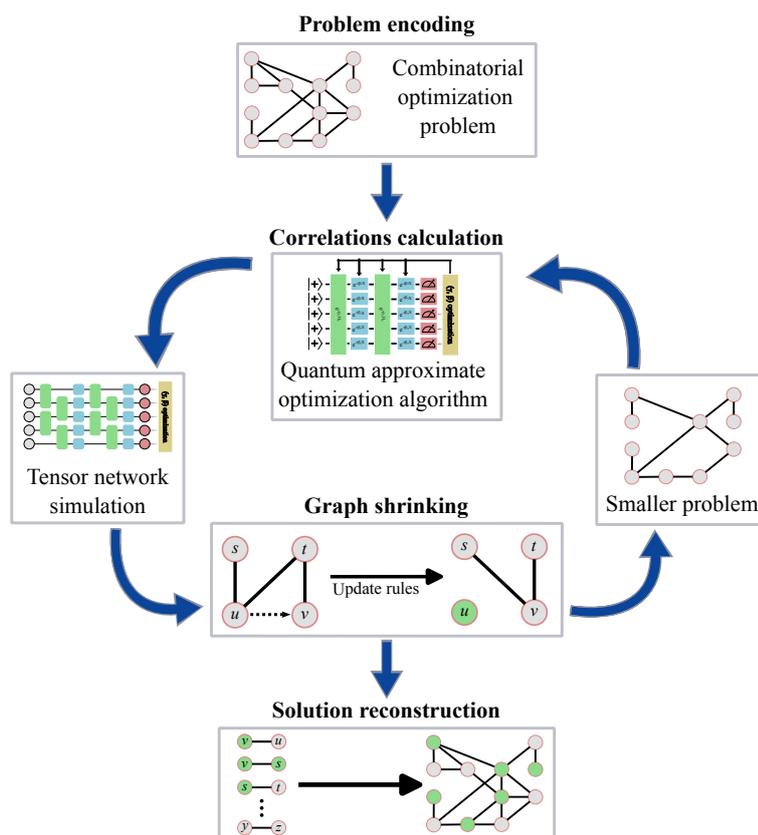


Tensor Network Simulations of Higher Depth Recursive Quantum Optimization Algorithms

Master Thesis in Quantum Science and Technology

by

Maximilian Passek



First Referee: Prof. Dr. Christian Mendl

Second Referee: PD Dr. Jeanette Miriam Lorenz

Supervisor: Jernej Rudi Finžgar

Submission date: 15. July 2024

Abstract

Combinatorial optimization problems (COPs) arise as important issues in various areas such as finance, telecommunication or industry. Specific examples in industry are supply chain management, logistics or production planning. Since it remains challenging to solve them with classical methods, in recent years much effort has been put into developing quantum algorithms to tackle these generally NP-hard problems. A promising family of these quantum routines consists of recursive quantum optimization algorithms (RQOAs). Although these algorithms display good results it remains unclear whether quantum approaches can outperform classical solvers. Therefore, this work investigates the performance of RQOAs. Specifically, to explore the role of quantum correlations in these recursive algorithms, classical as well as quantum variations of the recursive quantum approximate optimization algorithm are applied to MAXCUT instances of different densities. Furthermore, various quantum-informed recursive optimization (QIRO) routines based on the quantum approximate optimization algorithm (QAOA) with up to depths $p = 3$ are used to solve maximum independent set problems of sizes up to 160. This allows examining the performance of QIRO on relatively large problems with higher quality quantum correlations. Lastly, the SETCOVER problem is identified as a real world COP from industry, a QIRO algorithm is developed to solve it and first experiments thereof are conducted. Throughout the thesis, higher depth QAOAs are simulated using advanced tensor network techniques. The results of this work indicate that the recursive nature of RQOAs itself already improves the performance of the underlying routines for obtaining correlations. Furthermore, the experiments show the strong performance increase of RQOAs for larger problem instances when using correlations obtained from higher depth QAOAs. This result emphasizes the importance of the quality of quantum correlations in RQOAs. The investigations on the SETCOVER problem highlight the current feasibility of simulating quantum routines for challenging and industry-relevant problems. However, the used quantum algorithm performs worse than a simple classical greedy solver, showing the limitations of currently available quantum algorithms. Nonetheless, also in this case, quantum correlations of higher quality help to improve the performance.

Contents

| | |
|---------------------------------------------------------------------------------------------------|-----|
| Abstract | iii |
| Dedication and declaration | vii |
| Abbreviations | ix |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Goal of the thesis | 4 |
| 1.3 Outline | 4 |
| 2 Background | 7 |
| 2.1 Problem classes | 7 |
| 2.1.1 Quadratic unconstrained binary optimization (QUBO) | 7 |
| 2.1.2 MAXCUT problem | 8 |
| 2.1.3 Maximum independent set problem (MIS) | 10 |
| 2.1.4 Set cover problem | 13 |
| 2.2 Tensor networks | 15 |
| 2.2.1 Basic concepts of tensor networks | 15 |
| 2.2.2 Tensor operations | 16 |
| 2.2.3 Types of tensor networks | 17 |
| 2.2.4 Tensor networks for simulating quantum circuits | 18 |
| 2.3 Quantum optimization algorithms (QOAs) | 19 |
| 2.3.1 Quantum approximate optimization algorithm (QAOA) | 19 |
| 2.3.2 Recursive quantum approximate optimization algorithm (RQAOA) | 23 |
| 2.3.3 Quantum-informed recursive optimization algorithm (QIRO) for MIS problems | 24 |
| 2.3.4 Quantum-informed recursive optimization algorithm (QIRO) for SETCOVER problems | 28 |
| 3 Analysis of tensor network simulations of QAOA | 31 |
| 3.1 Simulation of higher depths QAOAs using tensor networks | 31 |
| 3.1.1 Encoding of QAOA as tensor network | 31 |
| 3.1.2 Quantum circuit simulation by tensor network contraction | 32 |

| | | |
|----------|--------------------------------------------------------------------------------------------------------------|-----------|
| 3.2 | Different methods of QAOA parameter initialization | 34 |
| 3.2.1 | Transition states initialization | 35 |
| 3.2.2 | Interpolation initialization | 36 |
| 3.2.3 | Fixed angles initialization for MAXCUT problems on k -regular graphs | 37 |
| 3.3 | Parameter optimization | 37 |
| 3.3.1 | MAXCUT parameter optimization | 38 |
| 3.3.2 | MIS parameter optimization | 41 |
| 3.3.3 | SETCOVER parameter optimization | 42 |
| 4 | Application of recursive quantum optimization (RQO) algorithms to different problem classes | 43 |
| 4.1 | Variations of RQAOA for the MAXCUT problem | 43 |
| 4.1.1 | Linear programming (LP) and semidefinite programming (SDP) correlations for MAXCUT problems | 43 |
| 4.1.2 | Experiments on Erdős-Rényi graphs | 44 |
| 4.1.3 | Experiments on 3-regular graphs with higher depth QAOA | 49 |
| 4.2 | Higher-depth variations of QIRO for the MIS problem | 51 |
| 4.2.1 | Experimental framework | 51 |
| 4.2.2 | Results | 52 |
| 4.2.3 | Discussion | 53 |
| 4.3 | Industry use case: Application of a QIRO algorithm to a sensor positioning problem | 55 |
| 4.3.1 | Experimental framework | 55 |
| 4.3.2 | Results | 56 |
| 4.3.3 | Discussion | 56 |
| 5 | Summary, conclusion and outlook | 59 |
| 5.1 | Summary | 59 |
| 5.2 | Conclusion | 60 |
| 5.3 | Outlook | 61 |
| A | Ising formulation of the SetCover problem | 63 |
| B | QAOA parameter analysis for MaxCut problems of sizes $n \in \{100, 150, 200\}$ | 65 |
| C | QAOA parameter analysis for MIS problems of sizes $n \in \{100, 150, 200\}$ | 73 |
| | Bibliography | 79 |

Dedication and declaration

Dedication

I would like to take this opportunity to thank everyone who directly or indirectly contributed to the success of this work. My special thanks go to:

Prof. Dr. Christian Mendl for making this work possible and supervising it.

PD Dr. Jeanette Miriam Lorenz for co-supervising the thesis.

Very most of all Jernej Rudi Finžgar for the year of working together on this thesis, for always giving great ideas and input, for inspiring and encouraging discussions, for teaching me a lot in research and life, for driving me to *pick the low hanging fruits* in the first week, for being available at really all times to answer my silly questions related to both work and everyday life, for giving good advice on all topics, for showing understanding regarding mistakes and life outside of work, for letting me surpass him in kicktipp, or to make it short, for being the best supervisor and partner in crime I can think of.

Victor Fischer, Jasper Stoffel, Johannes Frank, Johannes and Mirjam Bitzinger und Moritz Harkort for encouraging discussions and adventures regarding this work at day and night time.

Very much the Emmeringer Innovatoren (FG-160) and SAP Transformation Teams of BMW for providing such a welcoming, interesting, helpful, cordial and especially *agile* environment to work, discuss, develop skills in research, in presenting, in *agility*, in basically *winning* soccer tournaments and in foosball, and especially to laugh and enjoy lunches. I want to particularly thank Andre Luckow, Ann Christin Rathje, Johannes Klepsch, Lukas Müller, Philipp Ross, Carlos Riofrio, Marvin Erdmann, Christian Degel (oder Schick?), Flo Kiwit, Josef Pichlmeier, Leo Hölscher and whoever I forgot to mention.

My CAAD13 for overcoming frustrating times.

And my family for their never ending commitment to encourage me.

Declaration

I hereby confirm that I have prepared the present work independently and without outside help. The used literature and other aids are completely indicated.

Munich, July 15, 2024

Maximilian Passek

Abbreviations

Throughout the thesis the following abbreviations apply:

| | |
|-------|------------------------------------------------------|
| COP | Combinatorial Optimization Problem |
| Eq. | Equation |
| Fig. | Figure |
| GW | Goemans-Williamson |
| LP | Linear Programming |
| MERA | Multiscale Entanglement Renormalization Ansatz |
| MIS | Maximum Independent Set |
| MPS | Matrix Product States |
| PEPS | Projected Entangled Pair States |
| QAOA | Quantum Approximate Optimization Algorithm |
| QIRO | Quantum-Informed Recursive Optimization |
| QOA | Quantum Optimization Algorithm |
| QUBO | Quadratic Unconstrained Binary Optimization |
| Ref. | Reference |
| RQAOA | Recursive Quantum Approximate Optimization Algorithm |
| RQO | Recursive Quantum Optimization |

Abbreviations

| | |
|------|------------------------------|
| Sec. | Section |
| SDP | Semidefinite Programming |
| SVD | Singular Value Decomposition |
| TTN | Tree Tensor Network |

Chapter 1

Introduction

In this introductory chapter I give a short explanation of the recent developments in solving combinatorial optimization problems (COPs) with the help of quantum routines. Based on this prior research on recursive quantum optimization (RQO) algorithms, the motivation behind this thesis is given. In addition, I define the goals and describe the outline of this work.

1.1 Motivation

Combinatorial optimization problems (COPs) are highly relevant in diverse sectors across both science and industry, including logistics, finance, telecommunication, and machine learning [1]. These problems involve finding an optimal solution from a finite set of solutions. Achieving this is often NP-hard, making COPs challenging to solve efficiently as the size of a problem increases. The complexity arises due to the exponential growth of the solution space with increasing problem size, which makes exhaustive classical search impractical for large instances.

A prototypical example of practical relevance is the so-called maximum independent set (MIS) problem. For a given problem graph consisting of vertices and edges, the goal is to find a set of vertices with maximum size such that there is no edge between two elements of the set. An example graph and a corresponding MIS can be seen in part a) and b) of Fig. 1.1, respectively. The maximum independent set has a size of six vertices and is obtained by a search of all possible solutions.

Traditionally, a variety of classical strategies have been employed to tackle COPs. The simplest methods are heuristic greedy algorithms that rely on intuitive approaches but often achieve surprisingly good results. However, they frequently lack a mathematical reasoning and proven performance bounds. More advanced classical solving ideas are based on for example linear programming (LP) [2] and semidefinite programming (SDP) relaxations [3]. These methods work by relaxing the original combinatorial problem into a more tractable form and then finding approximate solutions. While these approaches have been successful in many applications, they often face limitations in terms of scalability and efficiency, especially for large-scale problems.

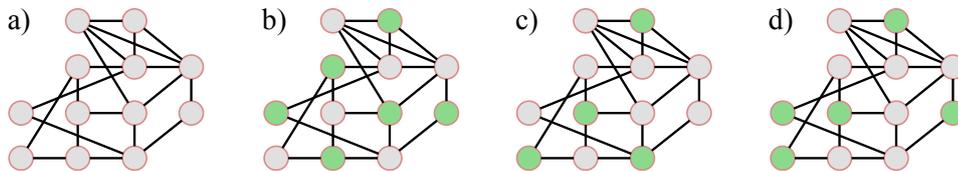


Figure 1.1: In a) an example of a graph representation of a maximum independent set problem is shown, in b) a maximum independent set of size six colored in green, in c) a non-maximum independent set of size four found by the greedy MIN algorithm and in d) a non-maximum independent set of size five obtained by a quantum-informed recursive optimization routine with low quality quantum information.

For the MIS problem, a widely used greedy algorithm is the MIN routine. It fixes iteratively one of the vertices with the lowest number of edges to be in the independent set and deletes all vertices that are connected to the chosen vertex by an edge. For the example of Fig. 1.1 it occurs that the MIN algorithm does not solve the problem optimally but finds an independent set with only four vertices as shown in illustration c) of the Figure. Of course the given example is hand-picked but nevertheless indicates that already for simple and small problems classical routines often reach their limitations.

In recent years, quantum computing has emerged as a promising alternative for solving COPs. Quantum algorithms have the potential to leverage the principles of superposition and entanglement to explore solution spaces more efficiently than classical algorithms. Among the various quantum algorithms proposed, the quantum approximate optimization algorithm (QAOA) [4] has earned significant attention. QAOA is a hybrid quantum-classical algorithm designed to find approximate solutions to unconstrained binary optimization problems. It is particularly suitable for implementation on noisy intermediate-scale quantum (NISQ) devices, which are the current generation of quantum computers.

However, the practical performance of NISQ algorithms, including QAOA, in providing speedups over classical methods for solving COPs remains an open question. Several challenges have been identified with near-term variational approaches, such as barren plateaus [5], which refer to regions in the optimization landscape where the gradient is nearly zero, making training difficult. Additionally, the inherent noise of quantum devices [6] poses a significant challenge to the reliable execution of quantum algorithms. Another significant limitation of QAOA is its locality: At any constant circuit depth, only qubits that are separated by less than a certain distance in the graph representation of a problem can interact. This locality limitation of QAOA has been demonstrated in various studies [7–10].

To address the locality-induced limitations of QAOA, the recursive QAOA (RQAOA) was introduced by Bravyi *et al.* [7, 11]. RQAOA operates by recursively

simplifying the problem based on correlations between problem variables obtained from QAOA. The process involves iteratively computing these correlations and fixing the variables until the problem is fully solved. This method introduces new connections between previously unlinked variables, effectively overcoming the locality constraint of QAOA. Extensions and generalizations of RQAOA have been proposed, including analog quantum devices [12] and classical correlation-based shrinking procedures [13]. These recursive algorithms have shown improved performance over the original QAOA for many problem instances [11, 14, 15]. However, it remains uncertain whether the enhanced performance is due to the recursive shrinking procedure or the quantum correlations. Therefore, it would be interesting to further investigate which role correlations obtained from quantum routines play in these recursive algorithms.

Building upon RQAOA, a new family of hybrid quantum-classical algorithms, termed quantum-informed recursive optimization (QIRO) routines, has been introduced by Finžgar *et al.* [12] and further developed by other researchers [16]. In QIRO, information generated by quantum resources is used to recursively reduce the size of the optimization problem through problem-specific classical optimization routines. This approach leverages decades of research in classical combinatorial optimization, allowing the tailoring of classical subroutines to specific optimization problems, thereby enhancing the algorithm's performance [12]. Coming back to the MIS problem of Fig. 1.1, the QIRO algorithm as introduced in Ref. [12] using $p = 1$ QAOA is able to outperform the greedy MIN routine by finding an independent set of size five as shown in part d) of the Figure. However, it requires correlations from $p = 2$ QAOA to find the optimal MIS of size six. This finding already reveals the importance of correlations obtained from higher depth QAOA, thus of high quality quantum correlations. Conventionally simulating higher depth QAOA circuits for large problems is computationally very demanding. However, recent developments in using tensor networks enable simulations of larger scale as before.

These prior works suggest that QIRO algorithms are promising tools for solving COPs. However, experiments were only conducted on small system sizes [16] or with low quality quantum correlations [12]. Thus, it is of interest to investigate the performance of QIRO routines on larger problem sizes with quantum data of higher quality to gain better insight into the capabilities of QIRO.

Apart from the promising results quantum optimization algorithms demonstrate in general, critics often include that the issues solved by quantum routines are small toy problems only designed for showing the potential of quantum algorithms [17]. Therefore, successfully applying recursive quantum algorithms to real world problems occurring in industry would be a step forward into the direction of overcoming sceptics on quantum computing.

1.2 Goal of the thesis

This thesis aims to build on the above introduced developments in RQO algorithms. On the one hand the goal is to investigate the role of correlations obtained from quantum routines within shrinking routines to gain a better understanding on the underlying mechanisms of RQO algorithms.

In addition, the thesis further explores the performance of QIRO routines in solving large-scale COPs. Here, the goal is to solve problems of larger size than usual quantum experiments and maintaining high quality quantum correlations by integrating advanced tensor network techniques to calculate quantum information. Therefore, the thesis aims to push the boundaries of what can be achieved with hybrid quantum-classical optimization algorithms.

