 
4:  $continue = true$ 
5: while search tree not fully explored do
6:    $\phi_A, A \leftarrow \text{Inference}(\phi_A, A)$ 
7:    $\phi_A, A \leftarrow \text{Branch}(\phi_A, A)$ 
8:    $LB \leftarrow \text{LowerBound}(\phi_A)$ 
9:   if  $LB \geq UB$  then ▷ Check if new solution is better
10:    backtrack()
11:   else if  $|A| == n$  then
12:      $UB \leftarrow LB$ 
13:      $A_{best} \leftarrow A$ 
14:     backtrack()
15:   else
16:     continue
17:   end if
18: end while

```

1.3.2 Metropolis Hastings and Simulated Annealing

The Metropolis-Hastings algorithm is a so-called Markov chain Monte Carlo (MCMC) method [15]. These algorithms are designed to sample from a probability distribution of interest $p(x)$, where x is the state of the system. The Metropolis-Hastings algorithm is defined by its transition probability W_{xy} from a state x to a state y which governs the movement of the algorithm. In general, in order to design an MCMC algorithm, that converges to a unique stationary distribution that coincides with the probability distribution of interest it is sufficient that the algorithm exhibits two conditions:

- Ergodicity: Every state y can be reached by any state x in a finite number of steps.
- Detailed balance: For every pair of states x and y the following relation holds:

$$W_{xy}p(x) = W_{yx}p(y). \quad (1.9)$$

The Metropolis-Hastings algorithm fulfills both conditions and therefore is a suitable algorithm to sample from a given probability distribution $p(x)$. The probability distribution $p(x)$ in this chapter will always be considered to be the canonical probability distribution. The following section will only discuss the simulation of Ising-like spin systems defined via a problem Hamiltonian H as in Eq. 1.2. Nevertheless, a state of spins will be denoted as x or y instead of σ . A pseudocode of the algorithm is shown in Alg. 2.

Algorithm 2 Metropolis Hastings Algorithm

Input: Hamiltonian $H(\cdot)$ with number of spins n , inv. temperature β **Output** some quantity based on sampled states

- 1: Initialize random spin configuration x
 - 2: $i \leftarrow \text{Uniform}(1, n)$ ▷ Choose a site with uniform probability (natural number)
 - 3: $x' \leftarrow \text{Flip}(x, i)$ ▷ Flip spin at site i
 - 4: $\Delta H = H(x') - H(x)$
 - 5: **if** $\Delta H < 0$ **then** Accept state
 - 6: $x \leftarrow x'$ ▷ Accept state if energetically favourable
 - 7: **else if** $\text{Uniform}(0, 1) < e^{-\beta\Delta E}$ **then**
 - 8: $x \leftarrow x'$ ▷ Accept state with prob. $e^{-\beta\Delta E}$
 - 9: **end if**
 - 10: Measure and go back to line 2 unless sampling should stop
-

Theoretically, if the Metropolis-Hastings algorithm is used to sample from very low temperatures, one would get the ground state energy. However, in most scenarios, this proves to be infeasible since the convergence time of the algorithm explodes. The algorithm shows very little movement while simulating a low-temperature state. A heuristic algorithm that is often better suited for this purpose is the so-called simulated annealing [39, 62]. Here, similarly to the Metropolis-Hastings algorithm, an initial spin configuration is created. Randomly chosen spins are then flipped according to the same acceptance criteria used in the Metropolis-Hastings algorithm. However, simulated annealing starts with a certain temperature and gradually reduces this temperature after each update step. Consequently, as time progresses, the likelihood of accepting states with higher energy decreases, leading to the expectation that the final state at a low temperature will exhibit a lower energy level. Different annealing schedules for the decreasing temperature can be chosen. In this thesis, only linear annealing is taken into consideration. The simulations are set to start at an inverse temperature of $\beta_i = 0$ and end up with a final temperature of β_f . Between these two values, the inverse temperatures are evenly spaced over time. The number of iterations determines the distance between two neighboring inverse temperatures. A pseudocode of this algorithm is depicted in Alg. 3.

Algorithm 3 Simulated Annealing (linear schedule)

Input: Hamiltonian $H(\cdot)$ with number of spins n , β_f , number of iterations m
Output final state x , final energy E

- 1: $\beta = 0$
- 2: $\Delta\beta = \beta_f / (m - 1)$
- 3: Initialize random spin configuration x
- 4: $E = H(x)$
- 5: **for** $step = 1$ **to** m **do**
- 6: $i \leftarrow \text{Uniform}(1, n)$ \triangleright Choose a site with uniform probability (natural number)
- 7: $x' \leftarrow \text{Flip}(x, i)$ \triangleright Flip spin at site i
- 8: $\Delta H = H(x') - H(x)$
- 9: **if** $\Delta H < 0$ **then** Accept state
- 10: $x \leftarrow x'$ \triangleright Accept state if energetically favourable
- 11: **if** $H(x) < E$ **then**
- 12: $E = H(x)$ \triangleright Update best energy
- 13: **end if**
- 14: **else if** $\text{Uniform}(0, 1) < e^{-\beta\Delta E}$ **then**
- 15: $x \leftarrow x'$ \triangleright Accept state with prob. $e^{-\beta\Delta E}$
- 16: **end if**
- 17: $\beta = \beta + \Delta\beta$ \triangleright Reduce temperature
- 18: **end for**

The performance of the algorithm is significantly influenced by the chosen annealing schedule. If the temperature is reduced rapidly, there is an increased likelihood of the algorithm becoming trapped in a local minimum.

1.3.3 Parallel Tempering

Parallel tempering is an alternative method that can be used for computing the ground state or states close to the ground state. It proves to be very efficient for optimizing problems with rugged energy landscapes, outperforming simulated annealing in such scenarios [21, 59, 62].

The algorithm simulates m spin systems (replicas) in parallel with temperatures ranging from the lowest temperature T_{\min} to a maximum temperature T_{\max} . In one update cycle, on each replica N Metropolis-Hastings update steps are performed, followed by a certain number of replica exchanges between neighboring replicas x_i and x_{i+1} with probability $p = e^{-(\beta_{i+1} - \beta_i)(H(x_{i+1}) - H(x_i))}$.

1.4 Quantum Approximate Optimization Algorithm

Many optimization problems seem to be intractable for fast classical algorithms. It is hoped that quantum optimization can mitigate some of these shortcomings and provide an advantage over its classical counterpart. However, hardware limitations in terms of the number of qubits, coherence time, and connectivity pose severe challenges to the execution of useful quantum algorithms. Therefore, a significant effort has been put into developing quantum algorithms that can be executed on noisy intermediate-scale quantum (NISQ) devices. One such approach is to combine quantum computing with classical computing. These algorithms are called hybrid quantum-classical algorithms. They often use parametrized quantum circuits where the parameters are variationally updated. One prominent candidate is the Quantum Approximate Optimization Algorithm (QAOA) [25] which will be of utmost importance for this thesis.

1.4.1 Overview

QAOA is designed to solve combinatorial optimization problems on n -bit binary strings $\mathbf{z} = z_1 \dots z_n$ and aims to find bitstrings that minimize (analogous for maximization) a classical cost function $C : \{0, 1\}^n \rightarrow \mathbb{R}$. By transforming the classical bits to Pauli-Z operators, the cost function can be encoded via a quantum operator of the form $H_C = C(Z_1, Z_2, \dots, Z_n)$ which is diagonal in the computational basis. In this thesis, we will focus on Ising-like Hamiltonians (with degree of interaction lower or equal two) defined on an interaction graph. Finding an optimal solution to the optimization problem is then equivalent to finding the ground state of the corresponding problem Hamiltonian. However, much of the presented content can be generalized to higher-order interaction Hamiltonians. These generalizations can however come with some additional challenges. Throughout the thesis, we will comment on these extensions. The considered Hamiltonians are therefore:

$$H_C = \sum_{i \in V} Z_i + \sum_{(i,j) \in E} Z_i Z_j. \quad (1.10)$$

Furthermore, one defines a mixer Hamiltonian which usually has the simple form $H_M = \sum_{i=1}^n X_i$. The QAOA circuit is then defined as an alternating application of the time evolution generated by these two operators. The initial state should be easy to prepare and is often chosen to be $|+\rangle^{\otimes n}$. The quantum state of QAOA with depth p is then defined as an alternating time evolution of these two operators, where the evolution times of the corresponding operators appear as parameters:

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \prod_{i=1}^p e^{-i\beta_i H_M} e^{-i\gamma_i H_C} |+\rangle^{\otimes n}. \quad (1.11)$$

The $2p$ parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ are then determined by a classical optimization routine so that $\langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | H_C | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle$ is minimized. The minimum of this expectation value converges monotonically to $\min_{\mathbf{z}} C(\mathbf{z})$ for $p \rightarrow \infty$. However, increasing the depth of QAOA comes

with some serious challenges. On the one hand, finding the optimal parameters in a high dimensional parameter space is a computationally difficult task on its own [7]. Several contributions have been made in order to develop heuristic methods to find good parameter configurations [65, 60]. On the other hand, increasing the circuit depth evidently comes with higher requirements on the quantum hardware, which might be incompatible with NISQ devices.

1.4.2 Locality

QAOA is a local algorithm, meaning that the unitary gates of the algorithm connect only variables that are also connected on the interaction graph [38, 24]. The local structure of the algorithm has important consequences and turns out to be a limiting factor for the performance of various types of problems. In the following, a few important locality properties of QAOA are established, which mainly follow [24]. They are important ingredients for proving limitations of QAOA. Moreover, they also have to be kept in mind when designing new algorithms based on QAOA, see Sec.2.

We take into account the QAOA with initial state, mixer and the Ising-like cost Hamiltonian defined in Sec. 1.10, associated with an interaction graph with the usual metric $dist(i, j)$ to measure the distance between two vertices of the graph. To get an understanding of the locality of QAOA, it is instructive to have a look at the expectation value of an operator O which acts on a subset of the vertices of an interaction graph with respect to the QAOA evolved state $|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = U(\boldsymbol{\gamma}, \boldsymbol{\beta})|+\rangle^{\otimes n}$. Then the operator $U^\dagger(\boldsymbol{\gamma}, \boldsymbol{\beta})OU(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is only supported on vertices that have a limited distance in regard to the vertices on which the operator O acts non-trivially. These vertices are called the associated p -neighborhood of the operator O . This neighborhood includes all vertices $i \in V$ of the graph for which $\min_{j \in \text{supp}(O)} dist(i, j) \leq p$. Therefore, when calculating the expectation of an operator O , only the p -neighborhood of O plays a role. The expectation is unaffected by changes in the cost function outside of this neighborhood.

Another interesting property of QAOA, which is now relatively easy to see, is that two operators O_1 and O_2 , which act on subsets of vertices on the interaction graph that are far apart from each other, are uncorrelated. More precisely, if the p -neighborhood of O_1 and the p -neighborhood of O_2 are disjoint, it follows that:

$$\langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | O_1 O_2 | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle = \langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | O_1 | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle \langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | O_2 | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle. \quad (1.12)$$

1.4.3 Limitations

Recently, a lot of research has been conducted, that highlights the limitations of QAOA and also, more generally, deals with limitations of local quantum algorithms [38, 22, 24, 11, 12, 16]. The locality properties of QAOA are a critical element in demonstrating its limitations, particularly when it comes to addressing problems with topological barriers in the solution space. In this context, the Overlap Gap Property has emerged to be a useful approach to assessing the algorithmic complexity and limitations of local quantum algorithms. To gain insight into the limitations of QAOA, we briefly examine two distinct scenarios developed by Farhi et al. [22, 24]. The first limitation concerns the Max-Cut problem with n variables on random d -regular bipartite graphs [22]. The approximation ratio of QAOA for random d -regular graphs consists only of a linear combination of the expectation values of the different edges. This is easy to see from the Hamiltonian in Sec. 1.1.3 and the linearity of the expectation value. These expectation values only depend on their corresponding p -neighborhoods around the edge and it can be shown that if p doesn't scale at least like $\log(n)$, nearly all of these neighborhoods are trees. This, in combination with some graph theory, can be used to show that QAOA with $(d-1)^{2p} < n^A$, for an arbitrary $A < 1$, can only achieve an approximation ratio of $1/2$ on bipartite random d -regular graphs for large d . Another performance constraint of QAOA pertains to its application in solving the MIS problem on random d -regular graphs with n vertices and $dn/2$ edges, while d is a constant and n is going to be large. The emergence of the Overlapping Gap Property on these graphs prevents QAOA from achieving better results than finding 0.854 times the maximal independent set if the depth p grows less than a d -dependent constant time $\log(n)$ for large d (due to its locality) [24].

Chapter 2

Quantum Informed Algorithms

2.1 Recursive-QAOA

In light of the limitations of QAOA caused by its local structure [38, 11], Bravyi et al. introduced the so-called Recursive-QAOA (RQAOA). The algorithm is vital for this work since it served as a source of inspiration for the development of the Quantum Informed Branch and Bound (QI-BnB) Solver Sec. 2.2. This is why, we start this section with a detailed explanation of the algorithm, including its key properties and relevant findings from the literature. This includes its performance as compared to QAOA, its simulation complexity, and a thorough discussion of the recursive problem update, performed by the algorithm. After discussing possible modifications, an extensive numerical study is conducted on the Max-2-SAT problem with instances of almost 200 variables. The algorithm's efficacy is explored on a new problem class and the goal is to gain a more comprehensive understanding of the algorithm's strengths and weaknesses. This helps facilitate its modification and improvement. This then allows us to build similar hybrid quantum-classical algorithms that pursue a related strategy of recursively reducing the problem size and enforcing constraints on the solution.

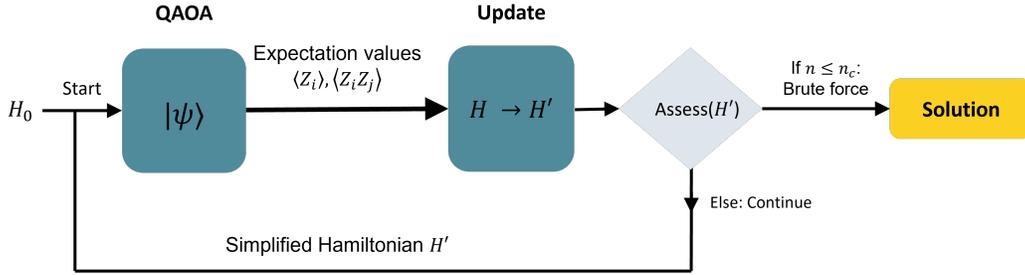


Figure 2.1: Schematic of the RQAOA. Starting with an input problem in form of a Hamiltonian H_0 . Recursive problem reduction and variable elimination, visualized via closed circuit. When the problem size n reaches the threshold n_c the remaining problem is solved by brute force.

2.1.1 Algorithm

The algorithm performs iterative update steps, in which it optimizes the variational parameters of a QAOA circuit according to the current problem Hamiltonian. This information is used to enforce a constraint on the prospective solution and reduces the problem via the elimination of one variable. Recursive application of this update step reduces the problem size until a specified threshold of remaining variables n_c is attained. The remaining problem can be solved by a classical solver (in the algorithm below by brute-force). A schematic of this procedure is illustrated in Fig. 2.1.

In the following, we will refer to operators of the form $\prod_{i=1}^K Z_i$ as K -point correlations. RQAOA was first applied to the \mathbb{Z}_2 -symmetric Max-Cut problem [11], whose Hamiltonian solely incorporates two-point correlations. The Max-2-SAT problem, which we are concerned with, also incorporates one-point correlations. These one-point correlations do not appear in the Max-Cut. This makes Max-2-SAT an interesting candidate for a performance investigation of RQAOA. After optimizing the QAOA state in each update step, the expectation values of all one-point correlations and all two-point correlations of the Max-2-SAT problem are calculated. A constraint is imposed on the correlation with the highest absolute expectation value. Depending on the sign of this expectation value, the constraint demands that this correlation is either perfectly correlated (+1) or anti-correlated (-1). As a consequence of enforcing the correlation constraint, one problem variable is eliminated. In the following, we will call this highest absolute expectation value of all one- and two-point correlations in an update step *rounding value* and the corresponding variables the *rounding variables*. The elimination step is akin to rounding fractional correlations in linear programming relaxation problems.

A generalization to Ising-like Hamiltonians with an arbitrary degree of interaction exists [11]. However, Hamiltonians with K -point correlations, with $K \geq 3$, come with an additional obstacle, since the degree of interaction can increase via the elimination step. Assuming a Hamiltonian of the form Eq. (2.1) a pseudocode of RQAOA is shown in Alg. 4 and the manifestation of the update step in the interaction graph is shown in 2.2.