A third goal of the thesis consists of tracking down a real world use case of a COP that is of relevance in industry. A suitable RQO algorithm for this problem should be developed and applied. Thus, one objective is to bring the family of recursive quantum optimization algorithms closer to solving legitimate problems as they are of high relevance in industry.

1.3 Outline

The remainder of the thesis is organized as follows. In Chapter 2 the background information is explained, that is necessary to understand the tasks of this work. Therefore, the examined COPs are introduced, the MAXCUT, MIS and SETCOVER problems. Subsequently, a short overview of tensor networks and how they can be used for simulating quantum circuits is given in Sec. 2.2.4. Furthermore, different quantum optimization algorithms are introduced in Part 2.3, namely the QAOA, RQAOA and the QIRO family. In the succeeding Chapter 3 the background knowledge is used to analyse tensor network simulations of QAOA circuits on the present problems. This analysis allows finding suitable parameter initialization methods and optimizers for achieving good QAOA results. In Chapter 4 these parameter initialization methods and optimizers are used to apply variations of RQAOA to solve MAXCUT instances with various densities. This allows analysing the role of quantum correlations within recursive shrinking algorithms. Furthermore, experiments are conducted on QIRO algorithms with high quality quantum correlations for solving large scale MIS problems in order to further investigate the performance of these routines. In addition, a QIRO algorithm for the SETCOVER problem is developed and simulated to solve the real world problem of positioning sensors in a production plant. In the last Chapter 5, a summary, conclusion and outlook is given.

Before continuing, it is noted that throughout the whole thesis, the performed calculations and programming is done in the programming language *Python* and

its libraries. Furthermore, it is important to clarify that all experiments of this work were run on classical computers. Whenever it is talked about the execution of a quantum algorithm, actually a classical simulation of the program running on a quantum computer is meant.

Chapter 2

Background

This chapter explains the background knowledge, which is necessary to understand the content of the thesis. First, I introduce different problem classes that are studied throughout the thesis. Then the basics of tensor networks are described before an overview over different established quantum optimization algorithms is provided. The explanations are given at a level comprehensible to readers with at least the background of a graduate physics student having knowledge in quantum computing and information theory. If the reader is familiar with the mentioned topics, he may skip parts of this chapter.

2.1 Problem classes

Beginning this section, I describe quadratic unconstrained binary optimization (QUBO) in general before explaining three explicit examples thereof, namely the MAXCUT, MIS and the SETCOVER problems. For each of these I will give the QUBO and Ising formulation as well as a classical way of solving the problem.

2.1.1 Quadratic unconstrained binary optimization (QUBO)

Many of the optimization problems that are of importance in practical applications can be formulated as mathematical models based on a cost function with quadratic binary variables x_i s [18]. Examples are very diverse and range from portfolio optimization in finance over supply chain optimization in industry to traffic management in the public sector [19, 20]. These so-called quadratic unconstrained binary optimization (QUBO) problems are also of utter importance in physics where they occur naturally as Ising models with only one- and two-body interactions between spins [21]. Solving QUBO instances consists of finding the set of x_i s that minimize the cost function. Due to the widespread appearing of these problems, next to many classical solvers such as exact, approximate or heuristic solvers with performance guarantees [22] also quantum approaches have been developed in recent years.

Mathematically formulated, a QUBO corresponds to minimizing a quadratic function C with N binary variables $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \{0, 1\}^N$ over all possible 2^N

assignments [18]:

$$C(\mathbf{x}) = J + \sum_{i=1}^N J_i x_i + \sum_{i=1}^N \sum_{j<i} J_{ij} x_i x_j, \quad (2.1)$$

where J stands for a constant offset, J_i and J_{ij} represent constant prefactors. Often it is helpful to transform this formulation to an equivalent one with spin-like variables $z_i \in \{-1, 1\}$ instead of the binaries x_i s, thus to an Ising formulation.

2.1.2 MaxCut problem

Here the MAXCUT problem is defined and mathematically formulated. Furthermore, I introduce two classical solvers: Linear programming (LP) and the Goemans-Williamson (GW) algorithm.

Definition and QUBO formulation

One prime example of a QUBO is the so-called MAXCUT problem. It consists of a weighted undirected graph $G = (V, E)$ with N vertices $V = \{i\}$, edges $E = \{e\}$ and edge weights $\{w_e\}$. The present task is to find a subset of nodes such that the summed up weights of the edges that connect the chosen subset and its complement are maximized. More mathematically speaking, the aim is to look for a partition $W \subseteq V$ of all nodes that maximizes the weight $\sum_{e \in \delta(W)} w_e$ of the edge set $\delta(W) := \{ij \in E \mid i \in W, j \in V \setminus W\}$ [23]. An example of a small MAXCUT problem with all weights $w_e = 1$ and its solution is shown in Fig. 2.1.



Figure 2.1: In a) an example of a graph representation of a MAXCUT problem with all weights $w_e = 1$ is shown and in b) a possible solution colored in green.

Although the structure of the MAXCUT is simple, it is one of the most studied COPs because every QUBO problem can be transformed into a MAXCUT with an overhead of only one extra vertex and edges between this vertex and the original edges [24–27]. Also, solving MAXCUT instances for dense graphs with hundreds of vertices generally already overstrains state-of-the-art algorithms [28] since it is proven to be NP-hard [29].

The cost function optimization of the MAXCUT QUBO formulation in terms of

binary variables $x_i \in \{0, 1\}$ is given by

$$\min_{\mathbf{x}} C_{\text{MAXCUT}}(\mathbf{x}) = \min_{\mathbf{x}} \left(- \sum_{ij \in E} w_{ij} (x_i - x_j)^2 \right) = \min_{\mathbf{x}} \sum_{ij \in E} w_{ij} (2x_i x_j - x_i - x_j), \quad (2.2)$$

where $x_i = 1$ stands for the vertex i included in the partition W and vice versa. The intuition behind this cost function is clear given that the minimization is fulfilled when the sum of weights of edges between the two different partitions is maximized. However, in the case of MAXCUT it is more natural to think of of the QUBO formulation with Ising variables z_i . This is achieved by inserting the mapping $x_i = \frac{1}{2}(1 + z_i)$ into equation (2.2):

$$\min_{\mathbf{z}} C_{\text{MAXCUT}}(\mathbf{z}) = \min_{\mathbf{z}} \frac{1}{2} \sum_{ij \in E} w_{ij} (z_i z_j - 1). \quad (2.3)$$

Here, $\mathbf{z} \in \{1, -1\}^N$, and z_i indicates whether vertex i is in the subset W or not.

Solving MaxCut by linear programming (LP)

MAXCUT problems can be solved classically by linear programming (LP) in the following way. For a weighted undirected graph $G = (V, E)$, an edge-incidence vector $\mathbf{y} \in \{0, 1\}^{|E|}$ is defined such that $y_e = 0$ indicates that edge e is not cut, while $y_e = 1$ indicates that edge e is cut.

The MAXCUT problem can be formulated as an integer linear program, given by [30–33]:

$$\max \sum_{e \in E} \omega_e y_e, \quad (2.4a)$$

$$\text{s.t.} \quad \sum_{e \in Q} y_e - \sum_{e \in C \setminus Q} y_e \leq |Q| - 1, \quad (2.4b)$$

$$|Q| \text{ odd, } \forall Q \subseteq C \text{ cycle,}$$

$$0 \leq y_e \leq 1, \quad \forall e \in E, \quad (2.4c)$$

$$y_e \in \{0, 1\}, \quad \forall e \in E, \quad (2.4d)$$

Here, Eqs. (2.4b) are known as the odd-cycle inequalities, which are sufficient to define a cut. Although solving the MAXCUT problem using Eqs. (2.4a), (2.4b), and (2.4d) is generally intractable, the linear relaxation defined by Eqs. (2.4a)–(2.4c) can be efficiently solved in polynomial time. This relaxation, often addressed via

branch-and-bound techniques, provides an upper bound on the maximum cut [28]. If the relaxation results in an integer solution, the optimal solution to the problem is immediately obtained.

Solving MaxCut with the Goemans-Williamson (GW) algorithm

Another approach to solve MAXCUT problems is the so called Goemans-Williamson (GW) algorithm based on semidefinite programming (SDP) [3]. It is the leading polynomial-time approximation algorithm for MAXCUT problems. The routine approximates the NP-hard integer quadratic formulation of MAXCUT (as shown in Eq. (2.2)) by a relaxed problem that expands the solution space to include all feasible solutions of the integer problem, thus providing an upper bound on the integer problem's optimal solution.

In this relaxation, integer variables are substituted with multi-dimensional vectors $\mathbf{v}_i \in \mathbb{R}^N$, normalized to the $(N - 1)$ -dimensional unit sphere S_{N-1} . The relaxed optimization problem is formulated as:

$$\begin{aligned} \max_{\{\mathbf{v}_1, \mathbf{v}_2, \dots\}} \quad & \frac{1}{2} \sum_{i < j} w_{ij} (1 - \mathbf{v}_i \cdot \mathbf{v}_j), \\ \text{s.t.} \quad & \mathbf{v}_i \in S_{N-1} \quad \forall i \in V. \end{aligned} \tag{2.5}$$

This relaxed formulation corresponds to the semidefinite program:

$$\max_X \left\{ \left\langle \frac{1}{4} L, X \right\rangle : \text{diag}(X) = \mathbf{e}, X \succeq 0 \right\}, \tag{2.6}$$

where $L := \text{diag}(A\mathbf{e}) - A$ is the graph Laplacian and \mathbf{e} is the all-ones vector. The inner product $\langle a, b \rangle = \text{tr}(ab^T)$ represents the Frobenius inner product of matrices $a, b \in \mathbb{R}^{N \times m}$. This generic semidefinite program can be solved in polynomial time using various solvers such as `cvxopt` [34–36].

The matrix X represents the Gram matrix of the vectors $\{\mathbf{v}_i\}$. To retrieve the vectors, one computes the Cholesky decomposition of X : $X = B^T B$, which has a computational complexity of $O(N^3)$. In the resulting matrix B , the i th column corresponds to the vector \mathbf{v}_i .

2.1.3 Maximum independent set problem (MIS)

In this part, the maximum independent set (MIS) and its QUBO formulation are introduced. In addition, I explain two classical greedy solvers, MIN and MAX.

Definition and QUBO formulation

In research, the MIS is another stereotypical example of a QUBO with a wide range of applications in various fields, such as network design, portfolio optimization, scheduling, resource allocation, and many more [37, 38]. Generally speaking, finding a MIS solution is NP-hard [29].

In the language of graph theory, finding an independent set of an undirected graph $G = (V, E)$ with N vertices V and edges E consists of locating a subset of vertices $I \subseteq V$ such that no two vertices in the set are connected by an edge. The subsets I with the highest cardinality are called a maximum independent set of the given graph. The MIS problem is finding such a maximum subset. An example of a graph, a non-maximum and a maximum independent set thereof are illustrated in Fig. 2.2.

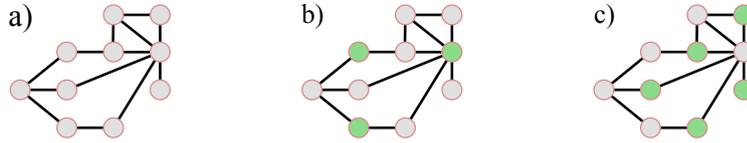


Figure 2.2: In a) an example of a graph representation of a MIS problem is shown, in b) a non-maximum independent set colored in green and in c) a maximum independent set colored in green.

Using the same logic as in the case of MAXCUT, a QUBO cost function can be constructed with $\mathbf{x} \in \{0, 1\}^N$ indicating whether the vertices are included in the MIS or not:

$$\min_{\mathbf{x}} C_{\text{MIS}}(\mathbf{x}) = \min_{\mathbf{x}} \left(-\sum_{i=1}^N x_i + \alpha \sum_{ij \in E} x_i x_j \right), \quad (2.7)$$

where α is a constant. The intuition behind is maximizing the number of included nodes $\sum_i x_i$ under the constraint that no neighbouring vertices are allowed in the subset. To make sure that the constraint is fulfilled, $\alpha > 1$ is required. Since I will be using the Ising formulation of the problem later in this thesis, I introduce it here as well. As in the case of MAXCUT, it can be obtained by using the mapping between x_i and z_i defined by $x_i = \frac{1}{2}(1 + z_i)$ such that it reads

$$\min_{\mathbf{z}} C_{\text{MIS}}(\mathbf{z}) = \min_{\mathbf{z}} \left(-\frac{1}{N} - \sum_{i=1}^N z_i + \frac{\alpha}{4} \sum_{ij \in E} (1 + 2z_i + 2z_j + 4z_i z_j) \right). \quad (2.8)$$

Classical greedy algorithms

Since a lot of effort has been put into finding efficient ways to solve the problem, polynomial-time classical algorithms for approximating the optimal solution with good performance have been developed [39–42]. A simple but very common classical greedy approach is the so-called MIN algorithm [16]. It iteratively chooses a vertex in the problem graph with the lowest degree, adds this vertex to the independent set and removes all nodes that are adjacent to the chosen vertex until the graph has no vertices any more. If multiple nodes have the same degree, one of them is chosen randomly. An example application of MIN is given in Fig. 2.3. Literature also provides a lower performance bound of the algorithm for a graph with maximum vertex degree δ : For any graph, MIN reaches at least an approximation ratio of $\frac{3}{\delta+2}$ with respect to the optimal maximum independent set size [43].

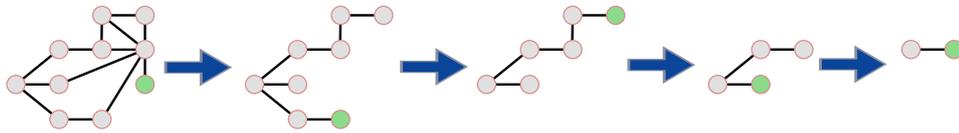


Figure 2.3: An example of the MIN algorithm applied to a MIS problem. The green vertex in each step indicates that it is included in the solution independent set and all neighbours are removed.

An alternative approach is the MAX algorithm. It instead iteratively chooses a vertex with maximal degree and removes it from the graph until there exist no edges any more. The chosen solution independent set corresponds to all remaining vertices. An example of MAX is shown in Fig. 2.4.

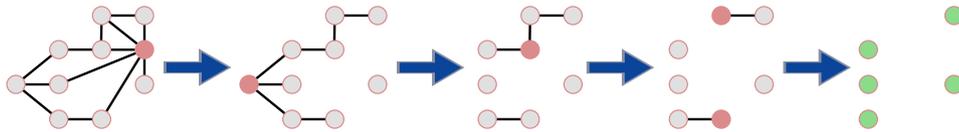


Figure 2.4: An example of the MAX algorithm applied to a MIS problem. The red vertex in each step indicates that it is excluded from the graph. The solution independent set is coloured in green representing the nodes that are left in the graph.

It is explicitly mentioned that there are more elaborate heuristic algorithms to solve MIS problems but especially MIN is widely used as classical benchmark. Furthermore, as can be seen later, there are interesting links between the introduced heuristics and common quantum approaches.

2.1.4 Set cover problem

A more complicated problem than MAXCUT and MIS is the SETCOVER problem. In this section I introduce the underlying question of the problem and that its QUBO formulation requires different means of encoding than MAXCUT and MIS. In addition, a simple classical solver and a use case example of the SETCOVER problem in industry are described in detail.

Definition and QUBO formulation

The set up of the SETCOVER problem consists of a set $U = \{1, \dots, n\}$ and subsets $V_i \subseteq U$ with $i \in \{1, \dots, N\}$ such that

$$U = \bigcup_{i=1}^N V_i. \quad (2.9)$$

The question is now to find the smallest possible number of sets V_i that still cover all elements of U . In general, finding such a set of sets is NP-hard [29]. Fig. 2.5 shows a SETCOVER problem, a non-minimum solution and a minimum solution.

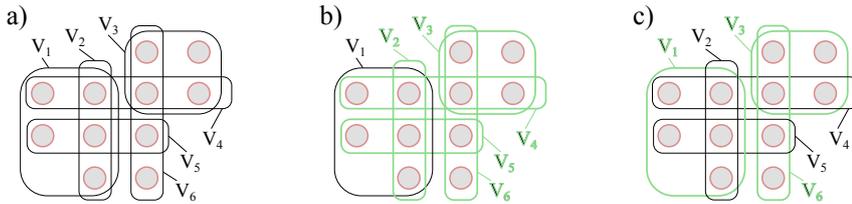


Figure 2.5: In a) an example of the SETCOVER problem algorithm is illustrated. b) shows a non-minimum solution to the problem and c) a minimum solution.

For finding a QUBO formulation of the problem, I follow Ref. [44] and define the binary variables $x_i \in \{0, 1\}$ with $i \in \{1, \dots, N\}$ which indicate whether subset V_i is included or not in the solution set of subsets. Furthermore, it is required to define an integer M as the maximum number a single element α of U is included in all subsets $\{V_i\}$. With this I further introduce variables $x_{\alpha,m} \in \{0, 1\}$ with $m \in \{1, \dots, M\}$ that are equal to 1 if the number of subsets V_i that include element α corresponds to m . Otherwise $x_{\alpha,m}$ is 0. Using these variables together with two constants A and B

allows us now to define the QUBO cost function of the SETCOVER problem as

$$\begin{aligned}
 C_{\text{SETCOVER}}(\mathbf{x}) = & A \sum_{\alpha=1}^n \left(1 - \sum_{m=1}^M x_{\alpha,m} \right)^2 \\
 & + A \sum_{\alpha=1}^n \left(\sum_{m=1}^M m x_{\alpha,m} - \sum_{i:\alpha \in V_i} x_i \right)^2 + B \sum_{i=1}^N x_i.
 \end{aligned} \tag{2.10}$$

This cost function has to be minimized over all x_i and $x_{\alpha,m}$ in order to find the minimum solution to the SETCOVER problem. The reasoning behind this QUBO formulation is as follows: The first term makes sure that only a single $x_{\alpha,m}$ equals 1 for each element α . This must apply because every element has to be included a fixed number of times. The second term ensures that the number of chosen subsets V_i that include element α is actually equal to m . Minimizing the third summand encodes the actual question of finding the minimum number of subsets. To make sure that the \mathbf{x} which minimizes the cost function fulfills these constraints, it is required that $A > B > 0$. Instead of using M it is also possible to define an integer M_α for each individual element α of U as the number this element is included in all subsets $\{V_i\}$. This allows to minimize over less variables.

The Ising formulation of the problem with variables $z_i \in \{1, -1\}$ is rather lengthy but since I use it in later Chapters, the formulation can be found in Appendix A.

Greedy MAX solver

A simple but already well performing classical greedy algorithm for solving the SETCOVER problem consists of iteratively choosing the subset V_i with the highest number of elements, including it in the solution and removing all elements in V_i from the other subsets V_j and from U until U is empty. In case the cardinality of multiple subsets equals each other, the algorithm chooses one randomly. I will call this algorithm MAX throughout the remainder of the thesis. It is always clear from the context whether MAX refers to the algorithm solving the MIS or the SETCOVER problem. An example of applying MAX to a given problem is shown in Fig. 2.6.

Sensor positioning use case

A use case of the SETCOVER problem is sensor positioning in an industry production plant where autonomous vehicles are used. The vehicles are moving in a dedicated space between different locations. To ensure that they do not hit any obstacles or even humans on their path, it is of utter importance to surveil this area. In practice, this is done by sensors. Generally these sensors are very expensive. Therefore, it is

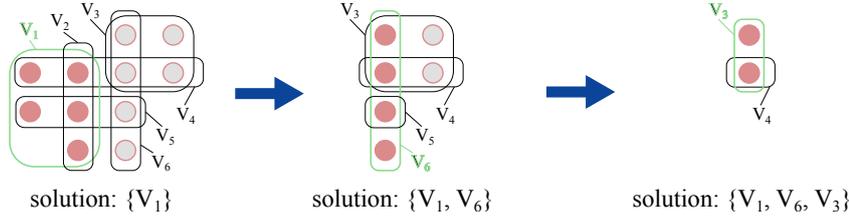


Figure 2.6: The application of the MAX algorithm to a SETCOVER problem is shown. The green boxes indicating the chosen subset V_i in each step and the red nodes the elements of V_i that are also removed from the set U .

advantageous not to install them in each vehicle but to rather place them externally at the facilities of the plant.

When doing so, all spots in the surveilled area need to be covered by the sensors' range. At the same time, to save further costs using as few sensors as possible is beneficial. Finding these is actually equivalent to solving a SETCOVER problem. Discretizing the monitored area allows representing it as a set U that needs to be fully covered by sensors. Therefore, the elements of U represent 3-dimensional cubes. The sensors and their covered area can be imitated by subsets V_i . Their elements consist of the discretized areas that a specific sensor installed at a certain spot can cover. Thus, solving this SETCOVER problem allows finding the minimum number of sensors that are required to monitor the whole area where the autonomous vehicles are moving.

2.2 Tensor networks

In this section I will lay out the basics of tensors, tensor networks and operations that are applied thereon. Furthermore, different types of tensor networks are presented and how networks can be used to simulate quantum circuits.

2.2.1 Basic concepts of tensor networks

Tensor networks provide a powerful framework for representing and manipulating high-dimensional data structures, which are essential in various fields, including simulations of quantum computing. A tensor is a multidimensional array. A tensor network is a collection of tensors connected by edges, representing contractions between tensors. Each edge stands for a summation over a shared index, allowing efficient representation and computation of complex operations.

Mathematically, a tensor T of rank n is an element of the tensor product of n

vector spaces:

$$T \in \bigotimes_{i=1}^n V_i, \quad (2.11)$$

where V_i are vector spaces. For example, a rank-2 tensor is a matrix, a rank-3 tensor can be thought of as a cube of numbers, and so on [45].

2.2.2 Tensor operations

In tensor networks, several operations are fundamental for constructing and manipulating the networks. The main tensor operations include the following:

Tensor contraction

Tensor contraction is the operation of summing over the common indices of two or more tensors. It generalizes matrix multiplication to higher dimensions. For two tensors T^{ij} and U_{jk} , the contraction over the index j is given by:

$$(T \cdot U)^{ik} = \sum_j T^{ij} U_{jk}. \quad (2.12)$$

This operation is crucial for calculating physical quantities and simulating quantum circuits [46].

Tensor decomposition

Tensor decomposition involves breaking down a tensor into a product of simpler tensors. Common decompositions include the Singular Value Decomposition (SVD) and the Tucker decomposition. For a matrix T , the SVD is given by:

$$T = U \Sigma V^\dagger, \quad (2.13)$$

where U and V are unitary matrices, and Σ is a diagonal matrix of singular values. Decomposition techniques are used for truncating tensors to reduce computational complexity while preserving essential features [47].

Tensor index permutation

Permuting the indices of a tensor reorders its dimensions without changing its values. For a tensor T^{ijk} , a permutation might result in T^{jik} . This operation is often required to align tensors correctly before performing contractions or other operations [45].

Tensor reshaping

Reshaping changes the dimensions of a tensor while keeping the total number of elements constant. For example, a tensor T with dimensions $(2, 3, 4)$ can be reshaped into dimensions $(6, 4)$. This operation is useful in preparing tensors for specific algorithms or decompositions [46].

2.2.3 Types of tensor networks

Several types of tensor networks are commonly used in simulating quantum computing and other fields, each with its own structure and advantages. Although most types are not used for the later simulations of quantum circuits, the main types are presented here for the purpose of completeness and giving a thorough overview:

Matrix product states (MPS)

Matrix product states are one-dimensional tensor networks that represent quantum states of spin chains or one-dimensional systems. A MPS for a state $|\psi\rangle$ can be expressed as:

$$|\psi\rangle = \sum_{i_1, i_2, \dots, i_N} \text{Tr}(A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_N}^{[N]}) |i_1, i_2, \dots, i_N\rangle, \quad (2.14)$$

where each $A_{i_k}^{[k]}$ is a complex square matrix of order χ . The latter is the bond dimension that determines the amount of entanglement that can be represented [47]. The indices i_j go over the states in the computational basis, i.e., for qubits $i_j \in \{0, 1\}$.

Projected entangled pair states (PEPS)

Projected Entangled Pair States (PEPS) generalize MPS to higher dimensions, making them suitable for representing quantum states on lattices. A PEPS on a two-dimensional lattice can be written as:

$$|\psi\rangle = \sum_{\{i\}} \text{Tr}(A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_N}^{[N]}) |i_1, i_2, \dots, i_N\rangle, \quad (2.15)$$

where $A_{i_k}^{[k]}$ are tensors that connect with neighboring tensors on the lattice, forming a network that can capture more complex entanglement structures than MPS [48].

Tree tensor networks (TTN)

Tree tensor networks (TTN) represent quantum states using a hierarchical structure. They are particularly useful for capturing the entanglement in systems where the

correlation length is large. A TTN for a state $|\psi\rangle$ is composed of tensors arranged in a tree-like structure, where the leaves represent physical indices and the internal nodes represent virtual indices:

$$|\psi\rangle = \sum_{\{i\}} T_{i_1 i_2}^{[1]} T_{i_3 i_4}^{[2]} \cdots T_{i_{2n-1} i_{2n}}^{[n]} |i_1, i_2, \dots, i_N\rangle, \quad (2.16)$$

where $T^{[k]}$ are the tensors at each node of the tree [49].

Multiscale entanglement renormalization ansatz (MERA)

Multiscale entanglement renormalization ansatz (MERA) is a tensor network that efficiently represents quantum states with scale-invariant structures. It is particularly suited for critical systems and those with logarithmic corrections to the area law of entanglement entropy. The MERA network includes disentanglers and isometries that act on different scales of the system, allowing for an efficient representation of states with long-range entanglement:

$$|\psi\rangle = \sum_{\{i\}} \mathcal{W}_{i_1 i_2}^{[1]} \mathcal{W}_{i_3 i_4}^{[2]} \cdots \mathcal{W}_{i_{2n-1} i_{2n}}^{[n]} |i_1, i_2, \dots, i_N\rangle, \quad (2.17)$$

where $\mathcal{W}^{[k]}$ are tensors that include both isometries and disentanglers [50].

2.2.4 Tensor networks for simulating quantum circuits

Tensor networks are particularly well-suited for simulating quantum circuits due to their ability to efficiently represent the exponential complexity of quantum states. In quantum circuit simulation, tensor networks are used to represent both the state of the quantum system and the operations (gates) applied to it.

A quantum circuit can be represented as a sequence of unitary operations U_i applied to an initial state $|\psi_0\rangle$:

$$|\psi_f\rangle = U_L \cdots U_2 U_1 |\psi_0\rangle, \quad (2.18)$$

where each U_i can be decomposed into tensors representing the action of quantum gates on the state-vectors. For instance, a two-qubit gate U_{ij} acting on qubits i and j can be represented as a rank-4 tensor $U_{i'j'}^{ij}$, with indices i, j representing the input qubits and i', j' representing the output qubits [51].

The simulation involves contracting these tensors to compute the final state of the system. The contraction process, which involves summing over shared indices, can be performed efficiently using various algorithms, depending on the network structure.

For example, in the case of MPS, the computational cost scales linearly with the number of qubits N and polynomially with the bond dimension χ [47].

Consider a simple example of a quantum circuit with two qubits and two gates, U_1 and U_2 . The initial state can be written as $|\psi_0\rangle = \sum_{i_1, i_2} \alpha_{i_1 i_2} |i_1, i_2\rangle$. After applying the first gate U_1 and then U_2 , the final state is:

$$|\psi_f\rangle = \sum_{i_1, i_2, j_1, j_2} U_{2, k_1 k_2}^{j_1 j_2} U_{1, j_1 j_2}^{i_1 i_2} \alpha_{i_1 i_2} |k_1, k_2\rangle, \quad (2.19)$$

where the tensors U_1 and U_2 are contracted with the initial state tensor α to yield the final state tensor.

By leveraging tensor networks, quantum circuit simulators can handle larger systems than traditional methods by exploiting the entanglement structure and sparsity inherent in many quantum states. In comparison to state-vector simulations, tensor network simulations do not require saving the whole state-vector that requires extensive memory. Applying a gate consists of a summation of a few indices instead of a matrix transformations of the full state with the applied gate. This again saves memory. Tensor network simulations have been important in advancing the understanding of quantum algorithms and their practical implementations [51–53].

2.3 Quantum optimization algorithms (QOAs)

In this section I introduce different quantum optimization algorithms (QOAs) that have been developed in recent years to tackle the problem of solving combinatorial optimization problems. Here, I will focus on the quantum approximate optimization algorithm (QAOA), the recursive QAOA (RQAOA) and different variations of the quantum-informed recursive optimization algorithm (QIRO) for solving MAXCUT, MIS and SETCOVER problems.

2.3.1 Quantum approximate optimization algorithm (QAOA)

Farhi *et al.* [4] introduced the quantum approximate optimization algorithm (QAOA) as promising alternative to classical approaches for finding good approximate solutions to unconstrained binary optimization problems. Since then the algorithm received a lot of interest because of its universality and adaptive complexity. Furthermore, it can already now be deployed on noisy intermediate scale quantum (NISQ) devices. In recent years the QAOA was successfully applied to MAXCUT [4], MIS [54, 55] and other QUBO problems such as binary paint shop [56], binary linear least squares [57] or multi-knapsack problems [58]. From an application point of view, researchers used the algorithm to solve problems in portfolio optimization [59,

60], object detection [61], text summarization [62], protein folding [63] and wireless scheduling [64] to name a few examples.

Technically speaking, QAOA's goal is to find a $\mathbf{z} \in \{-1, 1\}^N$ such that the cost function $C : \{-1, 1\}^N \rightarrow \mathbb{R}$ of a binary optimization problem is minimized. To achieve this, the algorithm starts from the uniform superposition state $|+\rangle$ of N qubits defined by

$$|+\rangle = \frac{1}{\sqrt{2^N}} \sum_{\mathbf{z} \in \{-1, 1\}^N} |\mathbf{z}\rangle. \quad (2.20)$$

In the next step, a sequence of parametrized layers $U_C(\gamma_i)$ and $U_M(\beta_i)$ is applied to the initial state. Here, γ_i and β_i represent real-valued parameters. These layers are defined by

$$U_C(\gamma_i) = e^{-i\gamma_i H_C} \quad (2.21)$$

and

$$U_M(\beta_i) = e^{-i\beta_i H_M}. \quad (2.22)$$

H_C denotes the cost Hamiltonian that encodes the cost function C of a problem as

$$H_C |\mathbf{z}\rangle = C(\mathbf{z}) |\mathbf{z}\rangle \quad \forall \mathbf{z} \in \{-1, 1\}^N, \quad (2.23)$$

and H_M is the so-called mixer Hamiltonian

$$H_M = \sum_{i=1}^N X_i, \quad (2.24)$$

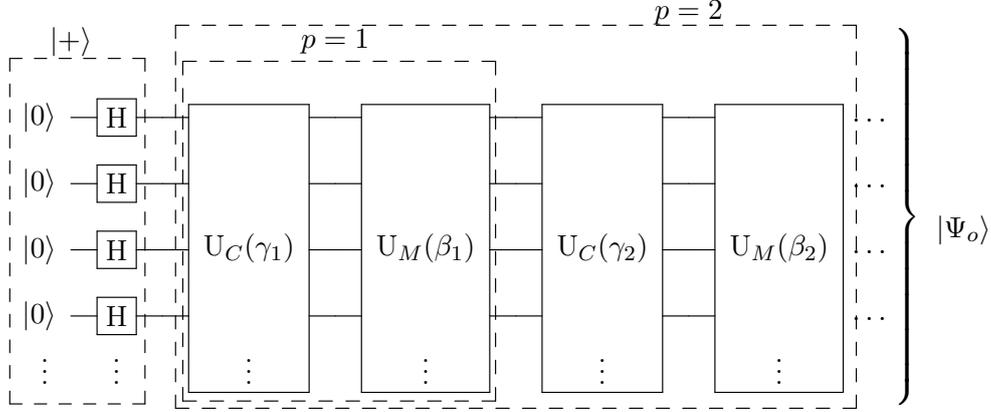
where X_i is the Pauli-X operator applied on qubit i .

After applying the layers, the output state corresponds to

$$|\Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} |+\rangle, \quad (2.25)$$

where $p \in \mathbb{N}$ is the so-called depth of the QAOA. The circuit representation of the

algorithm is as follows:



In the next step of the algorithm the expectation value $F(\boldsymbol{\beta}, \boldsymbol{\gamma})$ of the cost Hamiltonian with respect to the prepared state

$$F(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) | H_C | \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle \quad (2.26)$$

is obtained by repeated measurements of the prepared state.

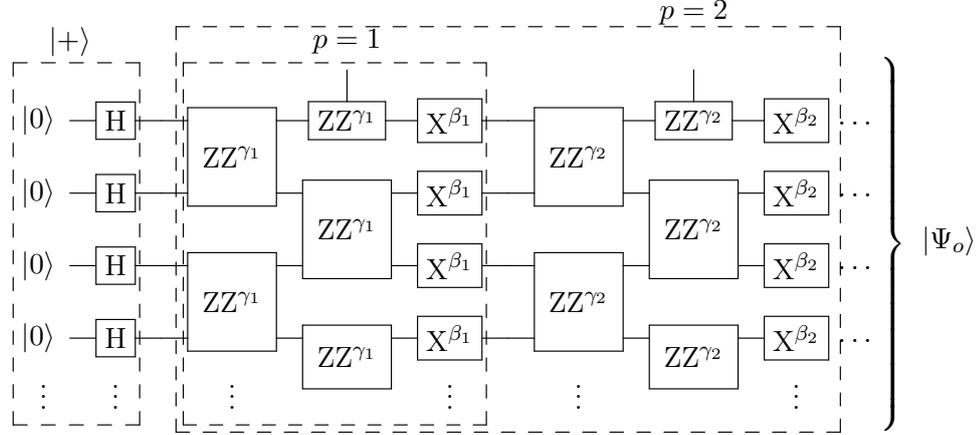
The process of obtaining the expectation value, together with a classical subroutine are used iteratively to minimize the expectation value $F(\boldsymbol{\beta}, \boldsymbol{\gamma})$ with respect to the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. In case of an optimal minimization, the ground state of the cost Hamiltonian H_C is obtained. This ground state corresponds to the optimal solution of the original COP.

For the QUBO problems defined in 2.1, the corresponding cost Hamiltonians H_C coincide with their Ising formulations with the variables z_i replaced by the corresponding Pauli operator Z_i and terms without variable supplemented by the identity operator I . For example, in case of MAXCUT the cost Hamiltonian becomes

$$H_{C_{\text{MAXCUT}}} = \frac{1}{2} \sum_{ij \in E} w_{ij} (Z_i Z_j - I) \quad (2.27)$$

such that the QAOA circuit for an example with only nearest neighbour connections

explicitly looks like



The cost Hamiltonians and therefore QAOA circuits of the MIS and SETCOVER problems have a more complicated structure than the MAXCUT problem but follow analogously.

A powerful tool for simulating $p = 1$ QAOA is the fact, that the expectation values $F(\boldsymbol{\beta}, \boldsymbol{\gamma})$ can be calculated analytically as shown in Ref. [65]. These expectation values are a sum of individual expectation values of the form

$$\langle \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) | J_{ij} Z_i Z_j | \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle \quad (2.28)$$

or

$$\langle \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) | J_i Z_i | \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle. \quad (2.29)$$

Each of these terms can be expressed analytically as functions of the QAOA parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$. This allows an efficient and very fast calculation of $p = 1$ QAOA results. Throughout the thesis, these expressions are used to evaluate $p = 1$ circuits.