Algorithm 4 RQAOA

Input: Hamiltonian $H_n = \sum h_i Z_i + \sum J_{m,n} Z_m Z_n$ with n variables,
brute force threshold n_c , QAOA depth p

Output solution S , energy E

- 1: $FrozenCorrel \leftarrow \{\}$ ▷ Container for enforced constraints for solution reconstruction
- 2: $V, E \leftarrow \text{CreateGraph}(H)$ ▷ Creating interaction graph
- 3: **for** $k = 1$ **to** $n - n_c$ **do** ▷ Starting recursive problem reduction
- 4: Prepare QAOA state with optimal parameters:
 $\beta^*, \gamma^* = \arg \min_{\beta, \gamma} (\langle \psi(\beta, \gamma) | H | \psi(\beta, \gamma) \rangle)$ ▷ $(\beta, \gamma) \in [0, 2\pi)^{2p}$
- 5: Store correlations in $M \in \mathbb{R}^{n \times n}$, initialize $M_{ij} = 0$:
 $\forall i \in V : M_{ii} = \langle \psi(\beta^*, \gamma^*) | Z_i | \psi(\beta^*, \gamma^*) \rangle,$
 $\forall (i, j) \in E : M_{ij} = \langle \psi(\beta^*, \gamma^*) | Z_i Z_j | \psi(\beta^*, \gamma^*) \rangle.$
- 6: $(i^*, j^*) = \arg \max_{(i \in V \text{ and } i=j) \text{ or } (i,j) \in E} |M_{ij}|$
- 7: $FrozenCorrel.append((i^*, j^*), sign(M_{i^*j^*}))$
- 8: **if** $i^* == j^*$ **then**
- 9: $H \leftarrow \text{Repl. } Z_{i^*} \text{ in } H \text{ with } sign(M_{i^*i^*}) \cdot \mathbb{1}$ ▷ One-point correlation rounding
- 10: **else**
- 11: $H \leftarrow \text{Repl. } Z_{i^*} \text{ in } H \text{ with } sign(M_{i^*j^*}) \cdot Z_{j^*}$ ▷ Two-point correlation rounding
- 12: **end if**
- 13: $V, E \leftarrow \text{CreateGraph}(H)$
- 14: **end for**
- 15: $S_{n_c} \leftarrow \text{BruteFore}(H_{n_c})$ ▷ Calculate optimal solution by brute force
- 16: $S \leftarrow \text{Reconstruct}(S_{n_c}, FrozenCorrel)$ ▷ Reconstruct complete solution
- 17: $E \leftarrow \text{CalcEnergy}(H, S)$ ▷ Calculate energy of final solution

Relation to QAOA

Numerical simulations show that RQAOA exhibits a significant improvement in performance on the Max-Cut problem when compared to QAOA, and even demonstrates comparable performance to widely used classical algorithms such as the Goemans and Williamson algorithm. Analytical results of the superiority of RQAOA compared to QAOA exist for very special problem classes like the so-called ring of disagrees [11]. Although the improvement is impressive, it is important to note that the computational complexity is also higher. While the classical update steps of enforcing constraints are negligible computational-wise, the process of finding the optimal variational parameters of the circuit must be repeated for each elimination step. However, RQAOA and QAOA have the same circuit size for the same depth p . Therefore no additional quantum hardware requirements are necessary, while for QAOA (and many other variational quantum algorithms) increased performance is normally realized via deeper quantum circuits. It is known that the performance of QAOA monotonically increases with the depth of the circuit p and a plethora of research has been conducted on how these improvements materialize in numerical simulations [4, 53, 3]. It is reasonable to also expect an increasing performance of RQAOA for higher depths of QAOA since the quality of the low-energy states provided to inform the elimination steps is improved. Since to the best knowledge of the author no experiments with higher depth RQAOA

have been published, we perform such in Sec. 2.1.2.

Classical Simulation Complexity

Considering problems with general Ising-like Hamiltonians with n variables of the form

$$H_c = \sum h_i Z_i + \sum J_{m,n} Z_m Z_n. \quad (2.1)$$

it was shown in [11, 54] that the expectation values of Z_i and $Z_m Z_n$ according to the depth $p = 1$ QAOA state $|\psi(\beta_1, \gamma_1)\rangle = e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} |+\rangle^{\otimes n}$ can be calculated in $O(n)$ time via the simple analytic expressions in Eq. 2.2, 2.3 (where all coupling constants which don't appear in the Hamiltonian appear in the formula as $J_{ij} = 0$). Since the cost Hamiltonian exhibits $O(n^2)$ terms, the computational complexity of simulating the calculation of the correlation matrix M in Alg. 4 is $O(n^3)$. Therefore, assuming $O(n)$ update steps this yields an overall complexity of $O(n^4)$ for the full RQAOA simulation.

$$\langle Z_i \rangle = h_i \sin(2\beta) \sin(2\gamma h_i) \prod_{k \neq i} \cos(2\gamma J_{ik}) \quad (2.2)$$

$$\begin{aligned} \langle Z_i Z_j \rangle &= \frac{1}{2} \sin(4\beta) \sin(2\gamma J_{ij}) \\ &\left(\cos(2\gamma h_i) \prod_{k \neq i, j} \cos(2\gamma J_{ik}) + \cos(2\gamma h_j) \cdot \prod_{k \neq i, j} \cos(2\gamma J_{jk}) \right) \\ &- \frac{1}{2} \sin(2\beta)^2 \cdot \left[\cos(2\gamma(h_i + h_j)) \prod_{k \neq i, j} \cos(2\gamma(J_{ik} + J_{jk})) \right. \\ &\quad \left. - \cos(2\gamma(h_i - h_j)) \cdot \prod_{k \neq i, j} \cos(2\gamma(J_{ik} - J_{jk})) \right] \end{aligned} \quad (2.3)$$

It is important to realize the difference in the measurement processes between RQAOA and standard QAOA, as discussed in Ref. [11]. The process of finding the optimal parameters according to the expectation value of the cost Hamiltonian only requires measurements with a few qubits involved (equal to the degree of the highest interaction of the Hamiltonian) for both algorithms. However, after the optimal parameters are found, QAOA samples classical bitstrings from the quantum state for which measurements with all qubits involved are required. This makes QAOA exponentially difficult to simulate for all depths p [26]. This last sampling step is replaced by a classical update step in RQAOA. Hence, using depth $p = 1$ QAOA as state preparation for RQAOA, allows us to simulate the algorithm efficiently.

Naturally, the question arises why this should be a quantum algorithm of interest, when it can be efficiently simulated classically. The reason for this is that the situation changes for higher-depth QAOA circuits with $p \geq 2$. For these, no way to compute the

expectation values of one- and two-point correlations classically in polynomial time is known. The same is true for $p = 1$ when calculating the expectation value of operators with a degree of interaction of 3 or higher. Therefore, these cases may lead to an algorithm that provides quantum advantage.

Elimination Step and Non-Locality

RQAOA has mainly been applied on the Max-Cut problem [11, 12, 6] for which the corresponding Hamiltonian exhibits a \mathbb{Z}_2 -symmetry and is missing on one-point correlations Z_i . Therefore, rounding one-point or two-point correlations in the update step has significantly different implications for the topology of the updated problem graph. While rounding one-point correlations can be seen as a local procedure, rounding two-point correlations is non-local. There are different definitions of local algorithms [38, 16]. We will refer to local quantum algorithms if the applied unitary gates only connect vertices that are connected on the interaction graph, as is the case in QAOA. We will define the locality in the context of our classical elimination step in regard to their implication on the quantum part of the algorithm. The classical elimination step will be called non-local if it can lead to the connection of vertices in the interaction graph which were previously not connected. If this is not the case, the update step will be called local. With this definition and a look at Fig. 2.2 it becomes evident that the one-point correlation rounding is local and the two-point correlation rounding is non-local. Since RQAOA was developed with the intention of counteracting the locality of QAOA, it becomes a striking question if the good performance of RQAOA extends to problems with Hamiltonians of the form of Eq. (1.10) like Max-2-SAT which incorporates one-point correlations.

As discussed in Sec. 1.4.2, measuring the expectation value of an operator with a p -level QAOA state only includes vertices in the p -neighborhood of the operator's support. Additionally, it can be seen in Fig. 2.2 that correlation rounding only affects vertices and edges of the graph in the 1-neighborhood of the support of the considered operator (independent of the depth p). Based on these observations and a revisiting of the locality properties of the QAOA state discussed in Section 1.4.2, we arrive at the following conclusion: Optimizing the QAOA state $|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$ and performing an RQAOA update step leaves the expectation values of all operators which lie in the complement of the $(p + 1)$ neighborhood of the vertices which were involved in the correlation rounding unaffected with respect to the optimized QAOA state. After each update step, RQAOA again optimizes the parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$. Therefore, expectation values of operators which lie outside the $(p + 1)$ neighborhood of the vertices that were involved in the correlation rounding of the previous update step will also change. However, if these changes in the parameters after the update step are small, the change in the expectation value will also be small. This follows from the fact that the expectation values $\langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | Z_i | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle$ and $\langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | Z_i Z_j | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle$ as a function of $\boldsymbol{\gamma}, \boldsymbol{\beta}$ from $\mathbb{R}^{2p} \rightarrow \mathbb{R}$ are continuous. For level $p = 1$ this can be explicitly seen in Eq. (2.2) and Eq. (2.3) where the expectation values are a composition of analytical functions and therefore analytical. In fact, our simulations of the evolution of the parameters during the course of RQAOA in Sec. 2.1.2 suggest that the change in parameters per update step is relatively small.

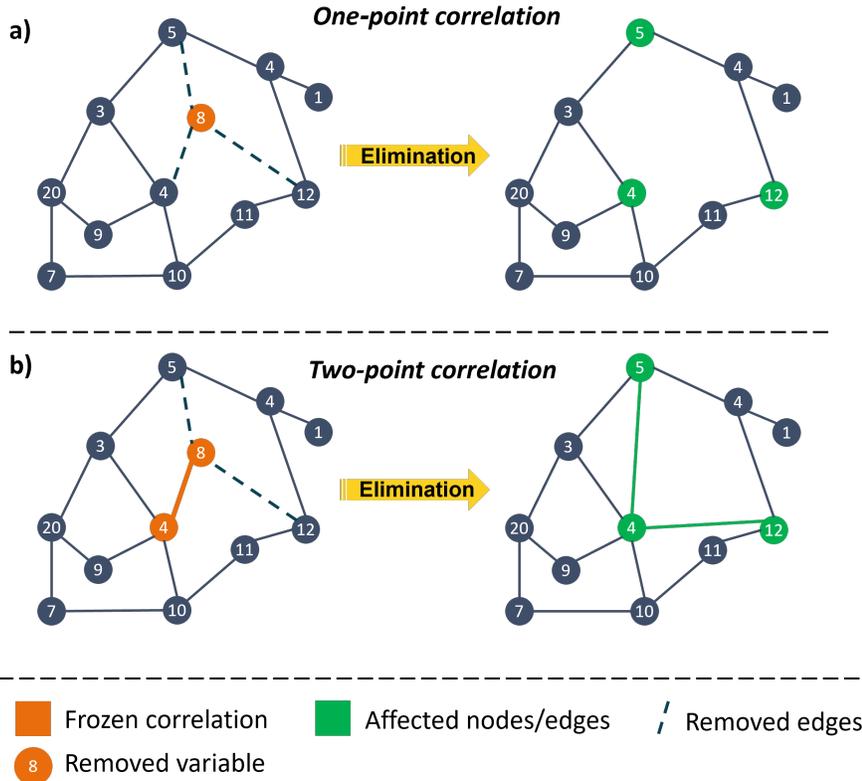


Figure 2.2: Manifestation of the RQAOA elimination step in the interaction graph. a) One-point correlation rounding on the rounding variable 8. b) Two-point correlation rounding on the rounding variables 4 and 8. Variable 8 is chosen to be removed.

Update rule modification - RQAOA-G

The discussion in Sec. 2.1.1 about the locality properties of QAOA and the change in parameters after each update step raises a question. Can similar performance be achieved with RQAOA when in each update step more variables are eliminated and therefore the QAOA circuit needs to be optimized less often? Based on this discussion we propose RQAOA with a modified update rule in which a group of variables is eliminated in each update step - we coin this algorithm RQAOA-G.

The modified update rule of RQAOA-G is a simple modification of the one in RQAOA. After optimizing the QAOA parameters, the expectation values of all correlation operators defined on the graph are stored in the matrix M (like in RQAOA). The one correlation with the maximum absolute expectation is picked and a constraint is enforced according to the RQAOA elimination step. Now, instead of retraining the variational parameters, the $p+1$ neighborhood of the graph, associated with the rounding variables is going to be disregarded during the rest of this update step. The correlation on the remaining graph with the largest absolute expectation value, stored in M , is rounded again. This procedure of rounding and disregarding the corresponding $p+1$ neighborhood is repeated until a lower grouping threshold n_g of remaining variables on the

graph is reached. The full graph according to the updated Hamiltonian is established again and the next update step starts with optimizing the QAOA parameters. The full RQAOA-G algorithm is described in Alg. 5.

The specific way of choosing on which correlations a constraint should be enforced is reasoned in a way that all the correlations which are chosen to be rounded in one update step don't influence the expectation values of each other if the parameters of the QAOA circuit are not retrained. This leads to a significant reduction of complexity. Imagining a d -bounded graph with n variables, RQAOA needs to optimize the QAOA parameters exactly n times (for $n_c = 0$). Since RQAOA-G eliminates $O(n)$ variables per update step, the QAOA circuit only has to be optimized in order of $O(1)$ times. Numerical simulations to compare RQAOA and RQAOA-G are performed in Sec. 2.1.2.

Algorithm 5 RQAOA-G

Input: problem Hamiltonian $H = \sum h_i Z_i + \sum J_{m,n} Z_m Z_n$, grouping threshold n_g , brute force threshold n_c , QAOA depth p

Output solution S , energy E

- 1: $p' \leftarrow p + 1$
- 2: $FrozenCorrel \leftarrow \{\}$
- 3: **for** $k = 1$ **to** $n - n_c$ **do**
- 4: Prepare QAOA state with optimal parameters:
 $\beta^*, \gamma^* = \arg \min_{\beta, \gamma} (\langle \psi(\beta, \gamma) | H | \psi(\beta, \gamma) \rangle)$ $\triangleright (\beta, \gamma) \in [0, 2\pi)^{2p}$
- 5: $V, E \leftarrow \text{CreateGraph}(H)$
- 6: Store correlations in $M \in \mathbb{R}^{n \times n}$, initialize $M_{ij} = 0$:
 $\forall i \in V : M_{ii} = \langle \psi(\beta^*, \gamma^*) | Z_i | \psi(\beta^*, \gamma^*) \rangle,$
 $\forall (i, j) \in E : M_{ij} = \langle \psi(\beta^*, \gamma^*) | Z_i Z_j | \psi(\beta^*, \gamma^*) \rangle$
- 7: $EliminateList \leftarrow \{\}$
- 8: **while** $|V| > n_g$ **do** \triangleright Starting modified update step
- 9: $(i^*, j^*) = \arg \max_{(i \in V \text{ and } i=j) \text{ or } (i,j) \in E} |M_{ij}|$
- 10: $FrozenCorrel.append((i^*, j^*), \text{sign}(M_{i^*j^*}))$
- 11: $EliminateList.append((i^*, j^*), \text{sign}(M_{i^*j^*}))$
- 12: $V, E \leftarrow \text{Remove_p'_Neighborhood}(V, E, (i^*, j^*))$ \triangleright Disregarding neighborhood
- 13: **end while**
- 14: **for** $(i^*, j^*), \text{sign}$ **in** $EliminateList$ **do**
- 15: **if** $i^* == j^*$ **then**
- 16: $H \leftarrow \text{Repl. } Z_{i^*} \text{ in } H \text{ with } \text{sign} \cdot \mathbb{1}$ \triangleright One-point correlation rounding
- 17: **else**
- 18: $H \leftarrow \text{Repl. } Z_{i^*} \text{ in } H \text{ with } \text{sign} \cdot Z_{j^*}$ \triangleright Two-point correlation rounding
- 19: **end if**
- 20: **end for**
- 21: **end for**
- 22: $S_{n_c} \leftarrow \text{Bruteforce}(H)$ \triangleright Calculate optimal solution by brute force
- 23: $S \leftarrow \text{Reconstruct}(S_{n_c}, FrozenCorrel)$ \triangleright Reconstruct complete solution
- 24: $E \leftarrow \text{CalcEnergy}(H, S)$ \triangleright Calculate energy of final solution

Finally, since we don't expect NP to be a subset of BQP, therefore RQAOA is doomed to fail on certain instances of the Max-2-SAT problem. This naturally raises the question of when and why RQAOA fails. In situations where RQAOA does not produce an optimal bitstring, it must have made a wrong decision, whereby a false correlation was enforced during the update step. Therefore, it is crucial to understand the causes of these incorrect decisions and pinpoint the stage of the algorithm where they occur. This is essential in gaining deeper insights into the flaws of RQAOA and developing improved algorithms, as attempted in [56]. In order to achieve this, we analyze the RQAOA performance on Max-2-SAT problems where it failed to provide optimal solutions, conducting an analysis of the incorrect correlations roundings enforced by RQAOA in Section 2.1.2.