Although researchers managed to gain good results using QAOA, it remains unclear whether the algorithm is able to outperform classical methods in solving COPs. Various analyses were able to find certain problems of using QAOA in practice such as barren plateaus [5], noise of quantum devices [66] or locality. Locality refers to the fact that at constant circuit depth p only qubits can interact with each other that are separated less than a certain distance in the circuit. Thereby the performance of QAOA is provably limited [7–10]. To overcome the issue of locality a recursive variant of the QAOA can be used as will be explained in the next section.

2.3.2 Recursive quantum approximate optimization algorithm (RQAOA)

In order to overcome the known problem of locality of QAOA, Bravyi *et al.* [7, 11] proposed the recursive QAOA (RQAOA). The underlying idea of the algorithm consists of iteratively using QAOA to eliminate a variable of the present problem to shrink the problem to smaller sizes. The shrinking step allows non-local interactions between qubits.

In this section I introduce the original form of the RQAOA and a variation thereof with so-called recalculation intervals.

The original RQAOA

More technically speaking, the original RQAOA finds an approximation to the ground state of a cost Hamiltonian H_C of a given quadratic unconstrained Ising problem using QAOA. Without loss of generality the Hamiltonian has the form

$$H_C = \sum_{ij \in E} w_{ij} Z_i Z_j. \quad (2.30)$$

After the QAOA minimization, for every $ij \in E$ the expectation values

$$b_{ij} = \langle Z_i Z_j \rangle \equiv \langle \Psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}}) | Z_i Z_j | \Psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}}) \rangle \quad (2.31)$$

are calculated. Here, $\boldsymbol{\beta}_{\text{opt}}$ and $\boldsymbol{\gamma}_{\text{opt}}$ stand for the optimized parameters after QAOA minimization. The values b_{ij} are also referred to as correlations.

In a next step, the algorithm seeks the correlation b_{ij} with the highest absolute value. Depending on the sign σ_{ij} of b_{ij} the two variables z_i and z_j are said to be correlated ($b_{ij} > 0$) or anti-correlated ($b_{ij} < 0$). The variable z_i is now eliminated by imposing the constraint

$$Z_i = \text{sgn}(b_{ij}) Z_j \quad (2.32)$$

in the cost Hamiltonian. This step effectively decreases the number of variables of the problem by one and possibly introduces new interaction terms in the Hamiltonian between variables that were not connected before. It actually is equivalent to removing variable i and updating the weights of all neighbouring variables $\{k \in V \mid ik \in E\}$ of node i in the following way:

$$w'_{jk} = \begin{cases} \omega_{jk} + \sigma_{ij} \omega_{ik}, & \text{if } jk \in E \\ \sigma_{ij} \omega_{ik}, & \text{if } jk \notin E. \end{cases} \quad (2.33)$$

The full RQAOA consists of repeating these steps of applying QAOA and shrinking the problem iteratively, whereas the problem size decreases in every step. This is

done until a certain threshold size of the remaining problem is reached. The smaller problem can then be solved classically by for example a brute force approach. In a last step the solution to the original problem has to be obtained by backtracking the individual shrinking steps of the RQAOA in order to find the final value of every variable z_i .

RQAOA with recalculation intervals

An adaptation of the RQAOA consists of introducing recalculation intervals as we have introduced them in Ref. [23]. When using a recalculation interval r , the usual RQAOA procedure of optimizing a QAOA and finding the correlations b_{ij} is applied. But now instead of doing only one shrinking step based on the correlations as defined by Eq. (2.32), r reduction steps are done.

Technically this means choosing not only the correlation with the highest absolute value but the r correlations with the r highest absolute values in descending order. These are then used to shrink r times. After multiple shrinking steps, it may happen that a node u associated with the correlation b_{su} no longer exists because it has been merged with another node v . This scenario is illustrated in Fig. 2.7 taken from our recent publication [23].

Initially, the correlations b_{uv}, b_{su}, b_{ut} , and b_{tv} are computed for the input graph on the left. These correlations are then sorted in descending order of their absolute values. The graph on the left side is shrunk based on the highest correlation b_{uv} , resulting in the output graph on the right, where node u has been combined with node v .

The next highest correlation in the sequence is b_{su} . Since node u is no longer present in the current graph, we reinterpret b_{su} as b_{sv} . To ensure consistency, the sign adjustment from the previous merging step is applied, setting $b_{sv} = \sigma_{uv} \cdot b_{su}$.

Additionally, similar rules are applied if nodes i and j have already been merged with other nodes. If the nodes i and j to be merged are already part of the same node k , this merging step is skipped, and the algorithm proceeds to the next highest correlation. This skip does not count as a step in the process.

2.3.3 Quantum-informed recursive optimization algorithm (QIRO) for MIS problems

Based on the RQAOA Finžgar *et al.* [12] introduced very recently the more general class of so-called quantum-informed recursive optimization (QIRO) algorithms. The basic idea of the algorithms is similar to RQAOA but here, also different correlations are calculated and especially problem-specific update steps are used for reducing the problem size recursively [12]. Furthermore, they introduced promising extensions to the algorithms, such as backtracking. In this section I will focus on different instances

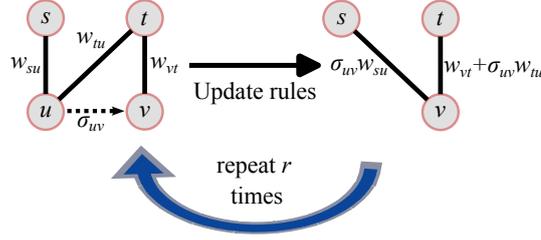


Figure 2.7: The illustration shows the shrinking step of RQAOA that can be repeated r times when using recalculation intervals in accordance with Eq. (2.32). The Figure is taken from our own work [23].

of these QIRO algorithms for the specific case of MIS problems. In more detail, the so-called MAXQ, MINQ, MMQ and 2-MMQ routines are discussed as introduced in Refs. [16] and [12], respectively.

MAXQ algorithm

The MAXQ algorithm is inspired by the classical MAX greedy routine for solving the MIS problem as explained in Sec. 2.1.3 [16]. For a given problem instance, MAXQ uses QAOA to find a good approximation of the ground state of the problem Hamiltonian. In the next step, all one-point correlations

$$b_i = \langle Z_i \rangle \equiv \langle \Psi(\beta_{\text{opt}}, \gamma_{\text{opt}}) | Z_i | \Psi(\beta_{\text{opt}}, \gamma_{\text{opt}}) \rangle \quad (2.34)$$

of the problem are calculated. Then, the correlation b_i with the lowest value is chosen and the corresponding variable i is fixed to not lie in the independent set. In the graph representation of the problem, the corresponding node is removed. This update rule is also shown in a more illustrative way in part (b) of Fig. 2.8 taken from Ref. [12].

These steps of QAOA, correlation calculation and shrinking are now recursively repeated until the problem graph is totally non-connected. The remaining nodes are chosen to be the solution to the problem.

MAXQ is inspired by MAX because with $p = 1$ QAOA, it chooses the same shrinking steps as the greedy MAX algorithm [16]. This means, the single variable expectation values b_i of $p = 1$ QAOA hold only information about the degree of the corresponding node. The correlation with the lowest energy corresponds to the node with the highest degree in the problem graph.

MINQ algorithm

Analogously the MINQ routine is understood as the quantum equivalent to the MIN algorithm [16]. It works the same way as MAXQ but instead of deleting the variable i with the lowest correlation b_i from the problem graph, the variable with the highest value is chosen to be in the solution independent set. The corresponding vertex in the graph and all nodes that are connected to the chosen vertex are removed and fixed to not lie in the independent set. An example case can be found in illustration (a) of Fig. 2.8 [12].

Similar to MAXQ, when using $p = 1$ QAOA for finding a ground state approximation and correlations, MINQ uses only information about the degrees of the graph vertices. The vertex with the highest value corresponds to the one with the lowest degree [16]. Therefore, in case of $p = 1$ MINQ represents the greedy MIN if the true ground state of the problem Hamiltonian is found by QAOA.

MMQ algorithm

Combining MINQ and MAXQ leads to the MMQ algorithm [16]. Here, QAOA is again used to find a good low energy state of the problem Hamiltonian before all single variable correlations b_i are calculated. Then the variable with the highest absolute correlation value $|b_i|$ is picked. Depending on the sign of the correlation, the MAXQ or MINQ shrinking rule is then applied: In case b_i is negative, the node i is removed from the graph and not included into the solution independent set. If $b_i > 0$, variable i is included into the solution independent set, its corresponding vertex and all neighbouring variables are removed from the problem. These update rules correspond to (a) and (b) in Fig. 2.8 [12].

These steps are repeated until only non-connected vertices are left in the problem graph. These are also added to the solution independent set.

2-MMQ algorithm

An extension to the MMQ routine can be made by also taking correlations between two variables into account as it has been done in the work of Finžgar *et al.* [12]. Throughout the thesis, I will call this extension 2-MMQ algorithm.

After using QAOA, 2-MMQ not only calculates one-point correlations b_i but also two-point correlations

$$b_{ij} = \langle Z_i Z_j \rangle \equiv \langle \Psi(\beta_{\text{opt}}, \gamma_{\text{opt}}) | Z_i Z_j | \Psi(\beta_{\text{opt}}, \gamma_{\text{opt}}) \rangle, \quad (2.35)$$

analogously to RQAOA. It then picks the correlation with the highest absolute value of all correlations $\{b_1, \dots, b_N, b_{1,2}, \dots, b_{N,N-1}\}$. Depending on the sign and whether it

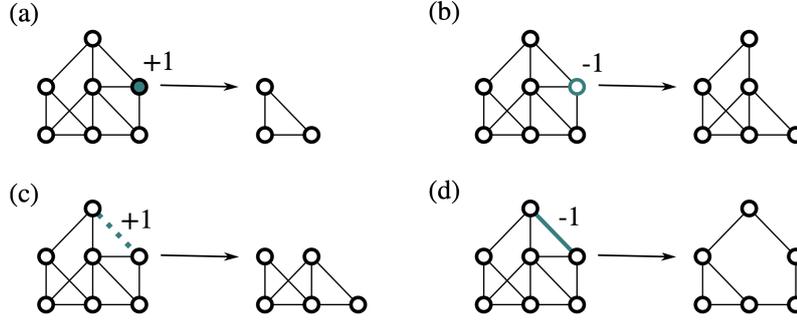


Figure 2.8: Update rules of the different shrinking algorithms for MIS problems. In case of shrinking based on a positive one-point correlation (a) applies. I is used in MINQ, MMQ and 2-MMQ. (b) showcases the example of a negative single variable correlation as it is used in MAXQ, MMQ and 2-MMQ. For positive and negative two-point correlations, the rules in (c) and (d) are used as it is the case in the 2-MMQ algorithm. The figure is taken from Ref. [12].

is a single or two variable correlation, different shrinking rules are applied in a next step [12]:

- In case a positive single variable correlation $b_i > 0$ is chosen, the MINQ update rule is used. Thus, the picked variable is fixed to lie in the independent set, its vertex and all neighbouring ones are removed from the problem.
- If the correlation with the highest absolute value is negative and of a single vertex $b_i < 0$, 2-MMQ makes use of the MAXQ shrinking step, i.e., the chosen vertex is removed and fixed to not lie in the independent set.
- When a positive two body term $b_{ij} > 0$ corresponds to the correlation with the highest absolute value, both variables i and j are removed from the problem and determined not to lie in the independent set. The reasoning behind is that this correlation indicates that both variables i and j have the same affiliation with respect to the independent set. Therefore, since the correlations are neighbours they have to not be part of the independent set. Otherwise just this constraint of the MIS problem is violated.
- In case of a negative two variable correlation $b_{ij} < 0$, 2-MMQ fixes all variables that are neighbours of both nodes i and j not to be in the independent set. They are also removed from the problem. The intuition behind is that the correlation $b_{ij} < 0$ indicates i and j being in different partitions of the graph, thus one of the variables is in the independent set and the other not. Therefore,

to not violate the MIS constraint, all vertices connected to both variables i and j are not part of the solution independent set.

These update rules are also summarized in Fig. 2.8 which is taken from Ref. [12]. As it is the case for the other shrinking algorithms as well, the steps of QAOA, correlations calculation and shrinking are repeated until there are no variables left or only non-connected ones. In the later case these variables are fixed to be part of the solution independent set.

2.3.4 Quantum-informed recursive optimization algorithm (QIRO) for SetCover problems

The QIRO framework can also be used to solve the SETCOVER problem as it is introduced in Sec. 2.1.4. For example an adaptation of the just introduced MINQ algorithm allows recursive shrinking of the problem with a guaranteed valid solution to the SETCOVER problem.

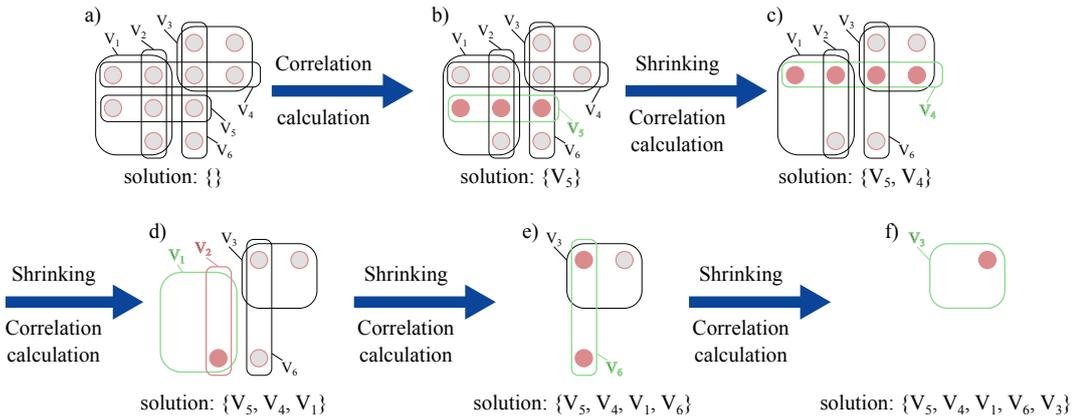


Figure 2.9: An example of the MINQ shrinking process on a SETCOVER problem. The pictured case is not an optimal solution to the SETCOVER problem. In each iteration the green subset mimics the one with the highest correlation. Thus, this subset is included in the solution, respectively. In step d) of the procedure, subset V_1 is added to the solution such that the subset V_2 indicated by red color has no elements any more. Therefore, subset V_2 is removed and not added to the solution.

In this case I first use QAOA to receive a good solution to the problem cost function from equation (A.1). In the next step only the single qubit correlations b_i of variables z_i are calculated. Each z_i represents a possible subset V_i being included in the solution or not. The correlations of the remaining variables $z_{\alpha,m}$ are not used within the shrinking procedure.

In the next step, shrinking is applied on the variable z_i with the highest correlation value b_i . The reduction step here consists of removing variable z_i from the problem and fixing the corresponding subset V_i to lie in the solution. Furthermore, all elements of V_i are removed from the set U and subsets V_j . Thus, it can occur that some subsets V_k are empty afterwards. These and the corresponding variables z_k are then removed as well and fixed not to be included in the solution.

These steps are applied in an iterative manner to the shrunk SETCOVER problem until set U is empty or the problem small enough to be solved by a classical routine. Since the QUBO formulation includes many ancilla qubits, the graphical representation of the shrinking routine in this problem is not intuitive and without natural encoding. Nevertheless, an example of a shrinking behaviour is shown in Fig. 2.9. In this case it is not an optimal solution to the problem. In panel d) of the Figure, red-colored subset V_2 and thus the corresponding variable z_2 is removed without being included into the solution because all of its elements have been removed as a result of other shrinking steps.

It is explicitly mentioned at this point that the name MINQ is chosen for this algorithm because of the similarity to the MINQ routine for the MIS problem. In the case of SETCOVER, the name has no deeper meaning. Therefore, the reader may not be confused when considering the greedy MAX algorithm for SETCOVER problems from Sec. 2.1.4 as classical benchmark for the MINQ.

Chapter 3

Analysis of tensor network simulations of QAOA

Throughout this chapter I will introduce how in this thesis tensor networks are used to encode and simulate QAOA circuits. Furthermore, I explain different types of parameter γ , β initialization of QAOA. The importance of choosing a suitable initialization method is shown in benchmarks of the different methods by numerical experiments on MAXCUT and MIS problems. In addition, within these experiments a thorough search is conducted on which optimizers and hyperparameters are suitable for finding good ground state approximations of the given problems using QAOA.

3.1 Simulation of higher depths QAOAs using tensor networks

In this thesis the QAOA circuits of higher depths are simulated using tensor networks similar to the explanation given in Sec. 2.2.4, however, with some adaptations. These are based on prior research, especially the `Qtensor` library introduced in Refs. [67–69]. First, the circuit needs to be mapped to a tensor network. This network is then contracted in order to simulate the circuit. I provide the details of these two steps in this section.

3.1.1 Encoding of QAOA as tensor network

The method to encode QAOAs as tensor networks used in this work requires a few process steps that are explained in the following. Illustratively, these steps applied to a $p = 1$ QAOA circuit of a simple MAXCUT problem are shown in Fig. 3.1. Although $p = 1$ circuits are not carried out with tensor networks in this thesis, the example can be easily expanded to demonstrate how higher depth circuits are simulated. In general, as already explained in Sec. 2.3.1, optimizing a QAOA for the present problems does not require calculating the full output state of the circuit but

only obtaining expectation values of the form

$$\langle \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) | J_{ij} Z_i Z_j | \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle \quad (3.1)$$

or

$$\langle \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) | J_i Z_i | \Psi_o(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle. \quad (3.2)$$

Furthermore, the quantum optimization algorithms investigated in this thesis are based on correlations obtained from QAOA, i.e., they also require only expectation values of the aforementioned form. This is also shown in part a) of the Figure. Therefore, all quantum circuits that need to be evaluated have the structure as the one in illustration b) of Fig. 3.1.

Before transferring such circuits to tensor networks, the so-called lightcone simplification as shown in part c) of the Figure is executed. It consists of removing all gates and qubits that do not influence the expectation value to be evaluated. This occurs because many gates are directly applied after their inverse leading to an output as if no gate was applied at all.

Then, the remaining QAOA circuit to be evaluated is mapped to a tensor network. This is done as explained in Sec. 2.2.4 by interpreting a quantum state of n qubits as tensor from $(\mathbb{C}^2)^{\otimes n}$ and a quantum gate as a tensor with input and output indices for each qubit it acts on [70]. An input index corresponds to the output index of the previous gate as outlined in part d) of Fig. 3.1.

3.1.2 Quantum circuit simulation by tensor network contraction

In this thesis, the exact simulation of a quantum circuit is achieved by contracting the corresponding tensor network by so-called bucket elimination [71]. In this procedure the different indices, thus edges, are contracted sequentially. Every step consists of selecting an index i of the network and summing over the product of all tensors containing this index. The computational complexity of this contraction is highly sensitive to the sequence in which the tensor indices are contracted. The complexity actually scales exponentially with the maximal rank of all intermediate tensors throughout the contraction [67].

Therefore, optimizing the contraction sequence is essential to reduce the required computational effort [72]. In this study, I employ a technique for determining an efficient contraction order utilizing the line graph representation of the tensor network, following Ref. [70]. In a line graph of a tensor network every vertex represents an edge (index) of the original network. An edge between two vertices of a line graph is present if the two (represented) corresponding edges of the tensor network share a same node in the original network. This might sound confusing in the beginning but it is helpful to see an example for overcoming the confusion. To illustrate such a graph, part a) of Fig. 3.2 shows the line graph of the tensor network of Fig. 3.1.

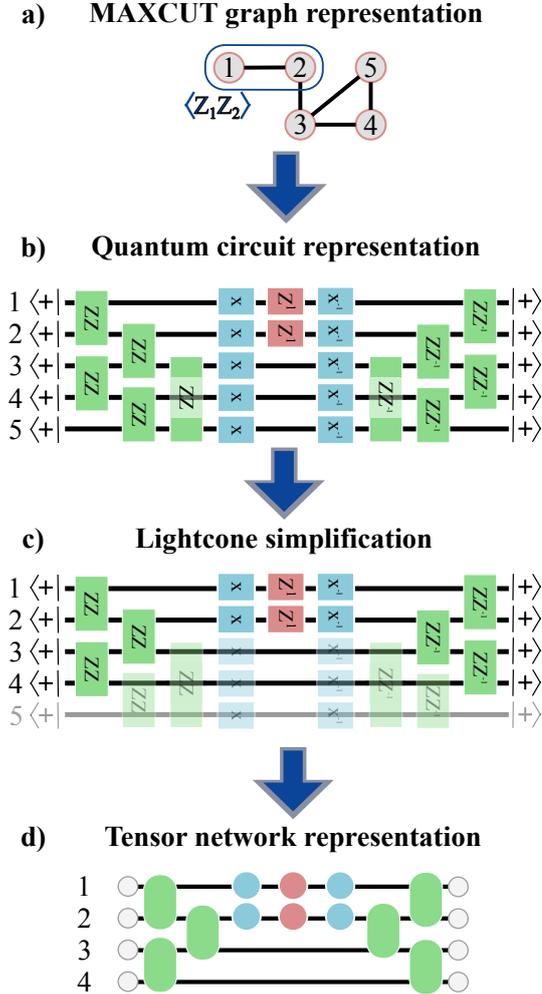


Figure 3.1: Part a) shows a simple MAXCUT graph. The $p = 1$ QAOA circuit to obtain the expectation value $\langle Z_1 Z_2 \rangle$ of the problem is pictured in b). The lightcone simplification shrinks the quantum circuit in c). Illustration d) displays the tensor network representation of the circuit of c).

Ref. [70] proved mathematically and Ref. [68] showed numerically that further simplifications can be done to such a line graph if tensors are diagonal. Since the ZZ gates in QAOA circuits are diagonal this leads to significant computational savings. The simplified line graph of illustration a) in Fig. 3.2 is pictured in part b) as an example. For more details on these simplifications the reader is addressed to read Refs. [68, 70].

The line graph representation can now be used to find a good contraction order of

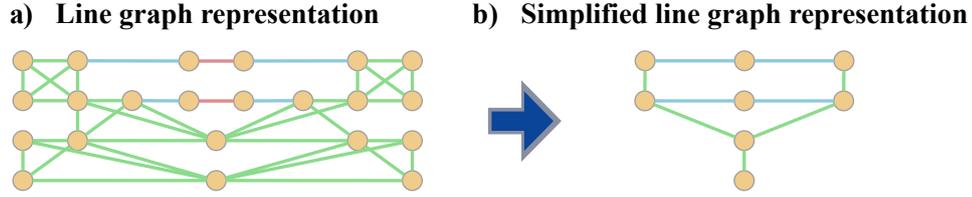


Figure 3.2: Illustration a) pictures the line graph representation of the tensor network from graphic d) in Fig. 3.1. The colors of the edges are identical to the colors of the nodes in the corresponding tensor network. This graph can be simplified to b) by using rules from Ref. [70].

the corresponding tensor network by calculating a tree decomposition of the graph with low treewidth [51]. A tree decomposition of a graph maps the graph into a tree structure with bags of vertices, ensuring every vertex and edge of the graph is included in at least one bag and that each vertex's bags form a connected subtree. The treewidth of the decomposition is the size of the largest bag minus one, and it measures how close the graph is to being a tree. For more information please see Ref. [73].

Having a tree decomposition of a line graph allows directly constructing a contraction order of the original tensor network with a maximal intermediate tensor rank equal to the treewidth of the tree decomposition [51]. Although finding tree decompositions with low widths is generally a NP-hard problem, it is still advantageous to use this approach for finding contraction orders because of the following reasons. On the one hand, the line graph of a QAOA tensor network can be significantly simplified as explained above. Therefore, good tree decompositions can be calculated with less computational effort compared to directly obtaining a suitable contraction order of the tensor network. On the other hand, finding tree decompositions with low treewidths is a common problem in probabilistic graph theory and thus allows knowledge transfer across different research fields [67]. Therefore, already established methods can be used to find tree decompositions with low widths, such as the Tamaki [74] or QuickBB [75] approaches. In the experiments of this thesis I use the latter method.

3.2 Different methods of QAOA parameter initialization

A common problem of QAOA is finding the actual or at least a good approximation to the minimum of the cost Hamiltonian of the given problem in the process of classical optimization [76]. It often occurs that rather a local minimum of the energy landscape with respect to parameters γ, β is found than the global one. It is actually

known that the number of low-quality local minima increases exponentially with the circuit depth p [55, 77].

Therefore, it is crucial that a good strategy is used to choose initial values of $\boldsymbol{\gamma}, \boldsymbol{\beta}$ in the beginning of the classical optimization routine within QAOA. This section introduces various means of choosing initial parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ of QAOA circuits. In particular, I will explain the methods *transition states* [78], *interpolation* [55], and *fixed angles* for the specific case of MAXCUT problems on regular graphs [79].

3.2.1 Transition states initialization

In general, transition states in an energy landscape $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$ are defined as stationary points with a single negative eigenvalue in the Hessian matrix [78]. A stationary point fulfills the terms $\partial_i E(\boldsymbol{\gamma}, \boldsymbol{\beta}) = 0$ for derivatives with respect to all variables. The Hessian matrix is defined by $H_{ij} = \partial_i \partial_j E(\boldsymbol{\gamma}, \boldsymbol{\beta})$.

More illustratively speaking, a transition state corresponds to the generalization of a saddle point in two dimensions to higher dimensions, where the energy landscape decreases only in one positive and negative direction starting from the transition state. This fact is nicely illustrated in Fig. 3.3 which is taken from Ref. [78]. From this Figure, it also becomes clear that moving from the transition state into both possible directions of the negative slope leads to two new minima.

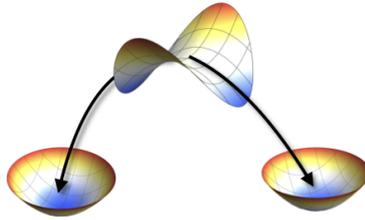


Figure 3.3: In the middle a transition state of an energy landscape is pictured. Following the two directions with negative slope leads to two new minima in the landscape. The graphic is taken from Ref. [78].

In Ref. [78] the authors show that transition states in a QAOA energy landscape with $p+1$ circuit layers can be constructed in the following way: Given the parameters $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) = (\gamma_1^*, \dots, \gamma_p^*, \beta_1^*, \dots, \beta_p^*)$ of a local minimum of the same QAOA with p layers, the with zero padded parameter vectors

$$(\gamma_1^*, \dots, \gamma_{j-1}^*, 0, \gamma_j^*, \dots, \gamma_p^*, \beta_1^*, \dots, \beta_{i-1}^*, 0, \beta_i^*, \dots, \beta_p^*) \quad (3.3)$$

lead to a transition state in the $p + 1$ energy landscape when $i = j$ or $j = i + 1$ $\forall i \in [1, p]$ and for $i = j = p + 1$.

Applied in practice, this means that for every local minimum of a p QAOA, Eq. (3.3) allows constructing $2p + 1$ transition states in the energy landscape of the corresponding $p + 1$ QAOA. From each of these transition states two local minima of the $p + 1$ energy landscape can easily be reached by following the vector with negative slope in both directions.

Because the parameters are padded only with zeros, the transition states of the $p + 1$ QAOA have the same energy as the local minimum of the p QAOA of which the parameters were taken. Therefore, when finding local minima starting from a transition state in the $p + 1$ case, it is guaranteed that the minima have lower energies than the local minimum in the p energy landscape.

By using this reasoning, the authors of Ref. [78] propose a greedy recursive strategy to find parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ of a QAOA that guarantees finding decreasing energy states with increasing depth p . For finding such a low energy state for a specific p the routine works as follows: In a first step a grid search is applied to find parameters $\boldsymbol{\gamma}, \boldsymbol{\beta} = \gamma_1, \beta_1$ that lead to a low energy state for the $p = 1$ QAOA. Because there are only two dimensions, this can be done efficiently. In the next step, these parameters are used to generate the three transition states according to the above introduced padding. Then, for each of the transition states the two corresponding local minima are found by following the direction with negative slope both ways. From these six new local minima, the lowest is chosen to create transition states for the $p = 3$ QAOA energy landscape. This procedure is repeated recursively until the desired depth p is reached.

For the remainder of the thesis I call this routine the *transition states* initialization method. To find a minimum, starting from a transition state, I use different gradient based standard algorithms as shown in the following Sections.

3.2.2 Interpolation initialization

Another method to initialize the QAOA optimization with suitable parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ consists of using interpolation. Here, interpolation means using parameters from lower depth QAOAs to find good parameters for a given depth p . Thorough research on the parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ of low energy QAOA states and the relation to each other with respect to different depths p has led to the following finding. Interpolating the parameters of lower depth circuits that reveal low energy states allows finding good initial parameters for higher depth circuits [55]. Thus, similar to the transition states case, the routine requires knowing parameters from lower depth circuits that lead to low energy states.

More technically speaking, starting from $p = 1$ QAOA, an optimization is conducted to find a low energy state with parameters $\boldsymbol{\gamma}, \boldsymbol{\beta} = \gamma_1, \beta_1$. In the next step the routine constructs the initial parameters for the $p = 2$ QAOA based on interpolating $\boldsymbol{\gamma}, \boldsymbol{\beta}$ to one more layer of gates. This procedure is repeated recursively until the de-

sired final depth of the QAOA is reached. Specifically, a well working interpolation rule from Ref. [55] is given by

$$\gamma_i^{p+1} = \frac{i-1}{p} \gamma_{i-1}^p + \frac{p-i+1}{p} \gamma_i^p, \quad (3.4)$$

where γ_i^{p+1} stands for the i -th entry of the initial γ value of the $p+1$ circuit and γ_i^p for the i -th entry of the optimized γ of the p QAOA. Furthermore, $\gamma_0^p = 0$ applies. For β the method uses the same interpolation strategy by replacing γ with β in Eq. (3.4).

This parameter choosing method is called *interpolation* initialization in this thesis. I use different gradient based strategies to carry out the optimizations after choosing the initial values.

3.2.3 Fixed angles initialization for MaxCut problems on k -regular graphs

In the specific case of QAOA for MAXCUT problems on k -regular graphs, I also use the fixed angles conjecture from Refs. [79, 80]. A k -regular graph consists only of vertices with exactly k neighbours. The fixed angles conjecture gives a worst-case guarantee for the approximation ratio when using specific parameters γ, β in a QAOA circuit for any MAXCUT on a k -regular graph. The corresponding parameters are given in Ref. [79], whereas the authors derived them by extensive numerical simulations of QAOA circuits.

Because the performance guarantees for the graphs looked at in the thesis are below 80% [79], I will use the fixed angles as initial parameters for QAOA optimization instead of directly applying them to calculate MAXCUT energies and results.

3.3 Parameter optimization

This section showcases experiments for analysing parameter optimization within tensor network simulations of QAOA. The experiments comprise executing different parameter initialization methods in combination with different classical gradient based optimization routines. The analysis allows choosing a suitable initialization method and classical optimization routine for using QAOA on the investigated problem types. Firstly, the experiments on MAXCUT problems are described in detail before experiments on MIS instances. Finally a short reasoning behind parameter optimization for the SETCOVER problem is given.

3.3.1 MaxCut parameter optimization

In the case of simulating QAOA circuits for MAXCUT problems, the analysis is conducted by benchmarking the final QAOA energies of the initialization methods from Sec. 3.2 in combination with different gradient based optimization routines with various learning rates against each other.

To be more precise, simulations of QAOA for depths $p \in \{1, 2, 3\}$ are carried out on 20 different random 3-regular graphs for the graph sizes $n \in \{50, 100, 150, 200\}$. In case of $p = 1$ the analytic formulas from Sec. 2.3.1 are used and $p \in \{2, 3\}$ circuits are executed using tensor networks as explained in Part 3.1. I simulate each of these instances with transition states, interpolation and fixed angles initialization of the QAOA parameters γ, β . These simulations in turn are conducted using each of the optimizers stochastic gradient descent (SGD), RMSProp and Adam with different learning rates (lr) to find minima in the QAOA energy landscape. For more details of these optimizers the reader is invited to read Ref. [81]. These optimizers are standard tools in machine learning to find minima in complicated landscapes and are all based on gradient descent methods. In every optimization routine, 50 steps are taken. Furthermore, I also used random initialization of the parameters with optimization as a benchmark. To have a fair comparison, random initializations on a specific problem instance are carried out as often as the transition states initialization uses different transition states at a given depth p . So for $p = 2$, three different random initializations are simulated, for $p = 3$ five different ones. Out of the results of different random initializations on the same graph, the best one is chosen for benchmarking. The results for the $n = 50$ instances are shown in Fig. 3.4.

In the Figure, for each combination of classical optimizer and parameter initialization the negative mean and corresponding standard deviation of the energies of the QAOA output states are plotted. The light colored bars correspond to the energies achieved by pure initialization without optimization. The full colored bars in turn show the energies after optimization. Therefore, in the case of $p = 1$, the transition states and interpolation initializations are only full colored since they use a grid search of the parameters and thus do not require an optimization.

As a result, one can conclude the following insights. For increasing p the energies of the QAOA output states decrease (remember, the Figure shows the negative energy values) systematically as expected. This fact resembles an incitement to benchmark the different depths as inputs for shrinking routines as it is done in the next chapter. Furthermore, at depth $p = 1$ neither random nor fixed angles initialization in combination with any classical optimizer achieve better results than a grid search but worse. Thus, at $p = 1$ a grid search is totally sufficient to find good low energy states.

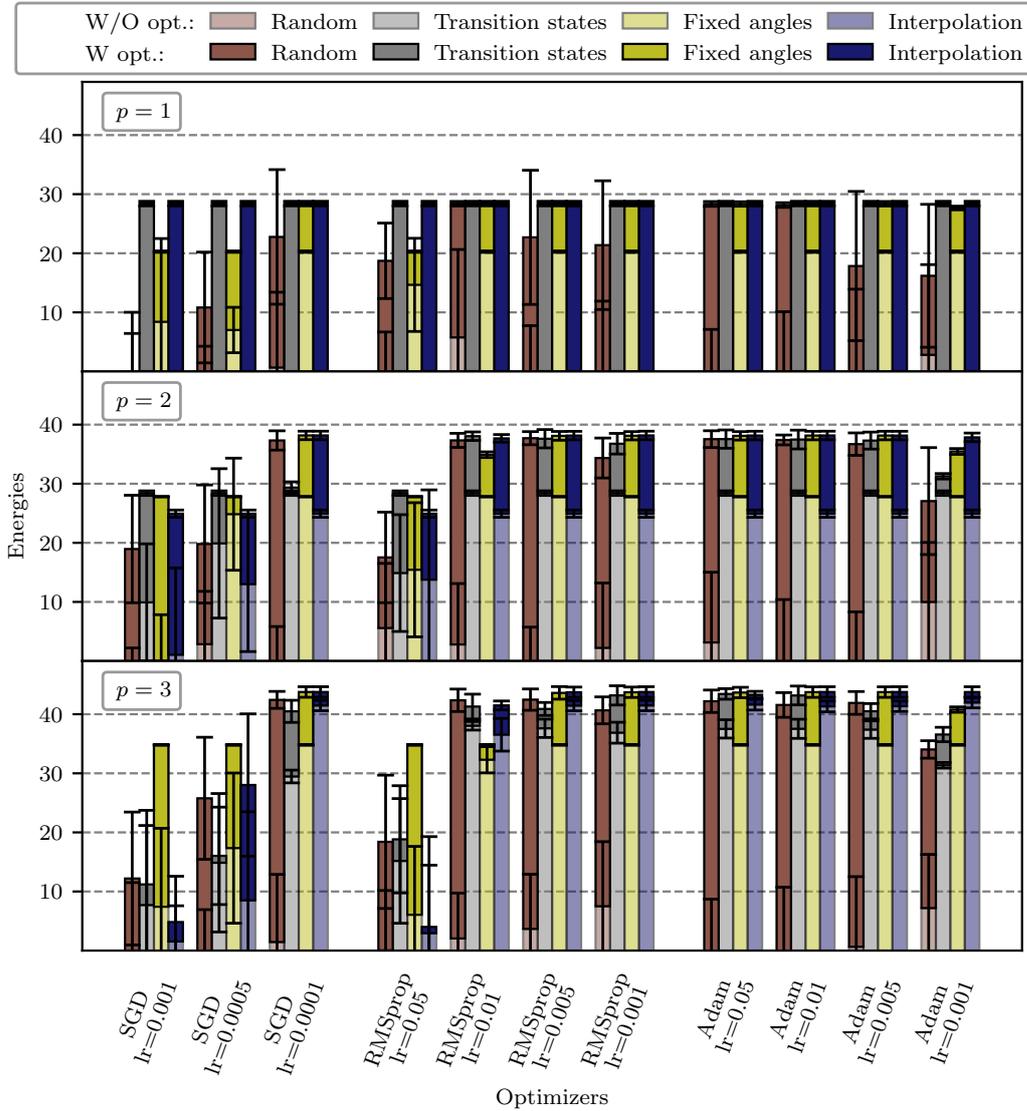


Figure 3.4: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MAXCUT problems with $n = 50$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

At depth $p = 2$ each of the three optimizers has at least one learning rate that resembles good results for every initialization, except for **SGD**. The latter does not find a low energy state in case of transition states initializations. Across all optimizers, interpolation and fixed angles initialization demonstrate the best results, but interpolation has mostly a slightly lower energy. Random and transition states initializations exhibit worse but still good results. For some optimizers, the former achieves lower energies but mostly the latter does.