2.1.2 Results

Our interest extends beyond comparing the performance of RQAOA with that of QAOA; we also aim to evaluate the effectiveness of RQAOA in solving Max-2-SAT problems and assess its ability to find optimal solutions. To do this, we obviously need to know the optimal solution of the problem. To achieve this, we utilized RC2, a deterministic Max-SAT solver that took part in the MaxSAT Evaluations competition in 2018 and 2019 and ranked first in two categories [41]. However, since it is a deterministic solver, the running time could still explode for larger instances, therefore the size of the Max-2-SAT problems is limited mostly to 60 variables. All Max-2-SAT problems are analyzed in the unsatisfiable clause to variable ratio regime past $\alpha = 1$, as explained in 1.2.3. This ensures that we are in the regime of difficult Max-2-SAT instances, which are after all the problems of interest. If not stated otherwise, the brute force threshold n_c of RQAOA was chosen to be 6. If not stated otherwise, Despite evaluating the performance of depth $p = 1$ RQAOA, the algorithm is analyzed on instances where it failed to produce an optimal solution. Our final objective is to assess the performance of QAOA and RQAOA for deeper circuits. However, due to the exponential scaling in simulating the algorithms classically, only small instances can be simulated. To infer the potential on actual quantum devices, QAOA is executed for higher depths on a trapped ion quantum computer.

Comparison to QAOA

First, in Fig. 2.3 the performance of RQAOA and QAOA is assessed on the random Max-2-SAT problem. We generated 100 instances of the problem with 60 variables for different clause-to-variable ratios α between 1 and 6. This was done with the intent of analyzing the algorithms over a large domain of the hard instances and seeing how robust they are relative to the density of graphs (high clause-to-variable ratios correspond to denser graphs). It can be seen that throughout all clause-to-variable ratios, RQAOA significantly outperforms QAOA and the approximation ratio of RQAOA is always close to 1. The approximation ratio of QAOA slightly improves for higher clause-to-variable ratios. As a reference: The approximation expectation value of an arbitrary Max-2-SAT problem is 0.75, therefore every reasonable algorithm should produce approximation ratios above this value.

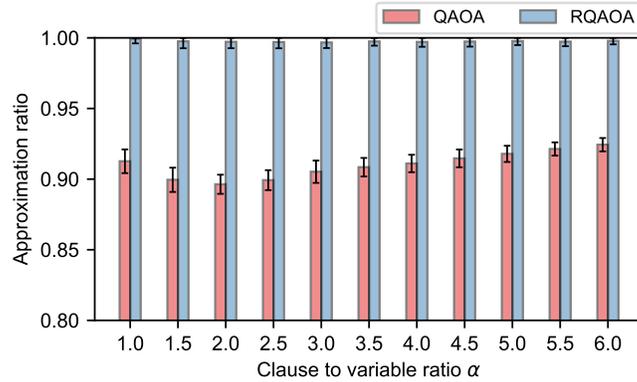


Figure 2.3: Performance of classically simulated level $p = 1$ QAOA and RQAOA evaluated on random Max-2-SAT instances with $n = 60$ variables on different clause-to-variable ratios α . The bars indicate the average approximation ratio for 100 randomly generated Max-2-SAT instances. The complete Max-SAT solver RC2 was used to calculate the optimal solutions and consequently the approximation ratio.

In Figure 2.4, we investigate whether the advantage of RQAOA over QAOA remains valid for larger system sizes of random Max-2-SAT. More specifically, 100 small problems were generated between 40 and 160 variables for each clause-to-variable ratio. The results show that the superiority of RQAOA is consistently maintained for all system sizes and clause-to-variable ratios examined. Since Max-2-SAT problems of this size are usually not feasible for a deterministic solver, the "optimal solutions" are obtained via parallel tempering, details can be found in App. 2.3.4.

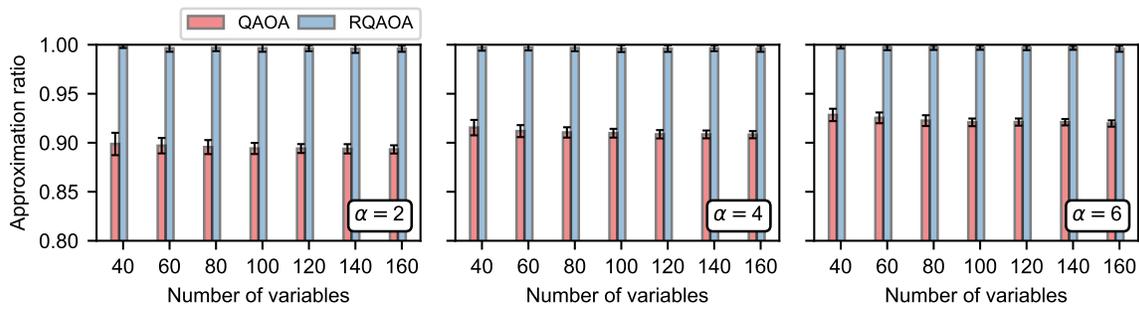


Figure 2.4: Analysis of the scaling behavior of classically simulated level $p = 1$ QAOA and RQAOA in terms of the problem size, evaluated on random Max-2-SAT instances with three different clause-to-variable ratios α . The bars show the average approximation ratio for 100 randomly generated Max-2-SAT instances for each different problem size and corresponding α . The allegedly optimal solutions were obtained with parallel tempering. The bars indicate how close the solutions of (R)QAOA are to the solutions of parallel tempering. Details about the parallel tempering implementation can be found in App. 2.3.4.

Update rule investigation

In this subsection results regarding the updates of RQAOA and their influence on the algorithm are analyzed. First, results of the investigation of instances in which RQAOA failed to achieve an optimal solution and the connection to the correlation rounding are presented. This is followed by results concerning the evolution of the optimal parameters over the course of the complete RQAOA procedure and how they change after each update step.

In Fig. 2.5 we evaluate if there exists a connection between incorrect elimination steps of level $p = 1$ RQAOA, which prevent us from finding the optimal solution, and the corresponding rounding values. Put simply, the question is: are correlation roundings with lower rounding values more likely to be wrong? To investigate this question, we generated random Max-2-SAT problems and for each instance, all optimal solutions were recorded via the RC2 solver. Afterward, we solved the problems with RQAOA and picked 100 instances for which it failed to find an optimal solution. For each of these instances, RQAOA subsequently rounds correlations and incrementally extends the partial assignment. Since RQAOA failed to produce an optimal assignment, there had to be an elimination step, that enforced a correlation on variables, that is incompatible with any of the potential optimal solutions. We label this, let us say, k -th elimination step, as *wrong decision*. We call the set of rounding values up to this k -th step the *wrong decision list*. The plot shows the wrong decisions of RQAOA and their corresponding position in the wrong decision list. A lower percentile corresponds to lower rounding values. It can be seen that wrong decisions are more likely to occur with low rounding values in the wrong decision list. However, this effect becomes less pronounced for increasing clause-to-variable ratios. This will be valuable information for modifying RQAOA and designing similar algorithms.

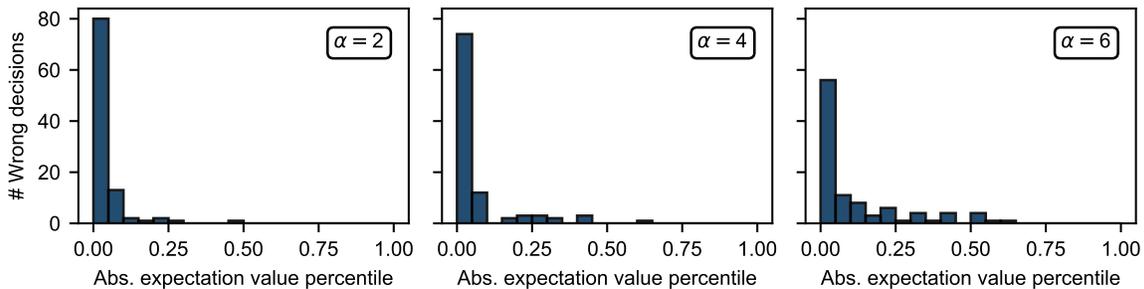


Figure 2.5: For three different clause-to-variable ratios α , 100 random Max-2-SAT instances with $n = 60$ variables and three different clause-to-variable ratios α were generated, on which level $p = 1$ RQAOA ($n_c = 5$) failed to produce the optimal solution. All optimal solutions for a certain instance were determined with the RC2 solver. The x-axis corresponds to the set of rounding values up to the elimination step where the wrong decision (see 2.1.2) happened, while lower rounding values are in the lower percentiles. The bars indicate at which percentile RQAOA made the wrong decision. Low percentiles correspond to lower absolute expectation values.

In Fig. 2.6 the evolution of the optimal variational parameters γ and β during the execution of the level $p = 1$ RQAOA is plotted against the elimination steps. This is done for 4 different Max-2-SAT instances with $n = 60$ variables and increasing clause-to-variable ratios α . In the top row, it can be seen that both parameters remain in a relatively small range of values. This small absolute change of the parameter values relative to the last elimination step can be seen in the bottom row. A clear trend can be observed that for higher α 's the change of parameters becomes even less. The sharp drop in the parameter values for $\alpha = 1$ is a consequence of the circumstance that the problem Hamiltonian becomes trivial $H = 0$. Therefore, the assignment of the remaining variables doesn't matter and the optimal parameters can be chosen to be 0. Overall, it can be seen that the change in the variational parameters is relatively small (compared to the range of possible parameters: $[0, \pi)$). These results clearly support the discussion in Sec. 2.1.2. It can be seen that using the optimized QAOA state to enforce more than one correlation constraint on the variables per elimination step, could lead to an algorithm with similar performance and significantly less computational cost.

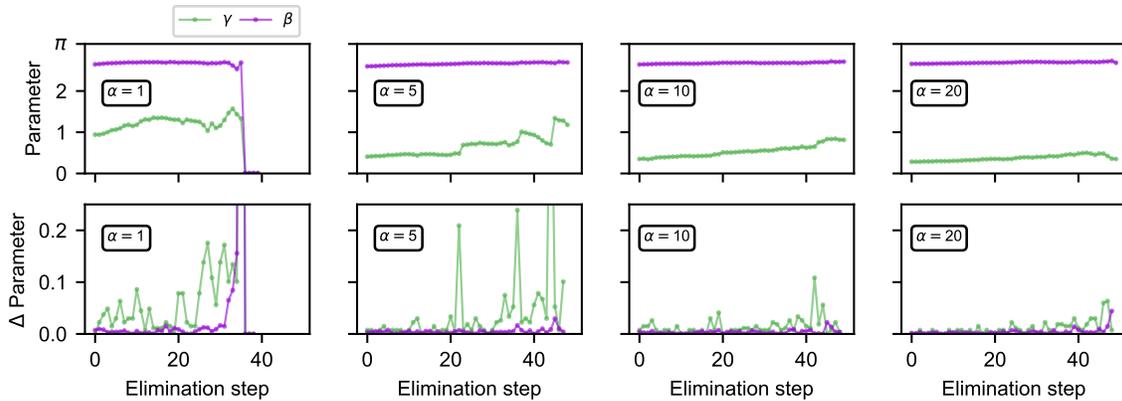


Figure 2.6: Evolution of the optimal variational parameters γ and β for depth $p = 1$ RQAOA with respect to the elimination steps of RQAOA with $n_c = 10$ on four different Max-2-SAT instances with 60 variables and four different clause-to-variable ratios α . The bottom row displays the change in the absolute value of the parameters relative to the previous update step. In the top row, the values of the parameters are plotted with respect to the elimination steps.

In Fig. 2.7 the performance of RQAOA and RQAOA-G with depth $p = 1$ is compared on the Max-2-SAT problem with $n = 60$ variables. It is no surprise that standard RQAOA achieves better results since the parameters are optimized for each update step separately. Nonetheless, the difference is marginal. For the approximation ratio (left bar plot) the difference of the mean values is within each other's error bars. Also, the difference in the number of optimal found solutions (right plot) is small. The performance difference seems to be consistent for the different clause-to-variable ratios α . This shows that comparable performance of RQAOA can be achieved with significantly fewer resources ($O(1)$ QAOA optimizations instead of $O(n)$).

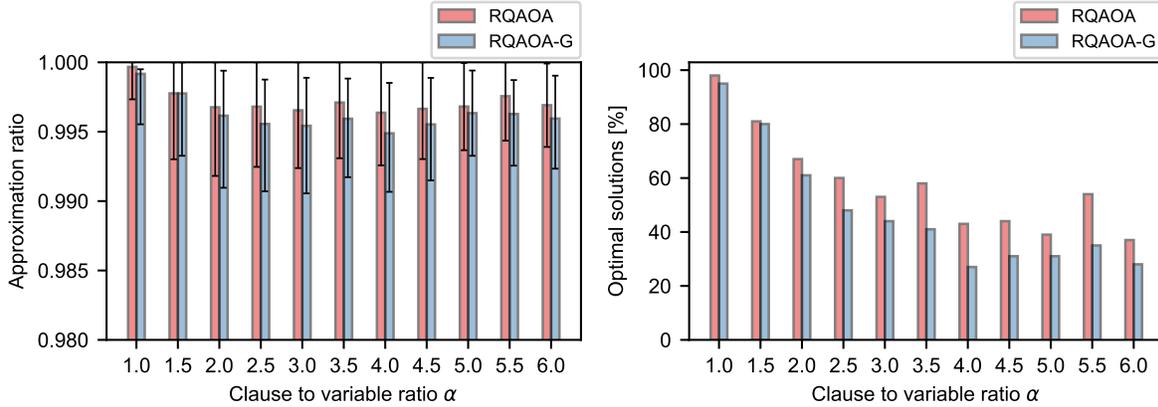


Figure 2.7: Performance of classically simulated level $p = 1$ RQAOA and RQAOA-G on randomly generated Max-2-SAT instances with $n = 60$ variables, at different clause-to-variable ratios α . The average approximation ratio on 100 Max-2-SAT instances per clause-to-variable ratio is represented in the left bar plot and the number of times the solvers were able to find an optimal solution is shown on the right. The RC2 solver was used to determine the optimal solutions.

Higher Depths

We could only perform classical RQAOA simulations with depths $p > 1$ for small system sizes, due to the exponentially scaling computational cost. It turned out that for 100 random Max-2-SAT problems RQAOA with depth $p > 1$ was able to find the optimal solution in all instances. Therefore, 100 instances were created on which the fast $p = 1$ RQAOA simulation failed to find an optimal solution. The performance of RQAOA for different depths was then evaluated on these 100 instances which can be seen in Fig. 2.8. As expected the performance of RQAOA increased with higher depths p . Almost all instances could be optimally solved for $p = 2$ and all instances for $p > 2$.

In Fig. 2.9 we evaluated the increase in performance of QAOA depending on the depth p classically and on a real quantum device, IONQ which is based on trapped ions. The examined problem is once more Max-2-SAT with 11 variables and a clause to variable ratio $\alpha = 2$. As expected, the approximation ratio of classically simulated QAOA improves significantly with depth p . However, the performance of QAOA executed on the quantum device with optimal parameters deteriorates as the circuit depth increases, eventually approaching the approximation ratio of QAOA executed with randomly generated parameters on a classical device. Despite the improvement of classically simulated QAOA and the resulting potential for RQAOA to produce better correlation expectation values, the simulation sheds light on the shortcomings of currently available quantum devices.

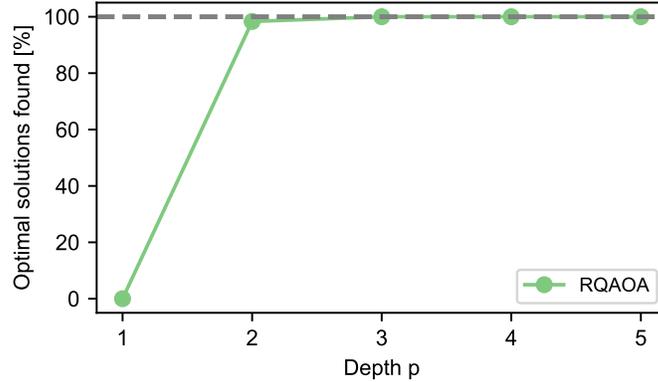


Figure 2.8: Performance of classically simulated RQAOA with $n_c = 1$ on 100 random Max-2-SAT instances with $n = 12$ variables and a clause to variable ratio $\alpha = 16.7$ on which depth $p = 1$ RQAOA failed to find an optimal solution. The percentage of optimally found solutions is plotted against the depth p . The simulations for $p > 2$ were carried out with the open-source software framework PennyLane for quantum computing. The optimization of the variational parameters was performed via the gradient descent method with 1000 optimization steps and 20 random initial starting points for each optimization procedure to increase the likelihood of finding the global minimum.