In case of circuit depth $p = 3$ the results are similar to $p = 2$. Interpolation and fixed angles generally reach the lowest energies across all optimizers. Transition states achieve the next best results before random initialization.

Since these tensor network simulations are still computational expensive, it makes sense to also look at the loss curves achieved during the different optimization routines. This allows discovering a combination of initialization and optimizer that quickly reaches the low energy states in just a few optimization steps, i.e. a combination to save resources. The loss curves of the most promising combinations of initialization and learning rate are shown for every optimizer in Fig. 3.5. Here, the mean of the energies and corresponding standard deviation are used.

From the Figure follows that in general the **Adam** optimizer takes the most steps to find a low energy state. The interpolation method finds good parameters the fastest for both, the **RMSProp** and the **SGD** optimizer. However, also fixed angles initialization in combination with **SGD** and a learning rate of $\text{lr} = 0.0001$ leads quickly to low energy states. At this point it makes sense to recall that fixed angles initialization for $p = 3$ only requires one run of optimization, whereas interpolation first needs to optimize $p = 1$ and $p = 2$ to have good starting parameters for $p = 3$. Taken these insights together, I use the combination of fixed angles initialization and **SGD** optimizer with a learning rate of $\text{lr} = 0.0001$ for the remainder of the thesis whenever simulating QAOA of 3-regular MAXCUT instances with $n = 50$ nodes. This allows reaching good results with the fastest computational runtime when introducing a threshold during the optimization at which the process stops. In detail, I choose the threshold of five consecutive energies in the optimization process not changing by more than 0.25% with respect to the energy of the prior step.

The same analysis for problems of sizes $n \in \{100, 150, 200\}$ is given in Appendix B. Interestingly, for different system sizes, different optimizers exhibit different performances when benchmarking them against each other. Therefore, it makes sense to use varying optimizers and learning rates, depending on the system size. The analyses in this thesis give an overview of which combinations to choose when working with the same simulations and problem classes.

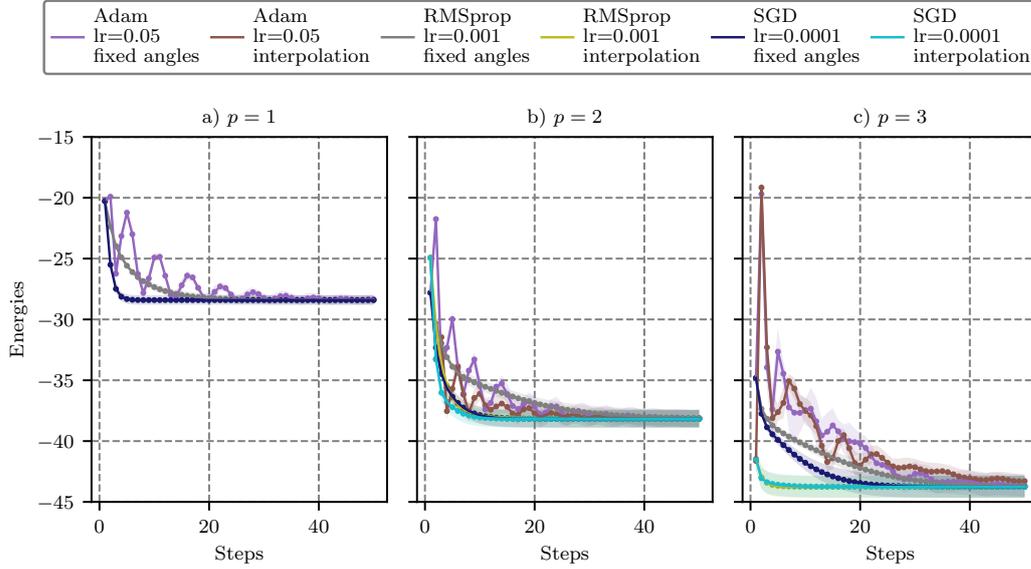


Figure 3.5: Loss curves during QAOA parameter optimization of the most promising combinations of initialization, optimizer and learning rate (lr) are shown. The losses correspond to the energy results of Fig. 3.4. Thus, mean and standard deviation of the losses during optimization of 20 different 3-regular MAXCUT graphs with $n = 50$ variables are illustrated.

3.3.2 MIS parameter optimization

Such an analysis is also required for the MIS problem because the energy landscape has different features. Without finding a suitable parameter initialization and optimizer, it is not guaranteed that QAOA finds low energy states. Therefore, I carry out similar experiments as in the previous Sec. 3.3.1. I apply $p \in \{1, 2, 3\}$ QAOA with different combinations of parameter initialization, optimizer and learning rate to 20 distinct random 3-regular MIS graphs and compare the obtained energies. In comparison to the MAXCUT case, here no fixed angles initialization is used because there is no prior work providing suitable angles. For the sake of clarity of the main part of the thesis, all results for graphs with $n \in \{50, 100, 150, 200\}$ are shown in Appendix C.

Evaluating the analysis reveals that the combination of transition states as well as interpolation initialization with the RMSProp optimizer with a learning rate of $lr = 0.005$ lead to the best results for all cases of p and n . Since using the interpolation initialization requires less computational resources, I will use this combination for the remainder of the work to evaluate QAOA on 3-regular MIS problems. When doing

so, I use the same threshold to stop the classical optimization as in the MAXCUT case. Optimization stops when five energies in a row during optimization do not change by more than 0.25% with respect to the energy of the prior step.

3.3.3 SetCover parameter optimization

For the SETCOVER problem, I mostly use $p = 1$ QAOA in this thesis. For $p = 1$ a thorough grid search with small enough steps already leads to low energy states. For the small experiments of $p = 2$ QAOA, interpolation initialization together with the RMSProp optimizer with a learning rate of $lr = 0.001$ is used throughout the thesis, given that this combination provides good results for both, the MIS and MAXCUT problems. However, during experiments I realized that it requires more than 50 optimization steps to find a low energy state. Therefore, for the SETCOVER experiments, I choose 100 optimization steps. Overall, I conduct no further analysis which initialization and classical optimizers are preferably used in the SETCOVER problem because of the small extent of SETCOVER experiments within this work.

Chapter 4

Application of recursive quantum optimization (RQO) algorithms to different problem classes

This chapter explains different experiments and results on using RQO algorithms on MAXCUT, MIS and SETCOVER problems. To be more precise, first, variations of RQAOA as introduced in Sec. 2.3.2 are applied to MAXCUT problems. Then I explain experiments and results of using QIRO algorithms from Sec. 2.3.3 on MIS instances. Lastly, the shrinking algorithm from Part 2.3.4 is employed to a real world use case from industry.

4.1 Variations of RQAOA for the MaxCut problem

This part of the thesis is strongly based on our publication [23] in which we applied the variation of RQAOA as explained in Sec. 2.3.2 to MAXCUT instances of different problem graph densities. By doing so, we are able to analyse the role of quantum correlations in recursive shrinking algorithms. Before explaining the experiments and findings thereof, I introduce two concepts of calculating classical correlations between variables in MAXCUT problems. This is necessary to understand the conducted experiments.

4.1.1 Linear programming (LP) and semidefinite programming (SDP) correlations for MaxCut problems

Instead of using correlations obtained from a quantum routine in shrinking algorithms as in the RQAOA, it is also possible to utilize correlations that were calculated by classical subroutines as it is done in Ref. [13]. Two options for this are linear programming (LP) and semidefinite programming (SDP) correlations. These are based on the classical standard routines to solve the MAXCUT problem as explained in Sec. 2.1.2. These correlations can then be used with the same update rules as the RQAOA to find solution approximations to MAXCUT instances.

LP correlations

Following the approach in Ref. [13], correlations from an optimal solution $\tilde{\mathbf{y}} \in [0, 1]^{|E|}$ to the linear program Eqs. (2.4a)–(2.4c) are computed using the transformation

$$b_e^{\text{LP}} := 1 - 2\tilde{y}_e \in [-1, 1]. \quad (4.1)$$

The rationale behind these correlations is that they are substantial when the edge weight is large or when cutting an edge does not conflict with other desired cuts. The term *conflict* refers to a situation where cutting one edge precludes the cutting of other edges that should also be cut, thereby assigning a lower correlation to such an interfering edge to discourage its cutting [23].

SDP correlations

The SDP correlation for each edge $\{i, j\} \in E$ is obtained from the GW vectors $\{\mathbf{v}_i\}$ as introduced in Sec. 2.1.2 by evaluating the dot product

$$b_{ij}^{\text{SDP}} = \mathbf{v}_i \cdot \mathbf{v}_j \in [-1, 1], \quad (4.2)$$

which measures the degree of alignment of the vectors [23]. A high absolute correlation indicates that the vectors are nearly parallel or anti-parallel, corresponding to dot products close to $+1$ or -1 . The relaxed formulation in Eq. (2.5) parallels the integer case, where an edge is cut if $x_i x_j = -1$ and not cut if $x_i x_j = +1$. The relaxed dot product correlations span from -1 to $+1$, and can be efficiently computed [23].

4.1.2 Experiments on Erdős-Rényi graphs

In the first experiment run, we applied the RQAOA-like shrinking algorithm with recalculation intervals from Chapter 2.3.2 to MAXCUT instances with different problem graph densities and recalculation intervals. We use both, classical and quantum correlations for the shrinking procedure in order to get an indication of the role of quantum correlations in this family of algorithms. First, the experimental framework is given before the results are stated and discussed.

Experimental framework

Instead of shrinking based on correlations obtained only from QAOA, we also applied the algorithm based on SDP and LP correlations in order to analyse the role of quantum correlations in shrinking routines. Thus, the algorithm used in the experiments can be illustrated as in Fig. 4.1 that is taken from Ref. [23]. For a given MAXCUT problem, correlations are calculated by either QAOA, LP or SDP before the problem is shrunk based on correlations by using update rules. There are

r shrinking steps before new correlations of the simplified problem are calculated again. This is recursively repeated until the remaining problem is trivial to solve. Finally the solution to the original problem is reconstructed by using information of the individual shrinking steps [23].

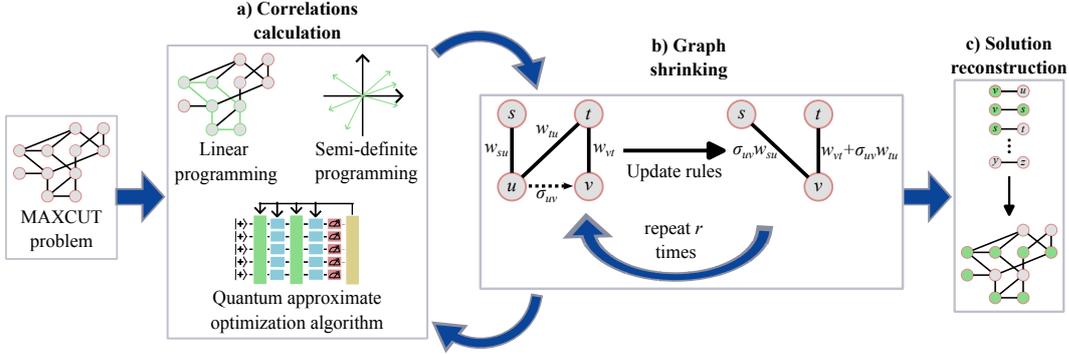


Figure 4.1: A summary of the shrinking routine used in the experiments is shown. For a given MAXCUT problem, correlations are calculated by either QAOA, LP or SDP before the problem is shrunk based on correlations by using update rules. There are r shrinking steps before new correlations of the simplified problem are calculated again. This is recursively repeated until the remaining problem is trivial to solve. Finally the solution to the original problem is reconstructed by using information of the individual shrinking steps. The illustration is taken from Ref. [23].

We conducted a first experiment run on 80 random Erdős-Rényi graphs with a hundred nodes and different densities. An Erdős-Rényi graph for a given number of nodes and density d is an undirected and unweighted graph with random edges. Each pair of nodes is connected by an edge with probability d . Each of the problem graphs we solved by applying the shrinking algorithm with recalculation intervals of $r \in \{1, 10, 50, \infty\}$ for any of the three means of calculating correlations. In case of quantum correlations, we used $p = 1$ QAOA. Furthermore, as a benchmark we also solved the problem graphs using the bare algorithms of the correlation calculations, i.e., linear programming, GW and QAOA, as explained in more detail in Ref. [23].

To benchmark the different performances against each other we use the median of the approximation error, whereas the latter is defined by

$$R_A := \frac{S_A}{S_G}. \quad (4.3)$$

Here, S_A stands for the cut size obtained by the investigated algorithm. S_G represents the cut size retrieved from Gurobi [82], a state-of-the-art commercial solver. For each instance, we run Gurobi on a single core (Dual AMD Rome 7742) and termin-

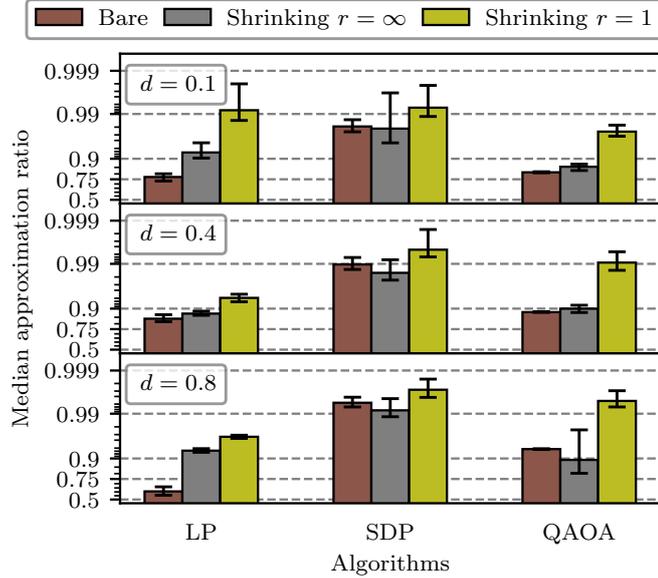


Figure 4.2: We report the median approximation ratio R_A for the bare LP, GW, and QAOA ($p = 1$) algorithms, alongside their shrinking variants with recalculation intervals $r = 1$ and $r = \infty$. These algorithms were tested on 80 randomly generated Erdős-Rényi graphs, each with 100 nodes, across densities of 0.1, 0.4, and 0.8. The error bars indicate the first and third quartiles. The Figure is taken from Ref. [23].

ate the optimization when the value of the optimization objective has not changed in an hour [23]. It should be noted that using this method does not ensure that the Gurobi optimizer reaches optimal solutions, particularly for dense Erdős-Rényi graphs. Nonetheless, we consider the Gurobi benchmark robust enough to meet the objectives of this study [23].

Results

The comparison of the bare algorithm with the shrinking algorithm can be found in Fig. 4.2, taken from our publication [23]. Here only densities of $d \in \{0.1, 0.4, 0.8\}$ and recalculation intervals $r \in \{1, \infty\}$ are shown.

Analyzing the LP algorithm, we observe that the shrinking technique without recalculations consistently outperforms the bare algorithm across all graph densities, despite using the same correlation values. Moreover, increasing the frequency of recalculations (thereby decreasing r) significantly enhances performance. This pattern is similarly observed in the SDP and QAOA algorithms. However, unlike SDP and QAOA, which maintain a nearly constant approximation ratio across densit-

ies, LP shows notably superior performance on lower-density graphs compared to high-density ones [23].

The bare GW algorithm and the SDP-based shrinking algorithm exhibit robust performance across all densities. Particularly, the shrinking algorithm with a recalculation interval of $r = 1$ achieves median approximation ratios above 99% for all densities, outperforming the others. Notably, the bare GW algorithm marginally surpasses the shrinking algorithm when no recalculations are performed.

In the case of QAOA, the shrinking method with a recalculation interval of $r = 1$ (equivalent to RQAOA) substantially improves performance over the bare QAOA algorithm. For varying densities, the median approximation ratio leaps from around 90% in the bare QAOA to nearly 99% with the $r = 1$ shrinking technique. This significant enhancement aligns with previously reported results [7, 11]. Remarkably, QAOA shows the most substantial relative improvement among the three methods of correlation computation compared to its bare version. Although the bare QAOA and the non-recalculating shrinking method yield similar results, the error bars indicate that the quality of the shrinking method without recalculations fluctuates more than the standard QAOA algorithm.

To analyse the effects of recalculation intervals and problem densities, we show the median approximation ratio of the different correlation calculations for all densities and recalculation intervals as explained above in Fig. 4.3, taken from our publication [23].

Let us first examine the general impact of the recalculation interval before analyzing how problem density affects performance. Across all three methods for computing correlations, a clear pattern emerges: More frequent recalculations yield better solutions. Notably, for LP correlations at low densities, the $r = 10$ shrinking algorithm slightly outperforms the $r = 1$ variant. Although the performance differences between $r = 10$ and $r = 1$ are minor for LP and SDP, QAOA shows a significant performance boost with $r = 1$. For all methods (SDP, LP, and QAOA), recalculation intervals of 50 and ∞ result in significantly poorer approximation ratios compared to smaller intervals, underlining the necessity of frequent recalculations.

Now, focusing on problem density, LP correlations exhibit the most pronounced dependence. Specifically, the approximation ratio for low-density instances is substantially higher than for any other type of correlation, which aligns with the generally strong performance of the bare LP algorithm on sparse graphs, as demonstrated in other studies such as Ref. [28].

Additionally, Fig. 4.3 reveals a systematic increase in the median approximation ratio for densities above $d = 0.3$. This trend likely reflects the diminishing quality of reference solutions obtained via Gurobi for denser instances, rather than an actual improvement in the shrinking algorithm’s performance. While Gurobi can certify optimal solutions for sparser graphs, it struggles with denser ones. Consequently, we attribute the rising approximation ratios to our choice of benchmark. Neverthe-

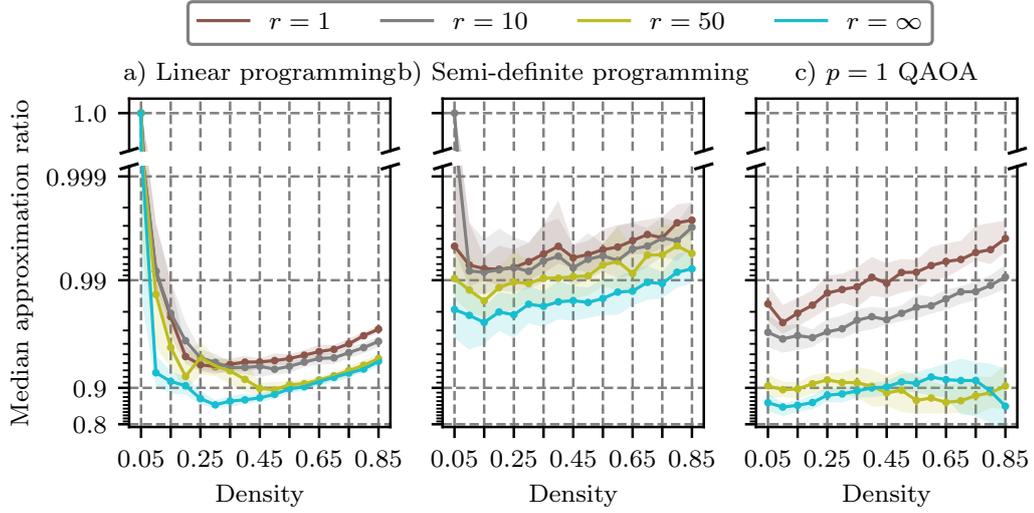


Figure 4.3: We compare the median approximation ratios of the shrinking algorithm using correlations derived from LP, SDP, and QAOA ($p = 1$). Each density between 0.05 and 0.85 was tested on eighty 100-node Erdős-Rényi graphs with varying recalculation intervals $r = 1, 10, 50, \infty$. The shaded regions illustrate the interquartile range, capturing results between the first and third quartiles. The Figure is taken from Ref. [23].

less, this setup still permits meaningful comparisons between different correlation methods.

Thus, we observe that the performance of the LP shrinking algorithm worsens rapidly as density increases from $d = 0.05$. In contrast, SDP and QAOA correlations provide more stable performance across the entire density range. Furthermore, SDP correlations consistently outperform QAOA ($p = 1$) for most densities, while QAOA surpasses LP correlations for densities greater than $d = 0.2$.

Discussion

In these experiments, we analyzed algorithms that solve combinatorial optimization problems through a recursive shrinking technique. We compared various methods for computing correlations, both quantum and classical, thereby establishing classical benchmarks that quantum-informed shrinking algorithms must surpass to demonstrate practical utility.

Our proposed shrinking procedure serves as a standalone heuristic that signific-

antly enhances the performance of the underlying algorithm. Through extensive numerical simulations, we applied this procedure to the MAXCUT problem.

Our results reveal that the shrinking algorithm not only boosts the performance of the quantum QAOA algorithm, as previously shown in [7, 11], but also markedly improves the performance of classical algorithms such as the Goemans-Williamson algorithm and the linear programming relaxation of MAXCUT.

By applying the shrinking algorithm to Erdős-Rényi graphs of various densities, we observed that LP correlations perform exceptionally well at low densities but degrade rapidly as density increases. This indicates a simple decision rule: Use LP correlations for low-density problems and switch to other methods for higher densities. This sensitivity is due to the LP's edge-based decision variables, which proliferate with increasing graph density, unlike the node-assignment variables used in other relaxations [23].

In contrast, SDP and QAOA correlations maintain consistent performance, with approximation ratios improving slightly as density increases. This trend may result from either enhanced solution quality of the shrinking algorithm or declining performance of the Gurobi reference solutions.

For all correlation methods, increasing the frequency of recalculations generally improves results, albeit at the cost of greater computational resources. Notably, the LP and SDP-informed shrinking algorithms show only slight performance gains when recalculations are performed at every step ($r = 1$) compared to every ten steps ($r = 10$) [23].

4.1.3 Experiments on 3-regular graphs with higher depth QAOA

To explore the performance of shrinking algorithms using quantum correlations of better quality, we conducted another experiment run on solving MAXCUT instances with the same routine as before but with higher depth p QAOA correlations. To achieve this we used sparser and smaller problem graphs. The experimental setup and results are explained in the following. Lastly, the results are discussed.

Experimental framework

To investigate the performance of the shrinking algorithm with higher depth $p \in \{1, 2, 3\}$ QAOA correlations, we apply the same algorithm as in the previous case with recalculation intervals $r \in \{1, 5, 10, 25, \infty\}$ to 25 different 3-regular MAXCUT graphs with 50 variables. The shrinking algorithm is employed five times for every instance. We again use the approximation ratio with respect to solutions obtained by Gurobi. In the case of 3-regular graphs, Gurobi manages to solve the instances to optimality.

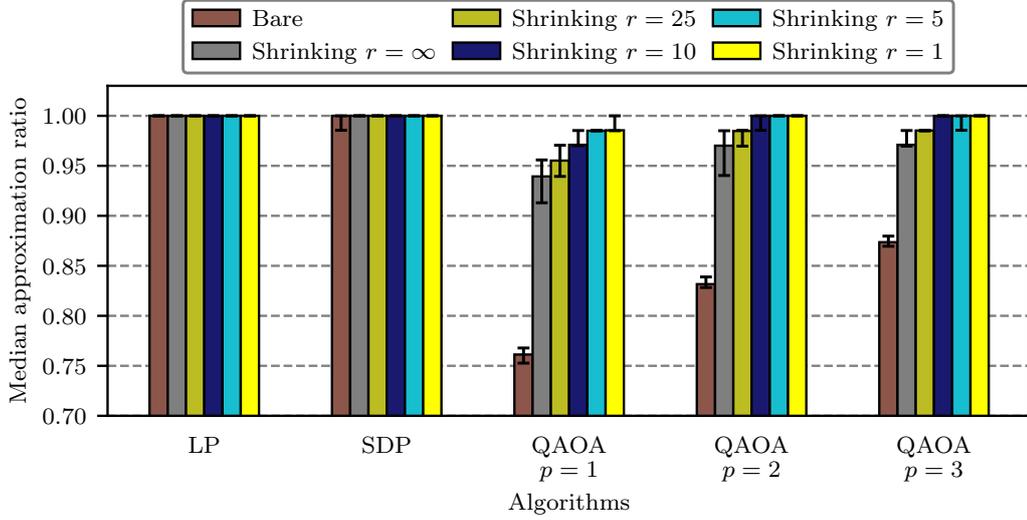


Figure 4.4: This Figure illustrates the median approximation ratios for the bare LP, GW, and QAOA algorithms (with depths $p \in \{1, 2, 3\}$) and their shrinking variants using recalculation intervals $r = 1$ and $r = \infty$. These algorithms were applied to 25 randomly generated 3-regular MAXCUT problem instances, each containing 50 nodes. The bars represent the median results from five runs per instance, while the error bars denote the first and third quartiles of the individual approximation ratios. The approximation ratios are calculated relative to the optimal MAXCUT solutions. The illustration is an extension to Fig. 5 in Ref. [23].

Results

Fig. 4.4 presents the outcomes of our experiments. For comparative purposes, we also include results derived from LP and SDP correlations. The Figure is an extension to Fig. 5 in Ref. [23].

In the Figure, the bar heights represent the median approximation ratio across all problem instances and algorithm runs. The upper and lower error bars correspond to the third and first quartiles of the obtained approximation ratios, respectively. Due to the randomness in tie-breaking during the shrinking process, we executed the shrinking algorithm multiple times for each instance, resulting in potentially different final solutions. This variability is especially pronounced in QAOA, where at low depths p , the correlations are influenced only by the local neighborhood of an edge. For 3-regular graphs, the limited number of distinct local neighborhoods leads to numerous ties in the correlations [65, 83].

Critically, the results in Fig. 4.4 validate the expectation that deeper QAOA cir-

cuts yield better correlations. This is reflected in the improved performance of the bare algorithms as the depth p increases. Furthermore, this performance enhancement extends to the shrinking algorithm, regardless of the recalculation frequency. Consistently, the shrinking algorithm’s performance improves when correlations are recalculated after each shrinking step. Remarkably, we achieved optimal solutions for all tested graphs at $p = 2$ when recalculations were performed after every shrinking step. However, it is anticipated that larger instances would necessitate higher depths. Additionally, even without recalculations ($r = \infty$), the shrinking algorithm generally outperforms the bare algorithm at higher depths. It can be nicely seen that decreasing recalculation interval r leads to an improvement in the performance for all depths.

Given the low density of 3-regular graphs, these instances fall within the density range where the LP shrinking algorithm and its classical counterpart exhibit superior performance, as just discussed above. Consequently, all LP variants reach optimal solutions. Moreover, the problem sizes are sufficiently small that all SDP algorithms achieve excellent approximation ratios, with median ratios at 100%. The SDP shrinking variant with $r = \infty$ solves more instances perfectly than the bare algorithm, but only the shrinking versions with $r = 1$ and $r = 5$ achieve optimal results for all MAXCUT instances.

Discussion

Our simulations of higher-depth QAOA have confirmed that better quantum correlations significantly improve the performance of the shrinking algorithm. This result suggests that advancements in quantum hardware will lead to substantial performance improvements in shrinking algorithms. Consequently, this provides valuable insights into the potential of quantum algorithms to offer practical solutions for combinatorial optimization problems [23].

4.2 Higher-depth variations of QIRO for the MIS problem

In this section, I showcase results of experiments on applying the QIRO variations of Part 2.3.3 to MIS problems of various sizes. First I explain the details of the experiments before showing the results and discussing them.

4.2.1 Experimental framework

For benchmarking the performance of the different QIRO algorithms MAXQ, MINQ, MMQ and 2-MMQ from Sec. 2.3.3, I apply them to MIS problems of various sizes. To be more precise, each of the algorithms is used to solve 40 different instances of 3-regular graphs of sizes $n \in \{60, 80, 100, 120, 140, 160\}$. I carry this out for every of the

QAOA depths $p \in \{1, 2, 3\}$ within the QIRO routines. For the QAOA optimization routines the `RMSProp` optimizer with a learning rate of $lr = 0.005$ in combination with the interpolation parameter initialization is used as described before in Chapter 3.3.2. For $p = 1$, I simulate QAOA via the analytic expressions. For depths $p \in \{2, 3\}$, tensor network simulations are used as explained in Sec. 3.1.

4.2.2 Results

The results of the experiments are shown in Fig. 4.5. For each of the four algorithms MAXQ, MINQ, MMQ and 2-MMQ I use the mean of the approximation ratio

$$R_A := \frac{S_A}{S_G} \quad (4.4)$$

as performance measure. Here, S_A stands for the MIS size obtained by the investigated algorithm. S_G represents the MIS size retrieved from Gurobi [82]. For the present problems, Gurobi is able to find the optimal solutions.

The mean is calculated with respect to the results of all 40 problem instances for each problem size. The shaded areas in the Figure indicate the maximum and minimum achieved approximation ratio of the individual problems. Furthermore, Fig. 4.5 shows the results of the classical MAX and MIN algorithms.

A first interesting insight is that MINQ and MAXQ based on $p = 1$ QAOA match the respective classical greedy variant perfectly as suggested by Ref. [16]. This indicates that the experiments are conducted correctly. MINQ achieves a lot better results than MAXQ for $p = 1$ over every problem size.

Across all algorithms the performance improves strongly with increasing circuit depth p . For both, MMQ and QIRO the $p = 1$ version performs worse than the classical MIN algorithm as already shown numerically in Ref. [12]. However, as can be seen from the results, already using $p = 2$ QAOA for obtaining correlations is sufficient to surpass MIN.

When comparing the approximation ratios of the different algorithms with each other, MAXQ performs generally worse than the others with same depth p . For depth $p = 1$, MINQ clearly achieves the best results with a mean approximation ratio of about 0.96. Here, MMQ and 2-MMQ perform worse with roughly 0.93 to 0.94. In case of QAOA circuits with depth $p = 2$, the mean approximation ratios of MINQ, MMQ and 2-MMQ lay on the same level and depending on the problem size one of them performs slightly better than the others. The same behaviour can be seen for $p = 3$ QAOA circuits. Here, MAXQ achieves worse results than the other algorithms but MINQ, MMQ and 2-MMQ demonstrate approximation ratios on the same level.

When analysing the results along the different problem sizes, all algorithms exhibit a rather stable performance except for $p = 1$ MAXQ, MMQ and 2-MMQ that perform

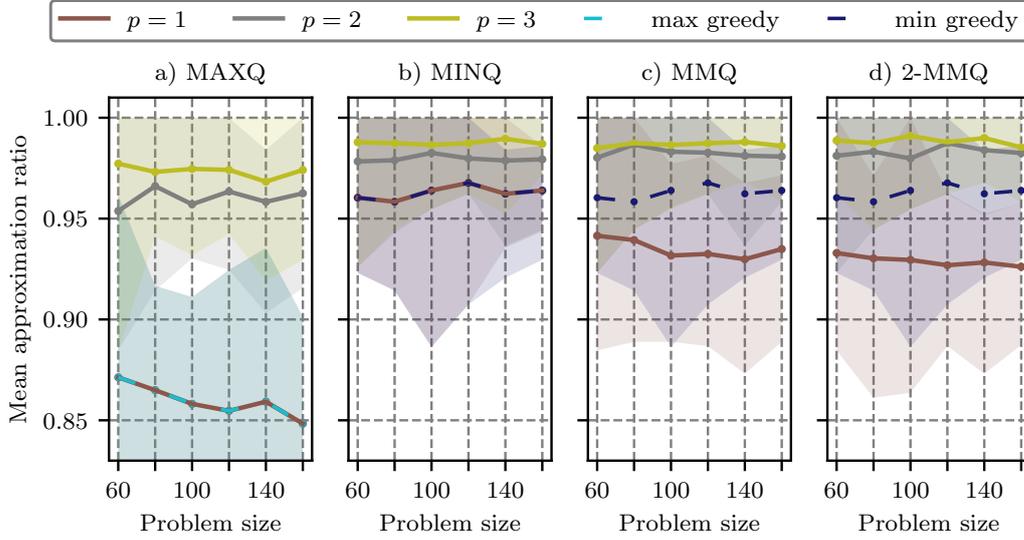


Figure 4.5: The mean approximation ratio achieved by the algorithms MAXQ, MINQ, MMQ and 2-MMQ for MIS are shown for different problem sizes and QAOA circuit depths $p \in \{1, 2, 3\}$. The approximation ratio corresponds to the proportion of the found MIS size and the optimal solution. The mean is taken over 40 different 3-regular problems for each of the sizes $n \in \{60, 80, 100, 120, 140, 160\}$. The shaded areas indicate the maximum and minimum achieved approximation ratio over all individual problem instances. The classical MIN and MAX algorithms are pictured as benchmarks.

worse for bigger problems. All other combinations of algorithm and QAOA depth p show no significant decreasing mean approximation ratio for increasing system sizes.

It is also interesting to analyse how many MIS instances are solved to optimality by the respective algorithm. The percentage of the perfectly solved instances in the same experiments is shown in Fig. 4.6.

The results of solved instances mimic in general the performance with respect to the mean approximation ratio. Thus, also here the performance of all routines increases strongly with higher depth p . MAXQ generally performs worse than the other three algorithms which are for the same p mostly on the same level. However, with increasing problem size the routines solve percentually less instances perfectly.