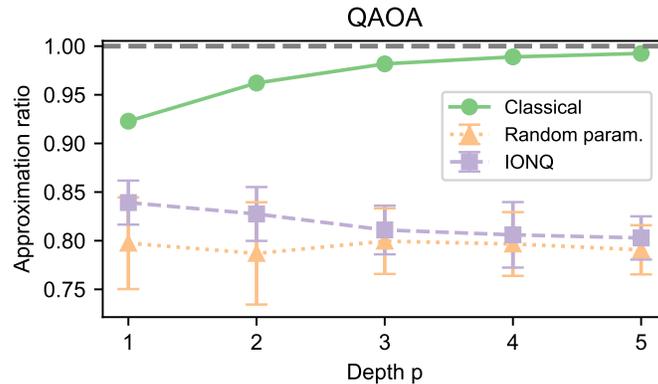


Figure 2.9: The achieved approximation ratios of QAOA on one randomly generated Max-2-SAT instance with 11 variables and a clause-to-variable ratio of 2 plotted against the circuit depth p . Trapped ions-based quantum computer IONQ, which offers all-to-all connectivity, was utilized as an 11-qubit quantum computer through Amazon Braket’s web service. To measure the results, 10 distinct independent runs of QAOA were performed, with 200 shots taken in the measurement processes. IONQ utilized the optimal variational parameters from the classical simulation of QAOA. The “Random param” curve represents the classical simulation of QAOA with parameters generated uniformly at random. The curve was generated by averaging the results of 100 randomly generated parameter sets for each depth p .

2.1.3 Discussion

After reviewing the existing literature on RQAOA, including its working principles, its relationship to QAOA, and the simulation complexity of the algorithm, we conducted a thorough analysis of the update step in RQAOA. The algorithm was initially designed to mitigate the performance limitations of QAOA posed by its locality via the introduction of non-local update steps. Existing studies of RQAOA have primarily investigated problems with Z_2 -symmetry which incorporate no one-local correlation terms [11, 12]. Therefore, we evaluated RQAOA on the Max-2-SAT problem to observe the impact of the local update steps. Our numerical simulations have shown that RQAOA also significantly outperforms QAOA on the Max-2-SAT problem. We conducted further experiments to analyze situations when RQAOA fails to find optimal solutions and to see how the optimal variational parameters of QAOA change over the course of the algorithm. These results, in combination with the locality properties of QAOA, motivated us to propose a modified version of RQAOA that eliminates several variables in each update step. The modified update step ensures that the enforced constraints do not interfere with each other, resulting in performance comparable to that of RQAOA. Finally, the performance of RQAOA is found to improve with the depth of QAOA. However, this behavior could only be shown in problems with small system sizes. Simulations on the IONQ device showed the obstacles the algorithm faces with higher depth QAOA on real quantum hardware where noise prevents the algorithm from improving on the approximation ratio of the problem.

2.1.4 Outlook

There are many interesting follow-up questions that could be pursued. Ultimately we are interested in whether quantum algorithms provide any meaningful advantage over their classical counterparts. Consequently, a particularly promising topic of interest could be a comparative analysis between RQAOA and similar classical methods in order to evaluate the role of the quantum part in measuring the correlations via QAOA. An inherently similar approach in mind would be to calculate classical analogs to the correlation expectation values of RQAOA and use them to recursively solve the problem. The renowned Goemans-Williamson [32] algorithm could be of particular interest in this context. It could also be interesting to compare RQAOA to simple greedy algorithms which also subsequently extend the assignment of the problem and see if a good local update strategy could be able to outperform RQAOA.

Another interesting investigation includes a more thorough analysis of RQAOA for higher depths p . For meaningful experiments, a certain system size is necessary. On the one hand, simulations on real quantum devices would be very interesting. Not only to see how the performance of RQAOA changes with lower-energy states but also to assess the influence of noise on the updates. On the other hand, efficient classical simulations with reasonable system sizes may also provide interesting insights. One strategy to calculate $p = 2$ expectation values could be to start from a classical bitstring. When calculating the expectation values of QAOA, it becomes apparent that the evolution of the cost Hamiltonian on the initial state does not contribute to these

values. This is because a classical bitstring acts as an eigenvector of this operator, and the corresponding eigenvalue cancels out within the expectation value. Therefore we think that it should be possible to derive an analytical formula for this scenario to compute the expectation values for $p = 2$.

Based on the exploration of RQAOA-G, one could extend this avenue in many directions. A more thorough analysis of rounding a group of correlations in each update step, including a scaling analysis, could be carried out. Many different rounding strategies in this context could be explored. Adding stochasticity to the update step would be one such possibility, to name just one example. This would allow for the expectation values of the correlations to inform a probability distribution, instead of making deterministic updates.

2.2 QAOA Informed Branch and Bound Solver

Drawing inspiration from RQAOA [11] and other literature [48] on developing novel quantum optimization algorithms for NISQ devices and combining them with classical optimization techniques, we propose a new hybrid quantum-classical Max-2-SAT algorithm, coined Quantum Informed Branch and Bound (QI-BnB) solver. The algorithm uses information from a low-energy QAOA state and optimization techniques of a classical branch and bound solver to iteratively solve the Max-2-SAT problem. After a discussion about the utilization of the classical optimization techniques and its benefits, the full algorithm is presented. Rigorous numerical simulations for the random Max-2-SAT problem with relatively large problem sizes of up to 160 variables are performed to assess its performance and characteristics. This will be concluded by a discussion, including strategies to enhance the algorithm’s performance, and potential extensions of the algorithm to other combinatorial optimization problems.

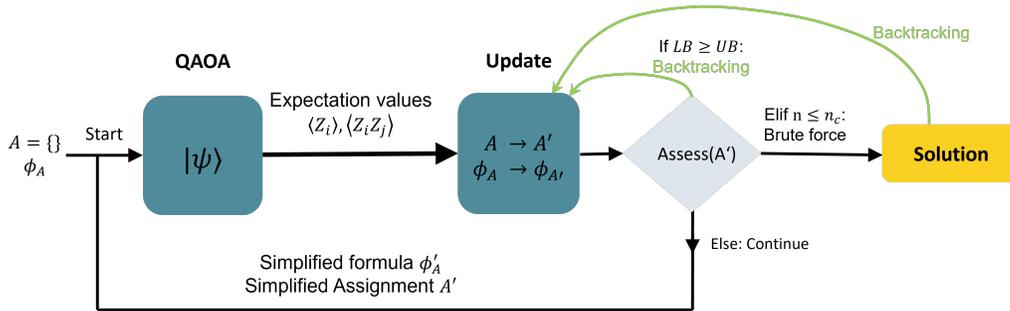


Figure 2.10: Schematic of the QI-BnB algorithm. Starting with input problem in form of Max-2-SAT formula and empty assignment. Recursive problem reduction and variable eliminations, visualized via closed circuit, is executed until lower bound on quality of current partial assignment (LB) is worse or equal to quality of best assignment found so far (UB) and backtracking to a previous, less complete, assignment is performed. Otherwise, if number of remaining variables n reaches threshold n_c , the remaining problem is solved by brute force. Compared to RQAOA the problem is formulated in its native formulation as CNF-formula ϕ .

2.2.1 Algorithm

While RQAOA directly simplifies the corresponding Hamiltonian of a problem, we focus on the update step on the Max-2-SAT problem in its native formulation as a propositional logic formula. This allows us to augment the quantum information with well-studied classical Max-SAT techniques. A schematic of this algorithm can be seen in Fig. 2.10. Utilizing the low-energy QAOA state to measure expectation values and performing update steps in a similar way as in RQAOA, the process of incrementally extending the Max-SAT assignment can be interpreted as traversing through a search tree, analogous to a classical BnB solver. Compared to the classical case the nodes correspond to classical variables but also to correlations (Z_i or $Z_i Z_j$). The edges are associated with their corresponding assigned values. This inspires us to borrow BnB techniques like inference rules, lower bound computations, and backtracking to augment our hybrid algorithm. In general, depending on the complexity of the update

rules, one can distribute the workload between the quantum and classical parts of the algorithm with this scheme. However, the utilized classical techniques require only a little computational cost, which makes it easier to assess the role of the quantum part in the algorithm.

The basic idea of this solver is to optimize QAOA, calculate expectation values of correlations, and - together with classical inference rules - extend the assignment of the problem. Since the update rule is heuristic in nature, obviously unfavorable assignments can occur. Due to this reason, backtracking is implemented to go back to parental parts of the search tree, looking for more favorable solutions. Exploring the whole search tree like in BnB would obviously come at an exponential computational cost. Since we are designing a heuristic algorithm, the idea is to only explore a small part of the search tree which exhibits optimal or near-optimal solutions. To avoid the computational cost of searching in unfavorable regions of the search tree and in order to detect prospective bad solutions early, lower-bound computations are incorporated.

Update Step

As already mentioned, in RQAOA the update steps focus on the problem Hamiltonian. Since QI-BnB performs problem-specific update steps, the enforcing of constraints happens in the native formulation of the problem as propositional logical formula ϕ . The notation of the algorithm in terms of variables, literals, clauses, and the propositional logic formula remains consistent with the one described for classical BnB solvers in Sec. 1.3.1. However, the assignment set differs in that it consists of tuples of the form $((i, j), s)$, with $s \in \{-1, +1\}$. The first entry (i, j) corresponds to the variable indices on which the correlation constraint is enforced, while $i = j$ corresponds to a one-point correlation constraint and $i \neq j$ to a two-point correlation constraint. The value of s indicates which constraint is enforced on the corresponding variables (perfect correlation or anti-correlation). The notation is analogous to the one used in RQAOA in Alg. 4. Analogous to the classical BnB solver notation, ϕ_A is the updated formula according to the partial assignment. Adding a tuple $((i, j), s)$ to the assignment A , the formula gets updated in the following way:

- $i = j$: For $s = 1$ (-1) all variables x_i in the formula are set to true (false) and the formula is updated analogously to Sec. 1.3.1, meaning all clauses containing x_i (\bar{x}_i) are removed and in all clauses containing the negation, only \bar{x}_i (x_i) is removed from the clause.
- $i \neq j$: If $s = 1$ all variables x_j , respectively \bar{x}_j in the formula are replaced by x_i , respectively \bar{x}_i . If $s = -1$ the update is identical, while the variables with index j are replaced by the variable with index i with opposite polarity.

This notation becomes necessary since the update step in the QI-BnB solver can also enforce constraints on two-point correlations. The update step consists of two parts. First, a set of inference rules is applied. As a reminder to Sec. 1.3.1 these inference rules exhibit very low computational cost and extend the assignment in a way, in which

the optimal solution of the simplified formula violates the same number of clauses as the optimal solution of the formula before applying the inference rules.

The inference rules incorporated in the QI-BnB solver were proposed in [45] and partly in [1]:

- The pure literal rule [9]: If a literal exclusively occurs with either positive (negative) polarity, then the value of the corresponding variable is set to true (false).
- The almost common clause rule [9]: If a MaxSAT instance includes two clauses, $\bar{x}_i \vee l_j$ and $\bar{x}_i \vee l_j$, then both clauses can be replaced with just l_j . If neither x_i nor \bar{x}_i appears in the formula anymore, the variable can be set arbitrarily either to true or false.
- The complementary unit clause rule [52]: If a clause solely contains the variable x_i and another clause only contains \bar{x}_i then these two clauses are removed. Again, if neither x_i nor \bar{x}_i appears in the formula anymore, the variable can be set arbitrarily either to true or false.
- The dominating unit clause rule [52]: If the total count of clauses, regardless of their length, that contains a variable x_i (\bar{x}_i) is not greater than the count of unit clauses that contain \bar{x}_i (x_i), then the variable x_i is set to true (false).

Enforcing a constraint via an inference rule is equivalent to enforcing a constraint on a one-point correlation, both just fix the value of a single variable. Consistent with the introduced notation, when a variable x_i is set to true (false) via an inference rule, a tuple of the form $((i, i), 1)$ ($((i, i), -1)$) gets appended to the assignment A . After applying the inference rules, the procedure is analogous to RQAOA. First, the variational parameters are optimized with respect to the current problem Hamiltonian. The expectation values of all correlations are measured according to this state. Dependent on the sign of the expectation value with the highest absolute value, the assignment is extended and the formula is updated according to the explanation above. In the following, we will refer to this second part of the update step as the *quantum update*. The complete update step can be seen in Alg. 6.

Algorithm 6 Update

Input: partial assignment A , formula ϕ_A
Output extended partial assignments A , Simplified ϕ_A

- 1: $\phi' \leftarrow \{\}$ \triangleright Initialize empty formula for comparison.
- 2: **while** $\phi_A \neq \phi'_A$ **do**
- 3: $\phi' \leftarrow \phi$
- 4: $A, \phi_A \leftarrow \text{PureLiteral}(A, \phi_A)$
- 5: $A, \phi_A \leftarrow \text{AlmostCommonClause}(A, \phi_A)$
- 6: $A, \phi_A \leftarrow \text{ComplementaryUnitClause}(A, \phi_A)$
- 7: $A, \phi_A \leftarrow \text{DominatingUnitClause}(A, \phi_A)$
- 8: **end while**
- 9: Prepare QAOA state with optimal parameters:
 $\beta^*, \gamma^* = \arg \min_{\beta, \gamma} (\langle \psi(\beta, \gamma) | H_\phi | \psi(\beta, \gamma) \rangle)$ $\triangleright (\beta, \gamma) \in [0, 2\pi)^{2p}$
- 10: $V, E \leftarrow \text{CreateGraph}(H_\phi)$
- 11: Store correlations in $M \in \mathbb{R}^{n \times n}$, initialize $M_{ij} = 0$:
 $\forall i \in V : M_{ii} = \langle \psi(\beta^*, \gamma^*) | Z_i | \psi(\beta^*, \gamma^*) \rangle,$
 $\forall (i, j) \in E : M_{ij} = \langle \psi(\beta^*, \gamma^*) | Z_i Z_j | \psi(\beta^*, \gamma^*) \rangle$
- 12: $(i^*, j^*) = \arg \max_{(i \in V \text{ and } i=j) \text{ or } (i, j) \in E} |M_{ij}|$
- 13: $A.\text{append}(((i^*, j^*), \text{sign}(M_{i^*j^*}))$

An important observation is that the update rules for MAX-2-SAT discussed in this context maintain the underlying problem structure. This ensures that all clauses in the formula contain only 2 variables or less throughout the algorithm. As a result, the form of the Hamiltonian remains consistent with Equation (1.10) after each update step.

Backtracking

Since decisions are made based on a heuristic quantum algorithm it is very possible to end up with a non-optimal or unfavorable solution. Backtracking to a previous node of the search tree and exploring other parts of the tree (via changing the decision of a previous correlation rounding) can help improve this solution. Since only a small part of the search tree is explored, the backtracking strategy can have a significant influence on the quality of the final solution. In the following, three very simple backtracking strategies are explained, which are later analyzed via numerical simulations:

- Strategy a): Following the iterative update steps, an initial solution is obtained. Subsequently, the backtracking process is executed, retracing the path of the initial solution and revisiting every node where a fixed value was determined through a quantum update. The enforced correlation constraint on all of these nodes is reversed. After changing the enforced constraint, a new path in the search tree emerges and this assignment is going to be extended via the update rules until a new solution is obtained. The algorithm does not backtrack to nodes whose values were assigned by inference rules, as reversing the decision at these nodes would not lead to improved solutions in the newly emerging search path, given the nature of the inference rules. A visualization of this backtracking strategy can be seen in Fig. 2.2.1. This backtracking strategy is implemented in the pseudocode of the QI-BnB algorithm in Alg. 8.
- Strategy b): This backtracking procedure is identical to strategy a) with the only difference that fewer nodes of the initial exploration path in the search tree are revisited. The algorithm only backtracks to a node at depth k in the search tree whose value was determined by a quantum update if the rounding value of the corresponding correlation is lower than all the rounding values of the nodes at a depth lower than k in the initial exploration path. This backtracking strategy is motivated by the findings in Sec. 2.1.2 which suggested that RQAOA is more likely to make mistakes at low rounding values.
- Strategy c): Again, backtracking and corresponding changing of enforced correlation constraints only happen at nodes of the initial exploration path whose values were fixed by a quantum update. The nodes which are re-visited in this strategy are chosen by a uniform probability.

Once a backtracking strategy is complete and the best solution has been obtained, an additional post-processing step is applied to the solution bitstring. This step involves individually flipping each bit in the final solution and evaluating the number of violated clauses for the resulting bitstrings. If one of the flipped bitstrings exhibits fewer clause violations, the final bitstring is replaced by this one. It is evident that the above-mentioned strategy a) is limited by a maximum of n (number of variables) backtracking steps. Notably, strategies b) and c) perform no more backtracking steps than strategy a). Strategy a) represents the standard technique used in our QI-BnB solver and is also integrated into the algorithmic description in Alg. 8 as well as in the visualization

of the algorithm in Fig. 2.16.

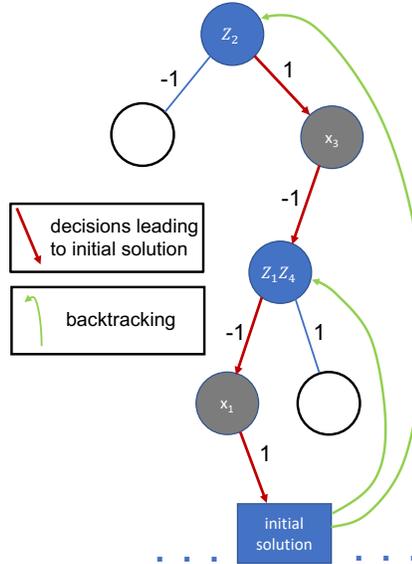


Figure 2.11: Trivial example of solving a Max-2-SAT problem with 4 variables by means of QI-BnB with backtracking strategy a) described in Sec. 2.2.1. The nodes represent variables or two-point correlations and the edges represent their corresponding value in the associated path of the search tree. Decisions that are made by the quantum update are depicted as blue vertices with the corresponding operator with the highest absolute expectation value. The grey vertices represent the variables whose values are chosen by inference rules. The initial solution found by QI-BnB, without backtracking, by repeatedly applying the update step, is represented by the path with the red arrows. The green arrows depict the backtracking steps, which are performed after the initial solution has been found. Backtracking leads back to all blue nodes encountered during the exploration of the first path in the search tree. Jumping back to these nodes and changing the previous decision leads to the emergence of a new path (white node) which subsequently will be explored until a new solution is found.

LowerBound

As explained in Sec. 1.3.1 the lower bound is, as the name implies, a lower bound for the minimal number of unsatisfied clauses if a current partial assignment of a Max-SAT problem would be extended to a complete assignment. For the QI-BnB solver the same lower bound algorithm as in [17] is used. It can be seen in Alg. 7. In this algorithm, the pure literal rule is used. The pure literal rule is applied to clauses that contain a single variable, then the variable gets assigned the proper value so that the clause is satisfied.

Algorithm 7 Lower bound [17]

Input: formula ϕ .
Output lower bound LB

- 1: $LB \leftarrow \text{CountEmptyClauses}(\phi)$
- 2: Applying the one-literal rule on ϕ until a new empty clause is derived
- 3: If no new empty clause can be derived: **return** LB
- 4: Remove all clauses of ϕ which were used to derive the new empty clause
- 5: $LB = LB + 1$
- 6: Back to line 2

QI-BnB - The Complete Algorithm**Algorithm 8** QI-BnB

Input: empty solution A , formula ϕ_A , n_c .
Output complete solution A

- 1: **while** $|A| > n_c$ **do** ▷ Explore initial path in search tree
- 2: $A, \phi_A \leftarrow \text{Update}(A, \phi_A)$
- 3: **end while**
- 4: $A, \phi_A \leftarrow \text{BruteForce}(A, \phi_A)$ ▷ Calculate initial solution by brute force
- 5: $UB \leftarrow \text{CountEmptyClauses}(\phi_A)$ ▷ Init. upper bound by # unsatisfied clauses of ϕ_A
- 6: $A_{UB} \leftarrow A$
- 7: **for** $k = \text{length}(A)$ **to** 1 **do** ▷ Revisit nodes of search tree in reverse order
- 8: $(i, j), \text{sign}(M_{ij}) = A[k]$
- 9: $A' \leftarrow A[1 : k - 1]$
- 10: $A'.\text{append}(((i, j), -\text{sign}(M_{ij})))$ ▷ Changing decision in initial path
- 11: **while** $|A'| > n_c$ **do** ▷ Explore new path
- 12: $A', \phi'_A \leftarrow \text{update}(A', \phi'_A)$
- 13: $LB \leftarrow \text{LowerBound}(\phi'_A)$
- 14: **if** $LB \geq UB$ **then** ▷ Check if new path should already be truncated
- 15: break out of while loop
- 16: **end if**
- 17: **end while**
- 18: $A' \leftarrow \text{BruteForce}(A, \phi_{A'})$
- 19: $UB' \leftarrow \text{CountEmptyClauses}(\phi'_A)$
- 20: **if** $UB' < UB$ **then** ▷ Check if new solution is better
- 21: $A_{UB} \leftarrow A'$
- 22: $UB \leftarrow UB'$
- 23: **end if**
- 24: **end for**

2.2.2 Results

In the following, we apply the QI-BnB algorithm to the random Max-2-SAT problem in order to evaluate its performance and characteristic properties. The simulations encompass a performance investigation across various clause-to-variable ratios and system sizes. Additionally, we evaluate the impact of inference rules and different backtracking strategies on the algorithm's performance and its computational cost. Unless otherwise specified, in all QI-BnB simulations, a brute force threshold of $n_c = 6$ is employed, and correlation expectation values are calculated using depth $p = 1$ QAOA states.

Solver Comparison

In Fig. 2.12 the performance of QI-BnB with backtracking strategy a) (see Sec. 2.2.1) and without backtracking (QI-BnB*) are compared to the ones of standard RQAOA and simulated annealing. The comparison is conducted on 100 randomly generated Max-2-SAT problems with 60 variables and a clause-to-variable ratio α ranging from 1 to 6. The bars indicate for how many instances the solvers were able to find the optimal solution. It can be clearly seen that QI-BnB* outperforms RQAOA on all clause to variables ratios, despite needing fewer optimization procedures for the variational parameters of QAOA. Additionally, using backtracking, the performance increases significantly and even outperforms simulated annealing for all α . Details of the setup of simulated annealing can be found in App. 2.3.4. Obviously, the performance of QI-BnB is lower bounded by the performance of QI-BnB*, but the computational complexity is also higher.

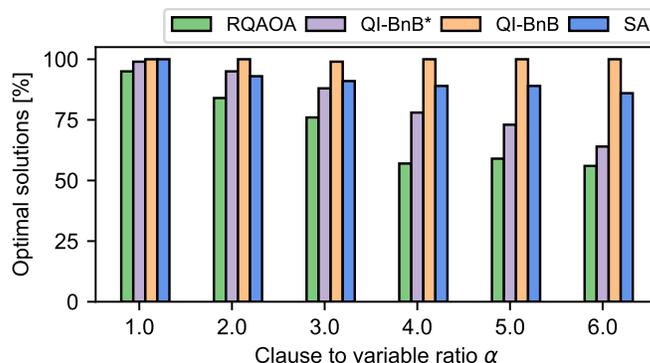


Figure 2.12: Performance of different solvers evaluated on random Max-2-SAT instances with $n = 60$ variables on different clause-to-variable ratios α . The bars show in how many percent of the instances the optimal solutions are found for each solver. 100 random Max-2-SAT instances are generated. The complete Max-SAT solver RC2 is used to calculate the optimal solutions. QI-BnB uses backtracking strategy a) as defined in Sec. 2.2.1 QI-BnB* uses no backtracking. Details about the simulated annealing (SA) setup can be found in App. 2.3.4.

In Fig. 2.13 it is analyzed if the superiority of QI-BnB over RQAOA and simulated annealing is also maintained for larger system sizes on random Max-2-SAT. For differ-

ent clause-to-variable ratios, 100 instances were generated for each system size. It was then evaluated how many of the instances could be optimally solved by the respective algorithm. The size of the systems varies between 40 variables and 160 variables. Since systems of this size can be unfeasible for the RC2 solver, parallel tempering was used to find the alleged optimal solution. A solution of a solver is counted as optimal if it exhibits at least the same number of satisfied clauses as the solution obtained by parallel tempering. Details about the implementation of simulated annealing and parallel tempering can be found in App. 2.3.4. As anticipated, the plots show that the number of optimal found solutions decreases for larger system sizes. The relative performance of the solvers, in terms of their ranking from best to worst in these instances, remains consistent for each clause-to-variable ratio. QI-BnB* outperforms RQAOA significantly for lower α 's but approaches the performance of RQAOA from above for higher clause-to-variable ratios. QI-BnB outperforms simulated annealing for all system sizes and clause-to-variable ratios. It should however be noted that the number of iterations of simulated annealing can be increased arbitrarily and therefore it eventually outperforms QI-BnB if enough update steps are performed.

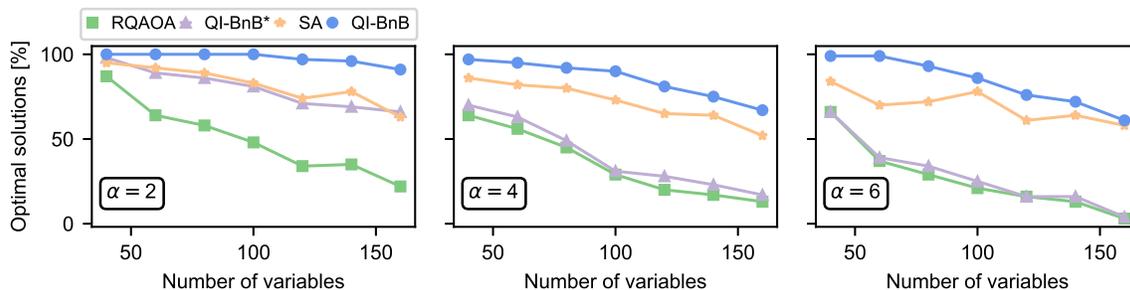


Figure 2.13: Scaling analysis for the performance of RQAOA, QI-BnB (with backtracking strategy a) Sec. 2.2.1), QI-BnB* (without backtracking) and simulated annealing on the random Max-2-SAT problem for different clause to variable ratios α . The system sizes range from 40 variables to 160 variables. The alleged optimal solutions were obtained via parallel tempering, conducted with sufficient resources. The plots show for how many instances the respective solvers could find a solution of equal or better quality than parallel tempering. Details about the implementation of simulated annealing and parallel tempering can be found in App. 2.3.4.

Update step

In Fig. 2.14 we assess the role of the quantum update and the inference rules in the update step of QI-BnB* (QI-BnB without backtracking). To this end, 100 random Max-2-SAT instances with $n = 60$ variables and different clause-to-variable ratios α are generated. In these instances QI-BnB* with the standard update step as defined in Sec. 2.2.1 (coined "informed" in the figure) is compared to QI-BnB* where the quantum update of this update is replaced. It is replaced by choosing one of the existing one- or two-point correlations with uniform probability and enforcing with uniform probability a perfect correlation or anti-correlation on these variables (coined "uninformed" in the

figure). The bars indicate the average approximate ratio of the two algorithms on the random Max-2-SAT instances. As expected, the information provided by the QAOA circuit proves to be very valuable and QI-BnB* outperforms its uninformed version significantly for all clause-to-variable ratios. Therefore, we see that the quantum part of the algorithm is still the work-horse of QI-BnB and the good performance is not mainly based on the augmentation of the classical subroutines. It should be noted that without the inference rules an approximation ratio of 0.75 would be expected. The approximation with random decisions augmented with inference rules can be seen to fall roughly between 0.80 and 0.85.

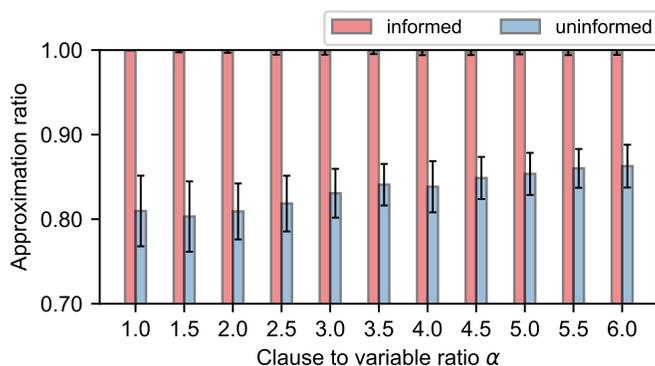


Figure 2.14: The approximation ratios of QI-BnB* (QI-BnB without backtracking) and the same algorithm substituting the quantum update with the enforcement of a random correlation constraint are compared. For various clause-to-variable ratios, this evaluation is conducted on 100 random Max-2-SAT instances, each consisting of 60 variables.

Backtracking

After an initial solution is found in the QI-BnB algorithm, backtracking is applied. Every node in the initial exploration whose value has been assigned by a quantum update is revisited and its previously assigned value is changed. These new emerging paths can potentially lead to an improved solution. The Hamming distances between these improved solutions and the initial solution of the algorithm are shown in Fig. 2.15 for 100 randomly generated Max-2-SAT problems with different clause-to-variable ratios α . In each of the three plots, the bars represent the frequency of the improved solutions having a specific Hamming distance among the 100 random Max-2-SAT instances. It can be seen that these Hamming distances are all between 2 and 30, with most of them found between 2 and 20. The plot demonstrates that the solutions do not only exhibit trivial improvement, indicating that the enhancements are not solely achieved by changing a single value of a node in the initial exploration path. If that were the case, the Hamming distances between the improved solutions and the initial solutions would be equal to one.

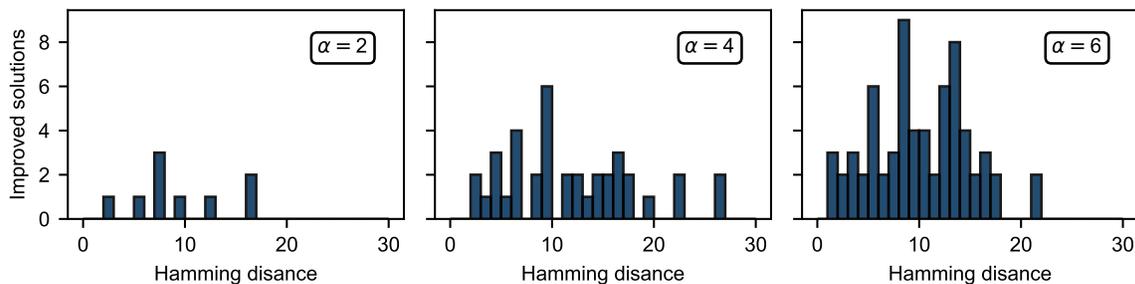


Figure 2.15: Measured Hamming distances between solutions found by the QI-BnB solver during the initial tree exploration and the best overall solutions after backtracking and executing the whole algorithm. The distances were measured on 100 random Max-2-SAT instances with 60 variables generated for 3 different clause-to-variable ratios α .

In Fig. 2.16 the QI-BnB algorithm with the standard backtracking strategy a) and the two other strategies described in Sec. 2.2.1 are compared. The solvers are compared on the random-Max-2-SAT problem with $n = 60$ variables over different clause-to-variable ratios. 100 instances were generated for each ratio.

The left plot presents bars that indicate the number of instances in which the QI-BnB algorithm, employing different backtracking strategies, was able to find an optimal solution. Consistently across all clause-to-variable ratios, strategy a) outperforms strategy b), and strategy b) outperforms strategy c). While these results were expected to some degree, it may surprise that the performance difference between strategy b) and c) is not more pronounced, considering the findings in Sec.2.2 regarding the occurrence of wrong decisions of RQAOA. However, the QI-BnB solver incorporates inference rules that are not part of RQAOA, which helps in avoiding the occurrence of wrong decisions made by RQAOA.

The right plot displays the number of times the variational parameters of QAOA need to be optimized for the different backtracking strategies. In the absence of inference rules and lower bound computation during the update step, the number of necessary parameter optimizations for strategy a) would be $\frac{n \cdot (n-1)}{2} = 1770$ times for these Max-2-SAT instances. We see that significantly fewer optimization procedures are needed for all different backtracking strategies, particularly for lower clause-to-variable ratios. But also for a clause-to-variable ratio of $\alpha = 6$, the number of parameter optimizations can be reduced by approximately one order of magnitude. The reason strategy c) requires slightly fewer optimizations than strategy b) is likely because quantum update decisions are reversed randomly in this strategy, rather than those with low rounding values, resulting in the lower bound computation truncating these regions of the search tree more effectively.

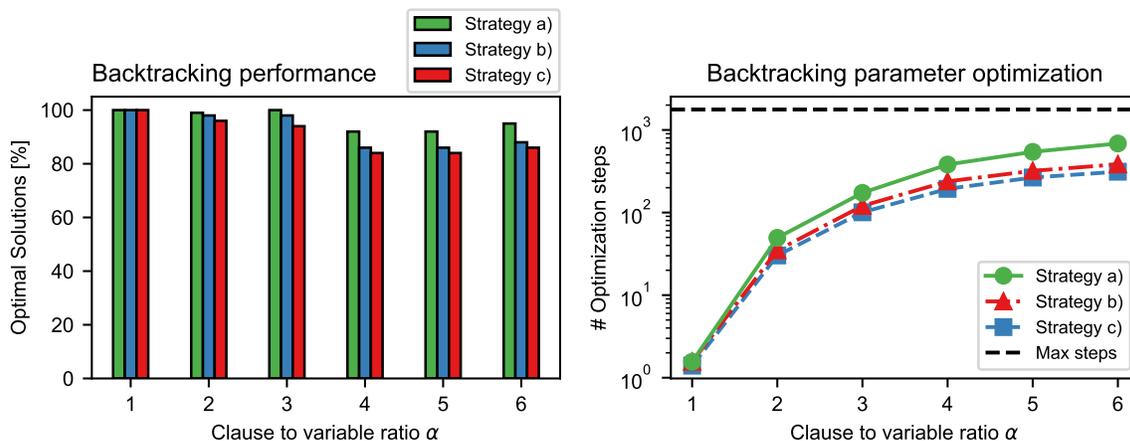


Figure 2.16: Backtracking performance (left): For six different clause-to-variable ratios α of the random Max-2-SAT problem, the performance of QI-BnB with the three different backtracking strategies described in Sec. 2.2.1, is evaluated. For each ratio, 100 instances with 60 variables were generated. Backtracking parameter optimization (right): The average number of optimizations performed on the variational parameters of the QAOA circuit for a Max-2-SAT instance is compared among the three different backtracking strategies, with the comparison being dependent on the clause-to-variable ratio α .

2.2.3 Discussion

Building upon the recursive approach of RQAOA for solving combinatorial optimization problems, we present a novel hybrid quantum-classical solver for Max-2-SAT. Instead of applying generic simplifications on the problem Hamiltonian, we propose problem specific classical update steps to recursively simplify the Max-2-SAT formula and extend the assignment. This allows us to leverage classical subroutines from the extensive literature of Max-SAT solvers. Incorporating techniques from branch and bound solvers allows us to address challenges faced by RQAOA and other quantum algorithms, while adding almost no classical computational cost and keeping the quantum hardware requirements unchanged. The incorporated techniques include inference rules, different backtracking strategies and lower bound computations.

Our numerical investigations have shown the promising performance of our Quantum Informed Branch and Bound solver when applied to the Max-2-SAT problem in various clause-to-variable regimes. The efficient classical simulation of level $p = 1$ QAOA expectation values of one- and two-point correlations enabled us to examine comparably large system sizes with a quantum algorithm. The implementation of problem-specific classical optimization techniques evidently improves the performance of previous iterative quantum algorithms like RQAOA. The results also show that the utilization of backtracking strategies allows QI-BnB to compete with classical heuristic optimization algorithms like simulated annealing. Moreover, the combination of inference rules

and lower bound computations show to significantly reduce the necessary calls of the QAOA algorithm. The results of QI-BnB motivate to extend its approach of combining QAOA with classical problem-specific subroutines to other combinatorial optimization problems and different quantum protocols for low-energy state preparation, beyond QAOA. Our work demonstrates not only the potential of the performance of these hybrid quantum-classical algorithms but also shows their flexibility in designing new algorithms.

2.2.4 Outlook

First of all, the potential research directions discussed in the RQAOA-outlook in Sec. 2.1.4 are equally interesting in regard to our QI-BnB algorithm. This includes evaluating the value of the quantum part in this algorithm. This could be done by replacing the quantum update of the algorithm with a classical counterpart to calculate the expectation values of the correlations. Furthermore, simulations with higher QAOA depths could be explored. Moreover, instead of employing the low-energy QAOA state to perform a deterministic update step, it could be utilized to inform a probability distribution. This enables a probabilistic selection of the correlation and its associated enforced constraint. This approach becomes particularly interesting when combined with the backtracking procedure. In QI-BnB, during the quantum update, it is possible to encounter situations where there is no significant distinction between the largest absolute expectation value of a correlation and others (even a tie among multiple correlations is no exception). In such scenarios, it can be beneficial to explore various correlations to enforce a constraint on. These options could be explored by means of backtracking.

The kind of modular design of the QI-BnB algorithm in terms of a quantum and a classical part in the update step, offers significant advantages, particularly in terms of its flexibility. It allows for the utilization of subroutines to mitigate performance limitations inherent to local quantum algorithms such as QAOA. In our QI-BnB algorithm, we have demonstrated how classical optimization techniques can be tailored according to a specific optimization problem, using the Max-2-SAT problem as an example. Akin to that, it could be valuable to design the algorithm to explore the geometric features of the cost function. Subsequently, utilizing this information assists the solver in alleviating these topological barriers within the solution space. These barriers are often crucial in limiting the performance of both classical and quantum algorithms. Successful classical random K-SAT solvers have been developed using similar methods [10, 50]. In RQAOA and QI-BnB, in each update step, the variational parameters of QAOA are optimized to minimize the energy of the cost Hamiltonian in regard to the QAOA state. However, it could be advantageous to optimize these parameters using a modified cost Hamiltonian. Several possibilities exist in this direction. For example, one can localize the Hamiltonian by adding a sum of one-point projectors to it, hence, favoring assignments within specific regions of the solution space (e.g. searching in clusters of near-optimal solutions, see Sec 1.2.3). Similarly, terms could be added to penalize solutions within certain regions of the solution space. By combining this approach with backtracking, it becomes possible to enforce hard constraints effectively, increas-

ing penalization when violations occur during the course of the algorithm. Another interesting approach, as demonstrated in [14], involves modifying the cost Hamiltonian to produce solutions that are optimized for subsequent post-processing. One can see the potential for various modifications and extensions of the developed algorithm.

In a more general direction the idea of QI-BnB to utilize problem-specific classical procedures to incrementally extend the assignment and simplify the problem could be extended to other combinatorial optimization problems such as the Max-Cut or the Maximum Independent Set problem. Furthermore, QAOA is already a hybrid quantum-classical algorithm, therefore the incorporation of quantum-informed classical update rules seems like a natural extension. However, different quantum protocols other than QAOA could be utilized to produce low-energy quantum states. Even non-gate-based devices, like quantum annealers, could be used for this purpose.

2.3 Quantum Informed Cluster Algorithm

As discussed earlier, a wide range of optimization problems can be formulated as ground-state problems for Ising-like spin glasses. Heuristic optimization algorithms like simulated annealing are widely used techniques to address such problems. However, simulated annealing encounters severe challenges when dealing with complex energy landscapes, as it can easily get stuck in local minima. In this section, we propose a novel quantum-informed non-local cluster algorithm to alleviate this issue. A low-energy quantum state is prepared to measure favorable correlations in the space of low-energy solutions. By utilizing this approach, non-local clusters are built probabilistically. These clusters aid in navigating the prospective solution through the intricate energy landscape and enable escape from local minima. A comprehensive numerical investigation is conducted on problems with tunable hardness defined on 2- and 3-high-dimensional lattices with over 200 variables. Key properties of the algorithm are explored and its performance is compared to well-established classical optimization algorithms such as simulated annealing and parallel tempering. Finally, different strategies and possible modifications of the algorithm are discussed, including the possibility of utilizing this strategy to construct an MCMC algorithm.

2.3.1 Algorithm

Cluster algorithms have a rich history in the simulation of spin-glass systems. Early examples of Monte Carlo algorithms of this kind include the Swendsen-Wang algorithm (applicable to both Ising and Potts models)[23] and the generalized Wolff algorithm[64]. These algorithms have demonstrated particular efficacy in simulating spin glasses close to criticality, mitigating the issue of critical slowing down. The random cluster model [33] serves as a valuable conceptual framework when working with cluster models. However, for our cluster algorithm, we adopt a more informal approach.

Our Quantum Informed Cluster Algorithm (QICA) is a heuristic non-local cluster algorithm designed to find the ground state of Ising-like spin glasses. To the best of our knowledge, this is the first cluster algorithm that utilizes quantum computing. Various quantum protocols could be used to prepare a low-energy state to inform the QICA. However, in the following, we will only use depth $p = 1$ QAOA to prepare the quantum state since this allows us (like in the case of the QI-BnB algorithm) to perform simulations with more than two hundred variables.

The algorithm initiates by optimizing the variational parameters of the QAOA state according to the given problem Hamiltonian. Subsequently, the correlation between connected variables on the interaction graph is computed. This involves measuring the expectation values of two-point correlations in the form of $Z_i Z_j$ according to the QAOA state. Each expectation value determines a corresponding link probability $p_{ij}^l(x) = \max(0, x_i x_j \langle Z_i Z_j \rangle)$ on every edge (referred to interchangeably as "edge" or "link" hereafter). These link probabilities contain information pertaining to the low-energy structure of the solution space (since they are calculated via the low-energy QAOA state).

Following the assignment of link probabilities to each edge of the graph, the algorithm follows a similar procedure to simulated annealing. Initially, a starting temperature is chosen and a random configuration is prepared. Subsequently, consecutive update steps are performed while gradually decreasing the temperature toward zero. In each update step, a random seed is chosen, and a cluster is built according to the link probabilities. We refer to a link as being *activated*, if it connects variables to the cluster. The entire cluster is then flipped. Notably, unlike simulated annealing, where individual spins are flipped, this algorithm flips clusters that are deemed to be energetically favorable for QAOA. The algorithm can be seen in Alg. 9.

Algorithm 9 QICA (linear schedule)

Input: Classical and corresp. quantum Hamiltonian $H(\cdot)$ & \hat{H} with number of spins n , final inverse temperature β_f , number of iterations m , QAOA depth p
Output final state x , final energy E

- 1: $V, E \leftarrow \text{CreateGraph}(\hat{H})$
- 2: Prepare QAOA state with optimal parameters:
 $\beta^*, \gamma^* = \arg \min_{\beta, \gamma} (\langle \psi(\beta, \gamma) | \hat{H} | \psi(\beta, \gamma) \rangle) \quad \triangleright (\beta, \gamma) \in [0, 2\pi)^{2p}$
- 3: Create link probabilities and store in p^l : \triangleright container incorp. $|E|$ link prob. functions
 $\forall (i, j) \in E : p_{ij}^l : (x_i, x_j) \mapsto \max(0, x_i x_j \langle \psi(\beta^*, \gamma^*) | Z_i Z_j | \psi(\beta^*, \gamma^*) \rangle)$.
 \triangleright note: probability function is symmetric: $\forall i, j : p_{ij}^l = p_{ji}^l$
- 4: $\beta = 0 \quad \triangleright$ Starting annealing procedure
- 5: $\Delta\beta = \beta_f / (m - 1)$
- 6: Initialize random spin configuration x
- 7: $E = H(x)$
- 8: **for** $step = 1$ **to** m **do**
- 9: $i \leftarrow \text{Uniform}(1, n) \quad \triangleright$ Choose seed with uniform probability (natural number)
- 10: $C \leftarrow \text{CreateCluster}(i, p^l) \quad \triangleright$ Function creates cluster according to link probabilities
- 11: $x' \leftarrow \text{Flip}(x, C) \quad \triangleright$ Flip spin at site i
- 12: $\Delta E = H(x') - H(x)$
- 13: **if** $\Delta E < 0$ **then** Accept state
- 14: $x \leftarrow x' \quad \triangleright$ Accept state if energetically favourable
- 15: **if** $H(x) < E$ **then**
- 16: $E = H(x) \quad \triangleright$ Update best energy
- 17: **end if**
- 18: **else if** $\text{Uniform}(0, 1) < e^{-\beta\Delta E}$ **then**
- 19: $x \leftarrow x' \quad \triangleright$ Accept state with prob. $e^{-\beta\Delta E}$
- 20: **end if**
- 21: $\beta = \beta + \Delta\beta \quad \triangleright$ Reduce temperature
- 22: **end for**

In each update step, the cluster building proceeds as follows: firstly, a random seed is selected. From this seed, the cluster construction begins. Each link that connects a spin outside the cluster with a spin inside the cluster is probed. If the orientation of the two spins relative to each other is favored in regard to the low-energy information produced by QAOA, then there exists a certain probability that the link between these

two variables is going to be activated and therefore the outlying spin becomes part of the cluster. The cluster grows in this manner until all outlying spins which are connected to the cluster via a link, fail to connect to the cluster. Such a cluster can be seen in Fig. 2.17. The pseudocode of the *CreateCluster* function is shown in Alg. 2.17.

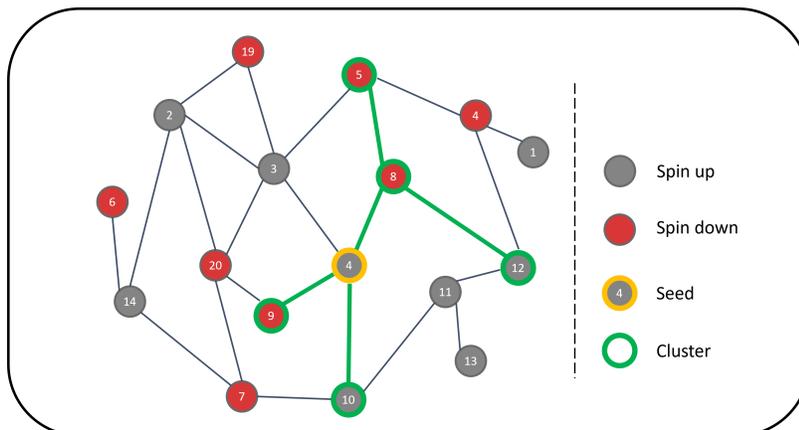


Figure 2.17: Illustrative example of a cluster, constructed on a lattice. The activated links (green lines) connect spins to form a cluster. The cluster began to grow starting from spin 4 (seed).

Algorithm 10 CreateCluster

Input: Seed index k , probabilities p^l .

Output C

- 1: $V, E \leftarrow \text{CreateGraph}(\hat{H})$
 - 2: $C = \{k\}$
 - 3: **while** *continue* == *True* **do**
 - 4: $\forall e \in E: \exists i \in e \cap C \wedge \exists j \in e \cap V \setminus C \implies$ activate e with prob. $p_{ij}^l(x_i x_j)$
 - 5: Add all vertices to C that are attached to activated links e
 - 6: **if** no vertices are added **then** *continue* = *False*
 - 7: **end if**
 - 8: **end while**
-

We also introduce a variation of QICA, called QICA-p algorithm. It aims to enhance the effectiveness of cluster flips in our approach. The algorithm is identical to the one presented in Alg. 9 but with one difference: the link probabilities are periodically multiplied by a factor of $(3 \cdot \text{update_count} \cdot \text{mod}_{24})/10$. By adjusting the link probabilities, the algorithm can flip both smaller and larger clusters. The idea is that sometimes larger cluster flips could be needed to escape local minima and applying smaller cluster moves could help to locally thermalize these flips.

2.3.2 Results

In the following, we investigate the performance of the QICA algorithm on a regular 2D and 3D lattice. As the algorithm design closely resembles that of simulated annealing, with the inclusion of quantum-informed cluster building, it is crucial to achieve better performance than simulated annealing itself. Otherwise, the formation of clusters would merely introduce unnecessary additional computational costs. In order to evaluate the benefit of the information about low-energy states provided by QAOA, QICA is also compared to a classical counterpart in which clusters are built randomly. Furthermore, the performance dependence of the algorithm in regard to the quality of the QAOA state is analyzed. Finally, QICA is compared to the commonly used parallel tempering algorithm.

All instances are generated using the tile planting technique implemented in the Chook library [57]. Besides knowing the ground state energy of the problem, this allows us to create problems with tunable hardness, thereby creating instances within a wide range of difficulty, up to highly challenging ones with very rugged energy landscapes. The hardness of these instances on the 2D lattice is investigated in [58]. 3 probability parameters can be chosen that influence the hardness of the problem. Perera et al. showed that setting $p_3 = 0$ the hardness of the problem increases with higher values of the parameter p_2 . Therefore, we create problems with varying hardness parameters $HP = p_2$ on the 2D lattice. On the 3D lattice, there are two free probability parameters to choose. Setting the parameter $p_{42} = 0$, it was shown in [36] that the hardness of the instance increases with increasing p_{22} . Therefore, we generate problems with varying parameters $p_{22} = HP$ on the 3D lattice. Another important quantity that appears throughout this chapter is the approximation error which we define as $\alpha = 1 - E/E_0$, where E_0 is the ground state energy of the problem and E the energy achieved by the algorithm. The chosen parameters for simulated annealing and parallel tempering can be found in App. 2.3.4. Annealing algorithms need to be restarted for evaluation at different times due to fixed start and final temperatures determined by the annealing schedule. Varying the number of flips alters the slope at which temperatures decrease until reaching the final value of $1/T = \beta = 6$.

To better understand the value of the information QAOA yields for building the clusters, we additionally implement a classical uninformed version of our cluster algorithm in which all link probabilities p_{ij}^l are set to a constant. For the 2D and 3D clusters, the classical counterparts are tested using ten different constant probabilities ranging from 0.1 to 1.0 in increments of 0.1. It turns out that for the 2D lattice, the optimal value for the link probabilities is $p = 0.3$ and for the 3D lattice it is $p = 0.2$. This algorithm is coined "Uninformed" in the following simulations. The annealing schedule of QICA(- p) and the classical uninformed cluster algorithm is chosen to be identical to the one of simulated annealing so that they can be better compared. The time for all these different algorithms is measured in terms of the number of clusters, respectively single spin flips. Even though building a cluster comes at a computational cost for the cluster algorithms, this aspect is not taken into account during simulations. The reason for this is the challenge of comparing it to non-cluster algorithms. The same is true for

the replica exchange step in parallel tempering. Finally, it is important to mention that we use an unfavorable metric for QICA in comparison to parallel tempering. The performance of the algorithms is evaluated by measuring the number of optimally solved instances, or the average approximation error depending on the running time. However, the performance of algorithms that employ an annealing schedule like QICA(-p) is usually evaluated differently. Typically, these algorithms are run multiple times, and the best solution is selected instead of relying on a single prolonged run [62].

Comparison to Classical Counterparts

In Fig. 2.18 we compare QICA(-p) to classical algorithms. For six different hardness parameters, a problem was generated on a 14×14 2D lattice. The number of flips (time) increases from top to bottom. The classical uninformed cluster algorithm outperforms simulated annealing. QICA and QICA-p both significantly outperform simulated annealing and their classical counterpart. The periodic change in link probabilities further improves the performance of QICA. This difference in performance for the different solvers holds for all different hardness parameters and for all times as well as for both metrics, the approximation error and the number of optimally found solutions. As expected, with increasing hardness parameters the number of optimal found solutions decreases for each solver. However, QAOA-p even finds all solutions for the hardest instances for the maximum number of cluster flips executed. It is also remarkable that for a very low number of flips, QICA-p solves almost all easier instances optimally and about half of the harder instances. Simulated annealing and the uninformed cluster algorithm on the other hand, solve almost no instances during that time.

The performance of the solvers on the $6 \times 6 \times 6$ 3D lattice in Fig. 2.19 resembles the results of the 2D lattice. QICA and QICA-p still outperform the classical algorithms. However, the difference is not as pronounced as for the 2D lattice. Especially for hardness parameters from 0.6 upwards, even QICA-p only finds less than half of the optimal solutions. It seems that building clusters becomes generally more difficult on this higher-dimensional lattice since the classical uninformed cluster algorithm provides no advantage over simulated annealing anymore.

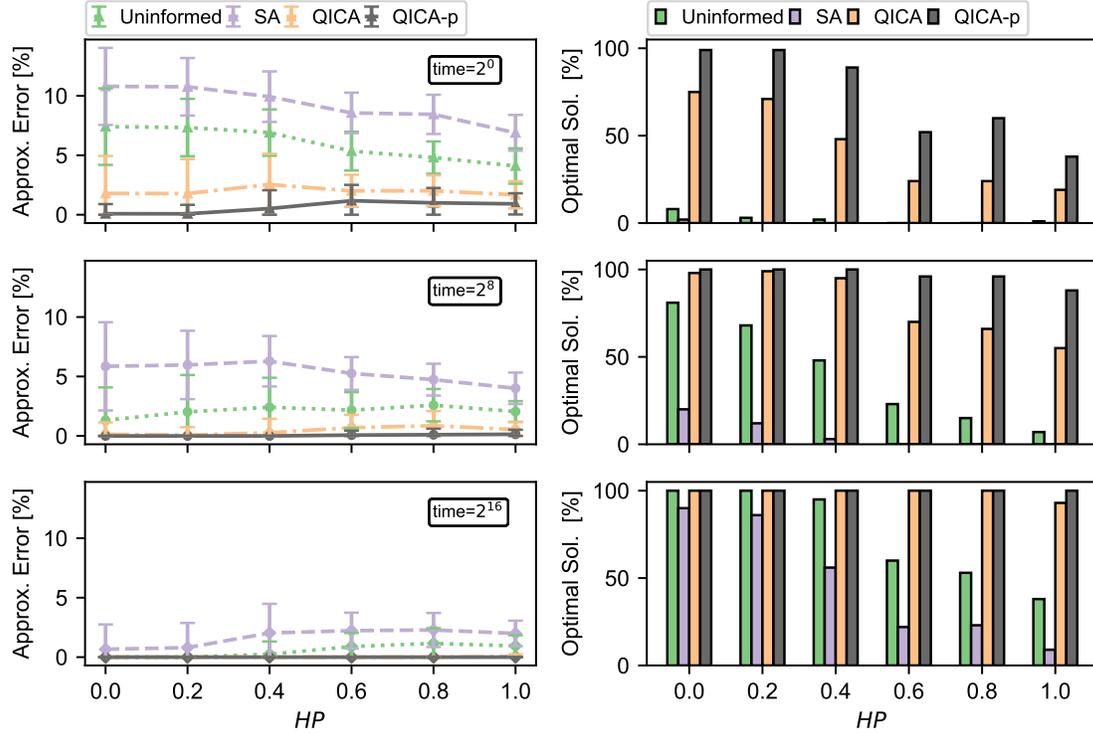


Figure 2.18: Performance comparison of cluster algorithms and simulated annealing (SA) on 6 different problems with 6 different hardness parameters (HP) with $n = 196$ variables on a 2D lattice. One hundred runs were performed for each solver on each instance. Horizontal plots show the performance of the algorithms in metrics of the approximation error (left) and the number of optimal found solutions (right) for a given time. One time unit corresponds to $60 \cdot n$ flip of the cluster, respectively a single spin. Details and chosen parameters for the generated instances and the algorithms can be found in the text.

Comparison to Parallel Tempering

The simulations in Fig. 2.20 show how fast QAOA-p and parallel tempering reach the ground state for 3 problem instances with different hardness parameters defined on the 14×14 2D lattice. For 100 different runs of the two algorithms on each problem instance, the percentage of optimal solutions found and the average approximation error is visualized at different times. The behavior of the algorithm in regard to their approximation error and how often they found the optimal solutions are qualitatively the same. It can be seen that QICA-p outperforms parallel tempering for all instances, independent of the hardness parameter. QICA-p reaches the ground state between 2 and 3 orders of magnitude fast than PT.

In Fig. 2.21 the same experiment is carried out as in Fig. 2.20 but on the $6 \times 6 \times 6$ 3D lattice. Again QICA-p outperforms parallel tempering. Although the difference is less pronounced and shows about 1 and up to 2 orders of magnitude.

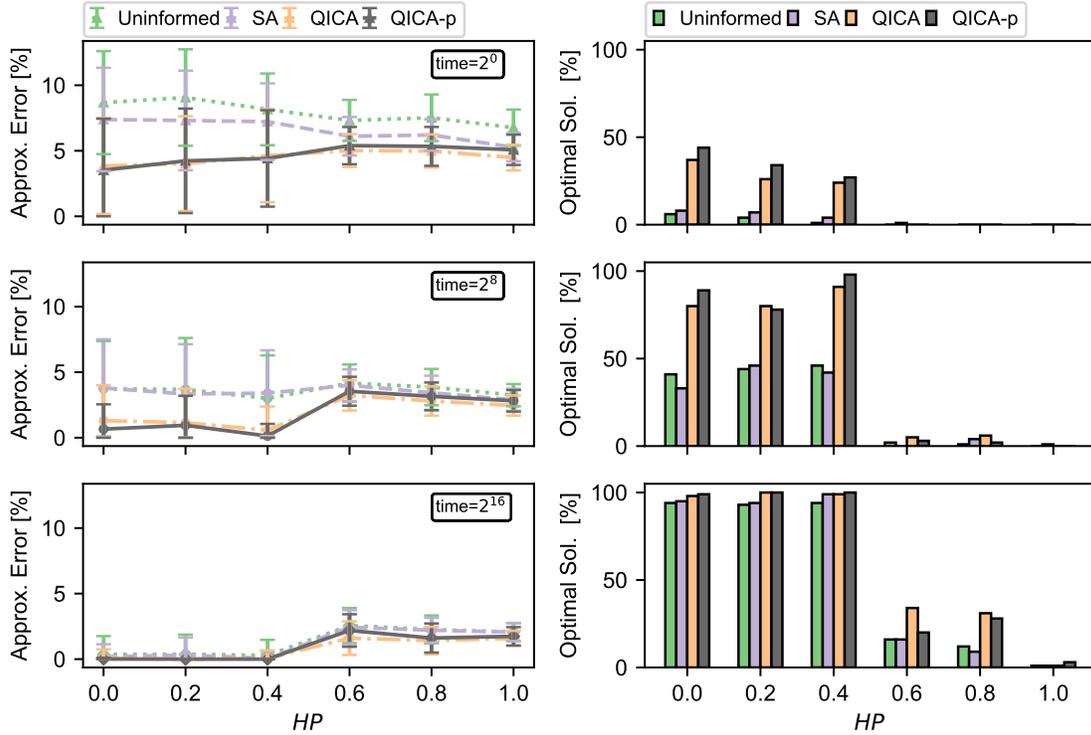


Figure 2.19: Performance comparison of cluster algorithms and simulated annealing (SA) on 6 different problems with 6 different hardness parameters (HP) with $n = 216$ variables on a 3D lattice. One hundred runs were performed for each solver on each instance. Horizontal plots show the performance of the algorithms in metrics of the approximation error (left) and the number of optimal found solutions (right) for a given time. One time unit corresponds to $60 \cdot n$ flip of the cluster, respectively a single spin. Details and chosen parameters for the generated instances and the algorithms can be found in the text.

Quantum State Quality and Performance

In Fig. 2.22 the average cluster sizes of QICA and for comparison also of the classical uninformed cluster algorithm are visualized. These cluster sizes were measured in the simulations of Fig. 2.18 and 2.19. The mean cluster sizes (number of sites involved in a cluster flip) throughout the course of the algorithms when applied on an instance are calculated. These values are then averaged over the number of independent runs on the corresponding instance. One should keep in mind that the constant link probabilities for the classical uninformed cluster algorithm were optimized. Therefore, it is interesting to see that the average cluster sizes of the uninformed algorithm are so similar to the ones of QICA. Furthermore, it can be seen that, for both lattices, there is a decrease in the cluster sizes with increasing hardness of the instances. While there is a rather slight decrease in cluster sizes for the 2D lattice, this correspondence becomes very evident for the 3D lattice.

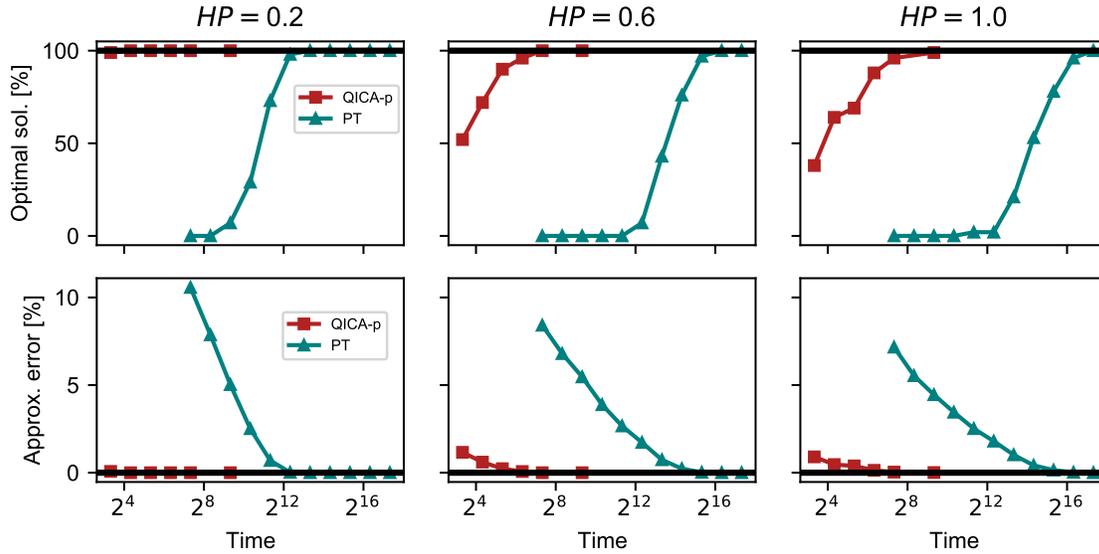


Figure 2.20: Convergence behavior on 2D lattice. The simulation was performed on 3 of the 6 problems with $n = 196$ from Fig. 2.18. One time unit on the x-axis corresponds to n cluster flips of QICA-p, respectively one flip in one replica of the parallel tempering algorithm (PT). The top row shows on how many of the 100 independent runs the two algorithms achieved to find the ground state within different times. The bottom row shows the average approximation error over the 100 independent runs for each problem.

In Fig. 2.23 the performance of QICA dependent on different variational parameters of the $p = 1$ QAOA state with varying quality is evaluated. The approximation error of QICA on a problem instance with $HP = 0.4$ on the 2D lattice is plotted against the different expectation values of the problem Hamiltonian according to their respective variational parameters. A very clear improvement in the performance of QICA can be seen as the quality of the QAOA states improves.

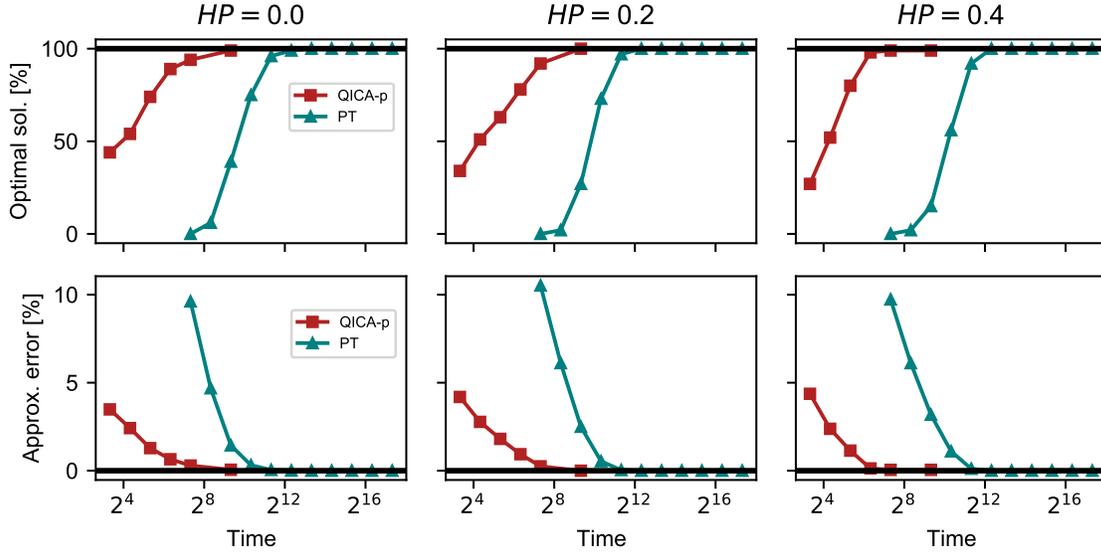


Figure 2.21: Convergence behavior on 3D lattice. The simulation was performed on 3 of the 6 problems with $n = 216$ from Fig. 2.19. One time unit on the x-axis corresponds to n cluster flips of QICA-p, respectively one flip in one replica of the parallel tempering algorithm (PT). The top row shows for how many of the 100 independent runs the two algorithms achieved to find the ground state within different times. The bottom row shows the average approximation error over the 100 independent runs for each problem.

2.3.3 Discussion

We have proposed a novel hybrid quantum-classical cluster algorithm. It is important to highlight that this approach differs inherently from the recursive optimization utilized in RQAOA and QI-BnB. QICA only executes the quantum protocol at the initial stage of the algorithm to produce a low-energy state. This state is subsequently utilized in the cluster algorithm, where it calculates link probabilities. These probabilities are then employed in the annealing procedure to repeatedly form energetically favorable clusters. These clusters are flipped to eventually reach a low-energy state. Utilizing quantum computing only in the initial phase of the algorithm to provide information to the classical solver to enhance its performance, is intriguing in and of itself. It represents a relatively unexplored approach in the literature and has the potential to beneficially leverage quantum capabilities in solving problems that are challenging to tackle classically.

Extensive numerical simulations have demonstrated the remarkably strong performance of QICA across problems in 2 and 3 dimensions in various hardness regimes. As required, QICA significantly outperforms simulated annealing as well as its classical counterpart, the uninformed cluster algorithm. By periodically varying the link probabilities, the performance of QICA is further improved. QICA-p even outperforms parallel tempering on the 2D and 3D lattice, despite the fact that PT is usually very strong on

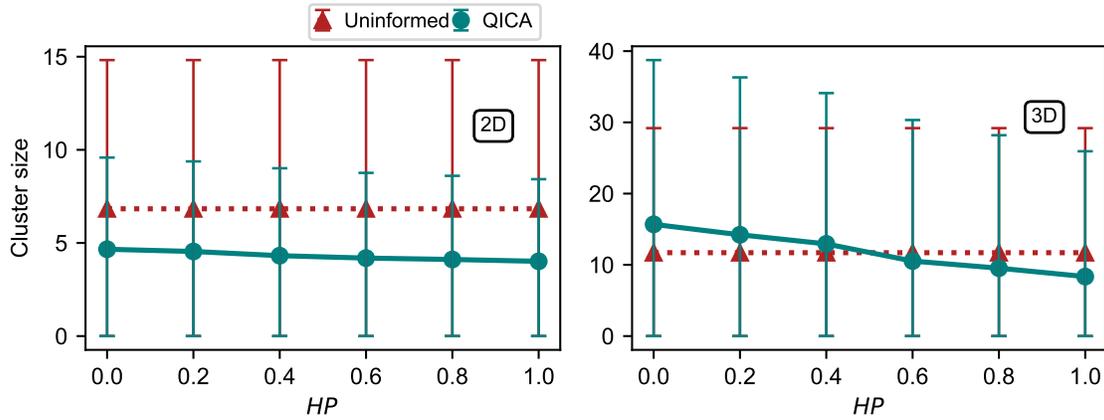


Figure 2.22: The average cluster sizes of QICA and the classical uninformed cluster algorithm on the 2D and 3D lattice. The cluster sizes were measured during the simulations performed in Fig. 2.18 and 2.19. Additionally, the mean value and standard deviation of the cluster sizes for the run of the algorithm on a single instance were calculated. These values were also averaged over the one hundred independent runs of the algorithm on the corresponding instances.

problems with rugged energy landscapes, like the ones investigated. This was achieved even though the used metric was particularly unfavorable for a heuristic algorithm like QICA-p. However, it should be mentioned that we did not account for the additional computational cost associated with building the cluster in QICA (we also neglected the replica exchange cost in PT). The performance difference to the aforementioned well-established classical solvers is less pronounced on the 3D lattice than it is on the 2D lattice. It turns out that the cluster sizes decrease with the increasing hardness of the problems. Furthermore, initial explorations have been carried out on the so-called Wishart lattice [37] (also incorporated in the Chook library), which exhibits all-to-all connectivity and can therefore be interpreted as an infinity-dimensional lattice. In this case, QICA and QICA-p were unable to improve on simulated annealing in a significant way. However, these results are not documented in this thesis, since the exploration is incomplete and further experiments with different setups of QICA still have to be conducted. To conclude, experiments demonstrated that our initial intuition regarding QICA has been validated since we have shown that QAOA states with lower energy indeed aid the algorithm in effectively traversing the energy landscape and achieving better performance.

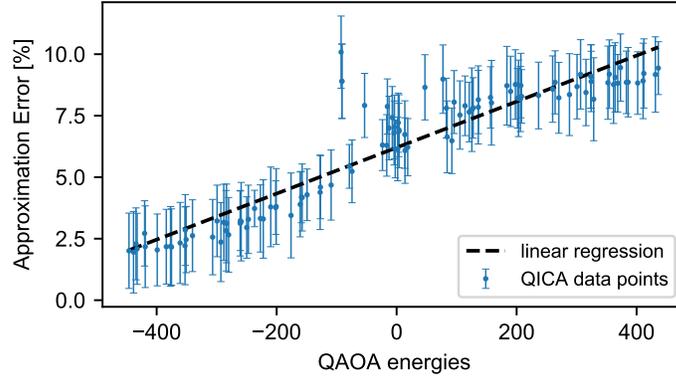


Figure 2.23: Performance of QICA dependent on the quality of the QAOA state. A problem with $HP = 0.4$ was generated on a 20×20 2D lattice. The parameter space $[0, \pi) \times [0, \pi)$ of depth $p = 1$ QAOA was divided in a 400×400 grid. After calculating all energy expectation values of the problem Hamiltonian according to the QAOA states evolved with these parameters, the parameters were sorted according to their energy values. Dividing this set of parameters into 10 equally spaced sections of parameters in regard to the energies of QICA, 11 random parameters were picked out of each section. With these 110 QAOA states the correlations of the problem variables were calculated to inform QICA. The achieved approximation error of QICA in dependence on the QAOA energy expectation values of the used parameters is plotted. A linear regression was calculated via the mean values of the approximation errors.

2.3.4 Outlook

The novel approach of incorporating quantum information into a cluster algorithm opens up a plethora of subsequent research directions. One crucial question that arises is whether it is feasible to design the algorithm as a Markov Chain Monte Carlo (MCMC) algorithm. However, it will probably be challenging to devise a MCMC cluster algorithm that shows a certain level of activity also for low temperatures. To achieve this, one could explore adjusting the probabilities of cluster construction and cluster-flipping acceptance, ensuring that the algorithm satisfies detailed balance (while ergodicity can be relatively straightforwardly fulfilled). However, a major hurdle is to construct clusters that are not always rejected at low temperatures, as this would lead to unnecessary computational costs. Building a rejection-free MCMC cluster algorithm like Swendsen-Wang (acceptance probability $1/2$ to be concise) might be challenging when leveraging the information from QAOA states. QAOA is a heuristic algorithm and is not inherently linked to the system's internal energy. To overcome this obstacle, one intriguing approach could involve simulating multiple replicas. This technique has demonstrated great utility in similar contexts [40, 66].

These aforementioned possible limitations aside, sticking with a heuristic optimization algorithm offers several advantages, as it grants greater flexibility in designing and tailoring the algorithm to specific problems. The proposed algorithm design should

serve primarily as a prototype, leaving room for exploring various different formulations of the algorithm and optimizing the parameters. This includes examining different link probability functions, optimizing the parameters of the algorithm, or establishing dependencies on the underlying lattice's structure. For instance, it may prove beneficial to scale the link probabilities based on the corresponding vertex's degree. Another possibility is to adjust the acceptance probability according to the cluster size, making it easier to flip larger clusters. This adjustment could also aid in mitigating percolation issues. Another interesting approach could be to simulate several replicas akin to parallel tempering and allow exchanges between them. However, instead of simulating each replica at a different temperature, each replica could be simulated with different link probability functions. One way to do this could be by starting each replica with a different bitstring. Choosing bitstrings separated by a large distance (possibly even using pre-optimized bitstrings) might help the algorithm to incorporate information about large parts of the solution space and share this information among the replicas via the exchange step.

Furthermore, a wide variety of quantum protocols could be utilized to calculate the low-energy quantum state. This includes not only gate-based approaches like QAOA, but also quantum annealers or neutral atom devices. It is particularly intriguing to investigate the algorithm's effectiveness on higher dimensional lattices and explore whether the suspicion that the performance of QICA deteriorates with increasing lattice dimension holds true. Hopefully, the proposed techniques can help to alleviate this issue and potentially yield an algorithm that outperforms parallel tempering even on the Wishart lattice.

Finally, the QICA raises the broader question if it may be useful to use quantum computing to provide information that is difficult to generate classically, to inform and improve classical heuristic solvers, instead of using the hybrid quantum-classical approach to directly solve the problem of interest.

Appendix

Parallel tempering and simulated annealing setup

For a Max-2-SAT problem with n variables simulated annealing has performed $600 \cdot n$ update steps. The annealing schedule consists of evenly spaced temperatures in $\beta = \frac{1}{T}$, with a starting temperature $\frac{1}{T} = \beta = 0$ and a final temperature of $\frac{1}{T} = \beta = 6$.

The same annealing has been chosen for the Ising-like problems generated via the Chook library. Only the number of update steps diverges. These are specified in each simulated problem.

In the parallel tempering simulations 24 replicas with a random configuration have been initialized. The temperatures of the replicas are evenly spaced between the replica with the lowest temperature $T = 0.1$ and the highest with $T = 2.4$. 2000 simulation cycles have been carried out, while one simulation cycle consists of n update steps per replica followed by one replica exchange. In all simulations in which parallel tempering has acted as a benchmark, for each solver only in less than 0.2% of the instances it has been outperformed.

The chosen parameters for the problems generated via the Chook library are identical, but 16 replicas have been used instead of 24 and the number of cycles varies for the different problems but are specified for each in the respective section.

More Error Analysis

Similar to the discussion in 2.1.2 the plots in 2.24 show where depth $p = 1$ RQAOA made the wrong decision (same definition as in 2.1.2) for 100 randomly generated Max-2-SAT problems with $n = 60$ variables and $n_c = 5$. However, in this figure, the bars show which of the 55 elimination steps (chronologically ordered) was flawed. It can be seen that generally, the wrong decisions happen over the whole course of the algorithm, at earlier but also at later elimination steps. While for the clause-to-variable ratio $\alpha = 2$ there are two weight concentrations, one at around elimination step 10 and another one after 40, for higher clause-to-variable ratios the occurrences of wrong decisions are shifting towards earlier elimination steps.

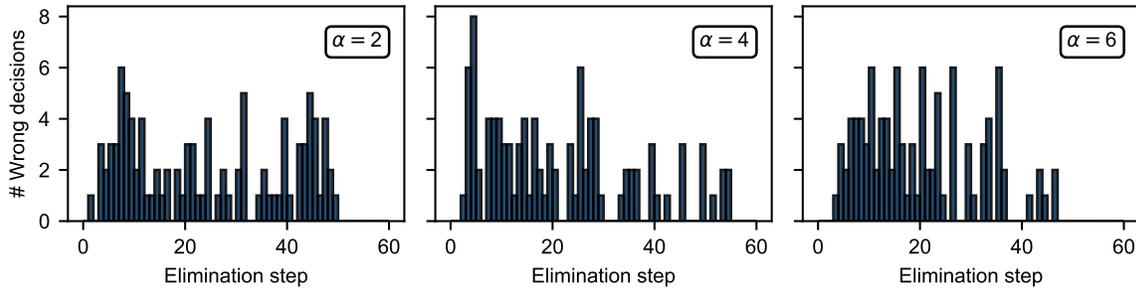


Figure 2.24: Wrong decisions dependent on the number of update step in RQAOA. For three different clause-to-variable ratios α , 100 random Max-2-SAT instances with $n = 60$ variables and three different clause-to-variable ratios α were generated on which level $p = 1$ RQAOA ($n_c = 5$) failed to produce the optimal solution. All optimal solutions for a certain instance were determined with the RC2 solver. A wrong decision is defined as in 2.1.2. The bars indicate at which of the 55 ($n - n_c$) elimination steps RQAOA performed enforced a wrong correlation constraint on the solution.

Bibliography

- [1] A. Abramé und D. Habet, *Journal on Satisfiability, Boolean Modeling and Computation* **9** (2015), 89.
- [2] D. Achlioptas, A. Coja-Oghlan und F. Ricci-Tersenghi, *Random Structures and Algorithms* **38** (2011), 251.
- [3] V. Akshay, H. Philathong, E. Campos, D. Rabinovich, I. Zacharov, X.M. Zhang und J.D. Biamonte, *Physical Review A* **106** (2022), 042438.
- [4] V. Akshay, H. Philathong, M. Morales und J. Biamonte, *Physical Review Letters* **124** (2020), 090504.
- [5] B. Aspvall, M.F. Plass und R.E. Tarjan, *Information Processing Letters* **8** (1979), 121.
- [6] E. Bae und S. Lee, (2022).
- [7] L. Bittel und M. Kliesch, *Physical Review Letters* **127** (2021), 120502.
- [8] B. Bollobás, C. Borgs, J.T. Chayes, J.H. Kim und D.B. Wilson, *Random Structures and Algorithms* **18** (2001), 201.
- [9] B. Borchers und J. Furman, *Journal of Combinatorial Optimization* **2** (1998), 299.
- [10] A. Braunstein, M. Mézard und R. Zecchina, *Random Structures and Algorithms* **27** (2005), 201.
- [11] S. Bravyi, A. Kliesch, R. Koenig und E. Tang, *Phys. Rev. Lett.* **125**, 260505 (2020) (2019).
- [12] S. Bravyi, A. Kliesch, R. Koenig und E. Tang, *Quantum* **6** (2022), 678.
- [13] J.G. C. Anstegui, *CPAIOR 2013* (2013).
- [14] L. Caha, A. Kliesch und R. Koenig, *Quantum Science and Technology* **7**, 045013 (2022) (2022).

- [15] S. Chib und E. Greenberg, *The American Statistician* **49** (1995), 327.
- [16] C.N. Chou, P.J. Love, J.S. Sandhu und J. Shi, (2021).
- [17] J.P. Chu Min Li, F. Many, (2005).
- [18] A. Coja-Oghlan. In *Automata, Languages and Programming*. Springer Berlin Heidelberg (2009), Seiten 292–303.
- [19] S.A. Cook: *The complexity of theorem-proving procedures. The complexity of theorem-proving procedures*, In *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*. ACM Press (1971).
- [20] W. Dean: *Computational Complexity Theory*. Metaphysics Research Lab, Stanford University, 2021.
- [21] D.J. Earl und M.W. Deem, *Physical Chemistry Chemical Physics* **7** (2005), 3910.
- [22] S.G. Edward Farhi, David Gamarnik, (2020).
- [23] R.G. Edwards und A.D. Sokal, *Physical Review D* **38** (1988), 2009.
- [24] E. Farhi, D. Gamarnik und S. Gutmann. *The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: A Typical Case*, 2020.
- [25] E. Farhi, J. Goldstone und S. Gutmann, (2014).
- [26] E. Farhi und A.W. Harrow. *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*, 2016.
- [27] E. Friedgut und appendix by Jean Bourgain, *Journal of the American Mathematical Society* **12** (1999), 1017.
- [28] Z. Fu und S. Malik. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2006), Seiten 252–265.
- [29] D. Gamarnik, *PNAS* (2021).
- [30] D. Gamarnik und M. Sudan, *SIAM Journal on Computing* **46** (2017), 590.
- [31] M.R. Garey: *Computers and intractability*. W. H. Freeman, 1979.
- [32] M.X. Goemans und D.P. Williamson, *Journal of the ACM* **42** (1995), 1115.
- [33] G.R. Grimmett. In *Probability on Graphs*. Cambridge University Press (2006), Seiten 156–188.
- [34] L.K. Grover: *A fast quantum mechanical algorithm for database search. A fast quantum mechanical algorithm for database search*, In *Proceedings of the twenty-*

- eighth annual ACM symposium on Theory of computing - STOC '96*. ACM Press (1996).
- [35] M.E. Halaby, *Electron. Colloquium Comput. Complex.* (2016).
- [36] F. Hamze, D.C. Jacob, A.J. Ochoa, D. Perera, W. Wang und H.G. Katzgraber, *Physical Review E* **97** (2018), 043303.
- [37] F. Hamze, J. Raymond, C.A. Pattison, K. Biswas und H.G. Katzgraber, *Phys. Rev. E* **101**, 052102 (2020) **101** (2019), 052102.
- [38] M.B. Hastings, (2019).
- [39] D. Henderson, S.H. Jacobson und A.W. Johnson. In *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Seiten 287–319.
- [40] J. Houdayer, *The European Physical Journal B* **22** (2001), 479.
- [41] A. Ignatiev, A. Morgado und J. Marques-Silva, *Journal on Satisfiability, Boolean Modeling and Computation* **11** (2019), 53.
- [42] H. Karloff und U. Zwick: *A 7/8-approximation algorithm for MAX 3SAT? A 7/8-approximation algorithm for MAX 3SAT?*, In *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc.
- [43] S. Kirkpatrick und B. Selman, *Science* **264** (1994), 1297.
- [44] F. Krzcakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian und L. Zdeborova, *Proceedings of the National Academy of Sciences* **104** (2007), 10318.
- [45] C.M. Li und F. Many, *Frontiers in Artificial Intelligence and Applications* **185** (2021), 613.
- [46] M.A.V. Marc Mezard, Giorgio Parisi: *Spin Glass Theory and Beyond*. World Scientific, 1987.
- [47] R. Marino, G. Parisi und F. Ricci-Tersenghi, *Nature Communications* **7** (2016).
- [48] J.R. McClean, M.P. Harrigan, M. Mohseni, N.C. Rubin, Z. Jiang, S. Boixo, V.N. Smelyanskiy, R. Babbush und H. Neven, *PRX Quantum* **2** (2021), 030312.
- [49] M. Mezard, T. Mora und R. Zecchina, *Phys. Rev. Lett.* **94**, 197205 (2005) **94** (2005), 197205.
- [50] M. Mohseni, D. Eppens, J. Strumpfer, R. Marino, V. Denchev, A.K. Ho, S.V. Isakov, S. Boixo, F. Ricci-Tersenghi und H. Neven. *Nonequilibrium Monte Carlo for unfreezing variables in hard combinatorial optimization*, 2021.

- [51] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman und L. Troyansky, *Nature* **400** (1999), 133.
- [52] R. Niedermeier und P. Rossmanith, *Journal of Algorithms* **36** (2000), 63.
- [53] M.Y. Niu, S. Lu und I.L. Chuang, (2019).
- [54] A. Ozaeta, W. van Dam und P.L. McMahon, *Quantum Science and Technology* **7** (2022), 045036.
- [55] G. Parisi. In *World Scientific Lecture Notes in Physics*. WORLD SCIENTIFIC (nov 1986), Seiten 157–163.
- [56] Y.J. Patel, S. Jerbi, T. Bck und V. Dunjko, (2022).
- [57] D. Perera, I. Akpabio, F. Hamze, S. Mandra, N. Rose, M. Aramon und H.G. Katzgraber. *Chook – A comprehensive suite for generating binary optimization problems with planted solutions*, 2020.
- [58] D. Perera, F. Hamze, J. Raymond, M. Weigel und H.G. Katzgraber, *Physical Review E* **101** (2020), 023316.
- [59] F. Romá, S. Risau-Gusman, A. Ramirez-Pastor, F. Nieto und E. Vogel, *Physica A: Statistical Mechanics and its Applications* **388** (2009), 2821.
- [60] S.H. Sack, R.A. Medina, A.A. Michailidis, R. Kueng und M. Serbyn, *PRX Quantum* **3** (2022), 020365.
- [61] P.W. Shor, *SIAM Review* **41** (1999), 303.
- [62] W. Wang, J. Machta und H.G. Katzgraber, *Physical Review E* **92** (2015), 013303.
- [63] J. Watrous. In *Computational Complexity*. Springer New York (2012), Seiten 2361–2387.
- [64] U. Wolff, *Physical Review Letters* **62** (1989), 361.
- [65] L. Zhou, S.T. Wang, S. Choi, H. Pichler und M.D. Lukin, *Physical Review X* **10** (2020), 021067.
- [66] Z. Zhu, A.J. Ochoa und H.G. Katzgraber, *Physical Review Letters* **115** (2015), 077201.