4.2.3 Discussion

By conducting the explained experiments, I provide further insights into the performance of QIRO routines on larger MIS problems, especially with respect to higher

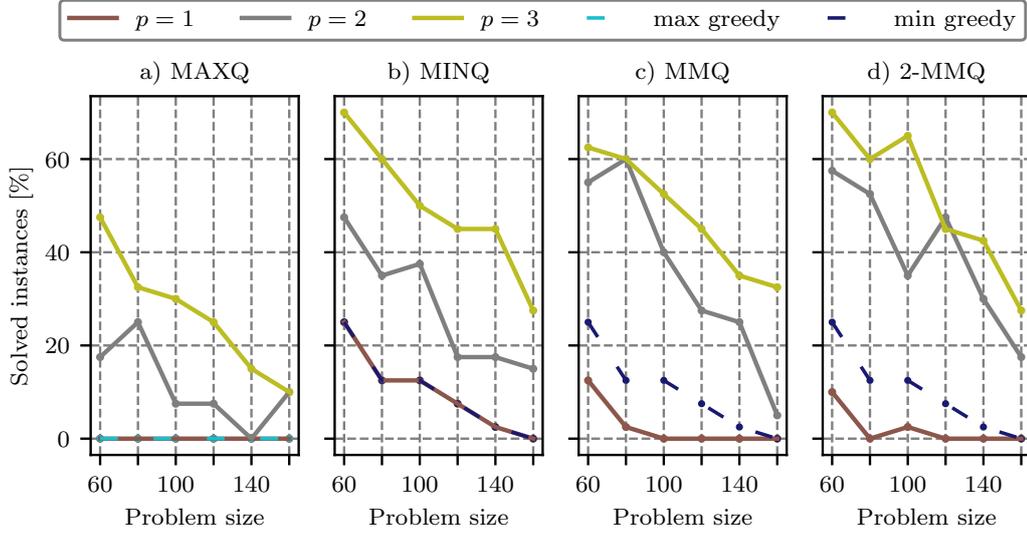


Figure 4.6: The percentage of solved MIS instances is shown for the different algorithms MAXQ, MINQ, MMQ and 2-MMQ. The experiments are conducted on 40 different problem instances for each of the sizes $n \in \{60, 80, 100, 120, 140, 160\}$. The algorithms with QAOA circuit depths $p \in \{1, 2, 3\}$ are used.

depth QAOA correlations, i.e., quantum correlations of better quality. Benchmarking the different routines MAXQ, MINQ, MMQ and 2-MMQ allows selecting an algorithm for further experiments with knowing its performance compared to the others.

The result of improved performance of all QIRO routines with increasing p is expected but has only been verified before for system sizes of up to 18 variables [16]. Thus, the present experiments further verify these expectations for larger problem instances.

The worse results of the MAXQ algorithm compared to the other routines strengthens the judgment to not choose this algorithm when solving MIS problems. In addition to its bad performance, it also requires longer computational run times than for example the MINQ algorithm because it removes only one node per shrinking step.

The fact that MINQ, MMQ and 2-MMQ perform similar for depths $p \in \{2, 3\}$ provides us with the information that in case of 3-regular MIS problems, two-point correlations do not contain helpful information for the present shrinking algorithm. Furthermore, it is interesting to see that the additional information available to MMQ in comparison to MINQ does not lead to a better performance.

Moreover, the experiments show that the increase in QAOA circuit depth, thus increase in correlation quality, not only improves the mean approximation ratio of MIS solutions but also improves the number of perfectly solved instances. When examining the relative increase of solved instances with increasing p , MMQ and 2-MMQ exhibit the strongest increase. However, this is not due to more solved instances but rather because of the worse $p = 1$ results in comparison to the MINQ results.

4.3 Industry use case: Application of a QIRO algorithm to a sensor positioning problem

This section explains experiments of applying the MINQ algorithm from Part 2.3.4 for the SETCOVER problem to the industry use case of solving a sensor positioning problem. This problem can be mimicked by the SETCOVER problem as shown previously in Sec. 2.1.4. First, I explain the experimental setup before introducing the results thereof and discussing them.

4.3.1 Experimental framework

The instances of the experiment are chosen in such a way that the resulting SETCOVER problems mimic the sensor positioning issue in two dimensions. The instances are created as follows. First, a two dimensional grid is chosen with a random size of maximal 12×12 grid points. The grid represents a two dimensional room that needs to be surveyed by sensors. On every grid point an area to cover is positioned with a probability of 60%. These areas correspond to the elements of the set U that needs to be covered. Next to each grid point sensors with a covering range of 1, 2, 3 or 4 are appointed randomly with probabilities of 70%, 10%, 10% and 10%, respectively. A sensor with a range of 2 corresponds to covering all areas that lie within the 2-neighbourhood, i.e., the area on the grid point itself and the nearest vertical and horizontal areas thereof. Accordingly, the other ranges of sensors cover neighbouring areas. A 1-neighbourhood consequently consists only of the possible area on the grid point. These sensors correspond to the subsets V_i of the problem, that are allowed to be used to cover the set U . Such a SETCOVER problem is only used in the experiments when all elements of U can be covered by the subsets V_i .

On the one hand, creating SETCOVER instances in this way imitates the sensor positioning problem in such a way that not every area in a room needs to be surveyed by a sensor. On the other hand, appointing sensors with different ranges on not every possible grid point mimics that sensors cannot be placed everywhere and that there can be obstacles in the room that lead to different ranges the sensor can cover.

For the experiments, 10000 instances of valid SETCOVER problems are created.

The size of sets U range from 2 to 59, leading to up to 400 variables and thus qubits. To each of these instances the MINQ routine for SETCOVER problems with $p = 1$ is applied as introduced in Sec. 2.3.4. As a benchmark also the greedy MAX solver for SETCOVER instances from Chapter 2.1.4 is applied to all problems. For avoiding confusion to the reader, it is again noted that MINQ for the SETCOVER problem is just a name of the described algorithm in alignment with the MINQ routine for MIS problem. Since the name has no further underlying meaning, the question why a MIN type of algorithm is compared with MAX routine is obsolete. Furthermore, all generated problem instances with a set size $|U|$ smaller or equal than 15 are also solved by MINQ using $p = 2$ QAOA.

4.3.2 Results

Figure 4.7 shows the results of the experiments. On the x-axis the size of the sets U that need to be covered is plotted. The left y-axis illustrates the mean of the number of possible subsets V_i for each problem instance of corresponding size $|U|$. This information is important to know because the number of possible subsets has a crucial influence on the difficulty of the problem and the number of required qubits. One can see that the number of possible subsets roughly increases linearly with the size of the set U .

The right y-axis indicates the mean approximation ratio of the result of the corresponding algorithm. The approximation ratio is here again calculated with respect to the optimal solution to the problem that is obtained using Gurobi [82]. For example, an approximation ratio of 2 corresponds to a solution to the SETCOVER problem with twice as many chosen subsets V_i to cover U as necessary in the optimal case. The individual data points of both y-axes show the mean of all instances with corresponding set size $|U|$. The shaded areas indicate the standard deviation to the mean value.

Interestingly, the greedy MAX solver already solves almost all problem instances perfectly. In contrast, the $p = 1$ MINQ only is able to solve problems with very small sizes to optimality. For set sizes of bigger than five, the mean approximation ratio quickly increases up to roughly 1.6 for sizes of $|U| = 50$. However, MINQ with correlations obtained from $p = 2$ QAOA performs already a lot better such that the mean approximation ratio stays below 1.1 for the solved instances.

4.3.3 Discussion

On one hand these results show that RQO algorithms such as QIRO allow simulations of quantum algorithms on large problems of up to 400 variables already now. Thus, applying this family of algorithm on near-term quantum devices seems feasible. On

4.3 Industry use case: Application of a QIRO algorithm to a sensor positioning problem

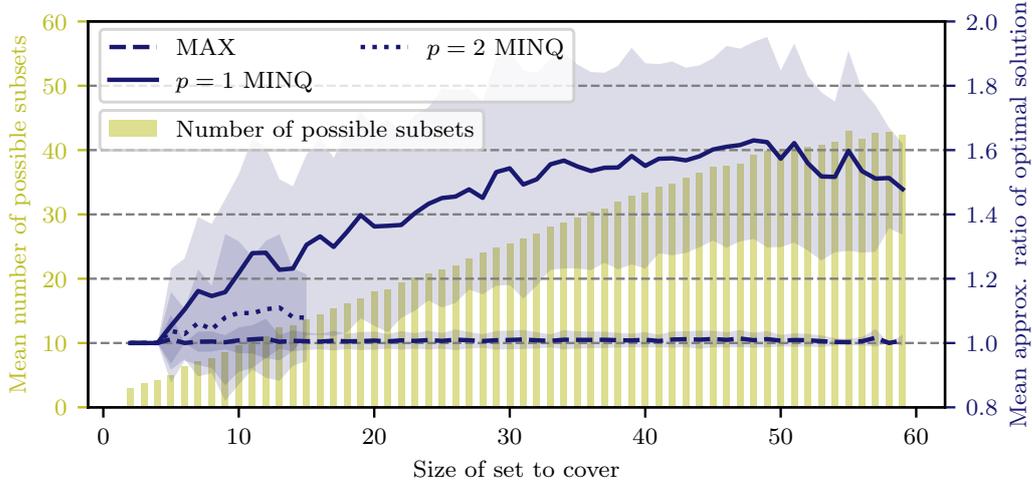


Figure 4.7: On the x-axis 10000 instances of the SETCOVER problem are divided by the size of the set U . The left y-axis illustrates the corresponding mean of the number of possible subsets V_i that can be chosen to cover U . The right y-axis shows the mean approximation ratio of the problem solutions obtained by the MAX and MINQ algorithms. For the latter the $p = 1$ MINQ is applied to all instances and the $p = 2$ variation only to instances with set size $|U| \leq 15$. The approximation ratio is the ratio of the obtained solution and the optimal solution.

the other hand the experiments indicate that not only simple toy problems but also difficult ones such as the SETCOVER can be investigated by quantum routines.

However, given that a simple classical greedy algorithm is able to solve most problem instances perfectly, the performance of $p = 1$ MINQ rather disappoints. One reason for the bad results is the unnatural encoding of the SETCOVER problem using many ancilla qubits. One of the underlying ideas behind recursive quantum algorithms is the natural encoding of the problem in terms of qubits and graphs that is not given for the present problem.

As already shown in the other experiments of the thesis, quantum correlations with higher quality improve the performance of the quantum routine. This result resembles another indication that better quantum correlations, e.g., higher depth QAOA correlations, help improving shrinking routines. The very strong performance increase of MINQ at already $p = 2$ QAOA correlations is an evidence for the importance of further developments of quantum hardware. The more tolerant against faults and the longer the coherence times of hardware become, the better quantum information can be obtained, leading to strong performance increases of algorithms.

Chapter 5

Summary, conclusion and outlook

In this chapter a summary of the thesis and a conclusion thereof is written. In addition, based on the results of this thesis I give an outlook to further research questions on RQO algorithms that can be explored in the future.

5.1 Summary

The purpose of this thesis consists of investigating the role of quantum correlations in recursive optimization algorithms and exploring the performance of QIRO routines with high quality quantum correlations on large problem instances. Furthermore, a RQO algorithm for a real word industry use case is derived and applied.

To achieve these goals, the problem classes MAXCUT, MIS and SETCOVER are introduced in the beginning as suitable examples. For each of the problems the corresponding QUBO formulations are given and at least one classical solver serving as benchmark is explained. Then I establish the basics of tensor networks and how to use them for simulating quantum circuits. In the final part of the background section the quantum optimization algorithms relevant for the thesis are introduced, namely QAOA, RQAOA variations for MAXCUT, QIRO adaptations for MIS and SETCOVER problems.

Before being able to apply these algorithms to the problems, it requires an analysis on which parameter initialization methods and which classical optimizers are suitable for achieving good QAOA results since QAOA serves as a subroutine to all recursive algorithms of interest. Therefore, I conduct QAOA simulations using tensor networks with different parameter initialization methods and classical optimizers on the problem classes with various sizes. From the results it is possible to choose a combination of parameter initialization method and optimizer for the individual problems and instance sizes that lead to good QAOA performance. This can also serve as a practical guide for future implementations of large scale QAOA simulations based on tensor networks.

In the next step of the thesis, the quantum optimization and classical benchmark routines are applied to the corresponding problems to gain insights into their performance. Concretely, I use quantum and classical variations of RQAOA to solve

MAXCUT instances of various densities. On one hand the focus lies on comparing the algorithm's performance based on quantum correlations with the one using classical correlations. On the other hand the performance increase with higher depth QAOA correlations is investigated.

In addition, QIRO variations with QAOA depths $p \in \{1, 2, 3\}$ are applied to MIS instances of different sizes of up to 160 variables. These experiments allow further insight into the performance of QIRO algorithms on larger graphs with increasing quantum correlation quality.

Lastly, SETCOVER instances imitating a sensor positioning problem from industry with up to 400 variables, i.e., 400 qubits, are created and solved by a QIRO algorithm with $p = 1$ QAOA correlations. In addition, the algorithm with $p = 2$ is applied to the smaller instances.

5.2 Conclusion

The results achieved in this thesis showcase the role of quantum correlations in recursive shrinking algorithms for solving COPs, the performance of QIRO algorithms on MIS problems and on the use case of sensor positioning.

When exploring the role of quantum correlations in comparison to classical ones, we found that the shrinking routines as such already improve the performance of the underlying classical as well as quantum routines. Furthermore, we showed that using classical LP correlations when solving COPs of low graph density with shrinking algorithms leads to the best result in comparison with correlations calculated by SDP or QAOA. The latter two variants exhibit constant performance over varying problem densities. In addition, we found that the shrinking algorithm works better the less shrinking steps we do between calculating new correlations, independent of the way of obtaining correlations. However, the performance gains can be very small. Given the computational effort of calculating correlations more frequently, it may still be good enough to calculate less often, depending on the desired performance. Although our results show that SDP correlations lead to better approximation ratios than $p = 1$ QAOA correlations, we demonstrate that higher depth QAOA lead to a strongly improved performance of the shrinking algorithm. This result indicates the potential of quantum algorithms in solving COPs when hardware improves with time.

Following on from this fact, the results of applying QIRO routines to MIS problems display the performance gain when the quality of the quantum correlations increases, i.e., when higher depth QAOA correlations are used. While $p = 1$ QAOA leads to QIRO variants of worse or same performance as classical greedy algorithms, the routines using higher depth QAOA outperform the classical greedy benchmarks. I also showed that using QIRO algorithms based on two- and one-point correlations

instead of only one-point correlations does not lead to a better performance in our experiments. For $p = 1$ the algorithms with access to more information even demonstrates worse approximation ratios. However, this may be due to the particular choice of the update rules employed here. Improving upon these may be a promising avenue for future research.

Lastly, our experiments on SETCOVER problems indicate that quantum shrinking algorithms can already now be applied to more difficult use cases from industry and not only simple or even toy problems. However, the fact, that a classical greedy algorithm solves the instances of our experiments already to almost optimality, demonstrates their simple nature. Nevertheless, QUIRO with $p = 1$ correlations performs disappointingly bad. I conclude that the reasons are the non-native encoding of the problem including ancilla variables in terms of qubits and the difficult energy landscape that complicates QAOA optimization. Thus, the results suggest that having a native mapping to a quantum device is essential. As before, the results indicate that higher depths lead to better results also in case of the SETCOVER problem. Therefore, I infer that it is crucial to improve the quality of quantum correlations for the successful application of RQO algorithms to real world industry problems.

5.3 Outlook

This thesis conducted research on the performance of RQO algorithms for MAXCUT, MIS and SETCOVER problems with a focus on large sparse graphs. A natural next step would be to apply similar experiments to denser problem graphs to investigate the robustness of the algorithms with respect to the degree of problem connectivity.

Another possible extension is to also look into other combinatorial optimization problems and weighted versions thereof. This would require developing different update rules but would allow further insights into the applicability of such algorithms on various problem classes instead of only the standard ones. For the sake of useful applications, looking for further problems that are of importance to industry would be advantageous.

A further step forward would be to extend the investigated quantum algorithms by new developing update rules, that are based on information going beyond one- and two-point correlations. This would be especially interesting when further increasing the depth of QAOA circuits used to obtain these correlations.

Furthermore it would be interesting to use other means of calculating quantum correlations going beyond QAOA. For example, much research has been conducted on analog quantum devices for quantum optimization [84, 85]. Using such devices to obtain correlations for shrinking algorithms seems like a promising path to follow in quantum optimization research [12]. Comparing the performance of RQO algorithms

using gate-based and analog quantum devices for obtaining correlations would allow further insights into the potential of quantum optimization routines [23].

Throughout this work all experiments involving quantum algorithms were done by classical simulations of actual quantum hardware. Thus, it would be of utter interest to develop and run similar explorations on real quantum devices. This would on one hand allow testing the simulations of the thesis against results on currently available devices. On the other hand, such experiments probe the influence of real world drawbacks such as quantum noise and hardware faults on RQO routines.

Appendix A

Ising formulation of the SETCOVER problem

The Ising formulation of the SETCOVER problem can be derived from Eq. (2.10) by using the mapping $x_i = \frac{1}{2}(1 + z_i)$:

$$\begin{aligned}
\min_{\mathbf{z}} C_{\text{SETCOVER}}(\mathbf{z}) = & An(1 - M + \frac{1}{4}M^2) + A \sum_{\alpha=1}^n \left[\sum_{m=1}^M z_{\alpha,m} \left(\frac{M}{2} - 1 \right) \right. \\
& + \frac{1}{4} \left(\sum_{m=1}^M z_{\alpha,m} \right)^2 + \frac{1}{4} \left(\sum_{m=1}^M m z_{\alpha,m} \right)^2 + \frac{1}{4} M(M+1) \sum_{m=1}^M m z_{\alpha,m} \\
& - \frac{1}{2} \left(\sum_{m=1}^M m z_{\alpha,m} \right) \left(\sum_{i:\alpha \in V_i} (z_i + 1) \right) + \frac{1}{16} M^2 (M+1)^2 \\
& \left. - \frac{1}{4} M(M+1) \sum_{i:\alpha \in V_i} (z_i + 1) + \frac{1}{4} \left(\sum_{i:\alpha \in V_i} (z_i + 1) \right)^2 \right] \\
& + \frac{1}{2} B(N + \sum_{i=1}^N z_i).
\end{aligned} \tag{A.1}$$

Appendix B

QAOA parameter analysis for MAXCUT
problems of sizes $n \in \{100, 150, 200\}$

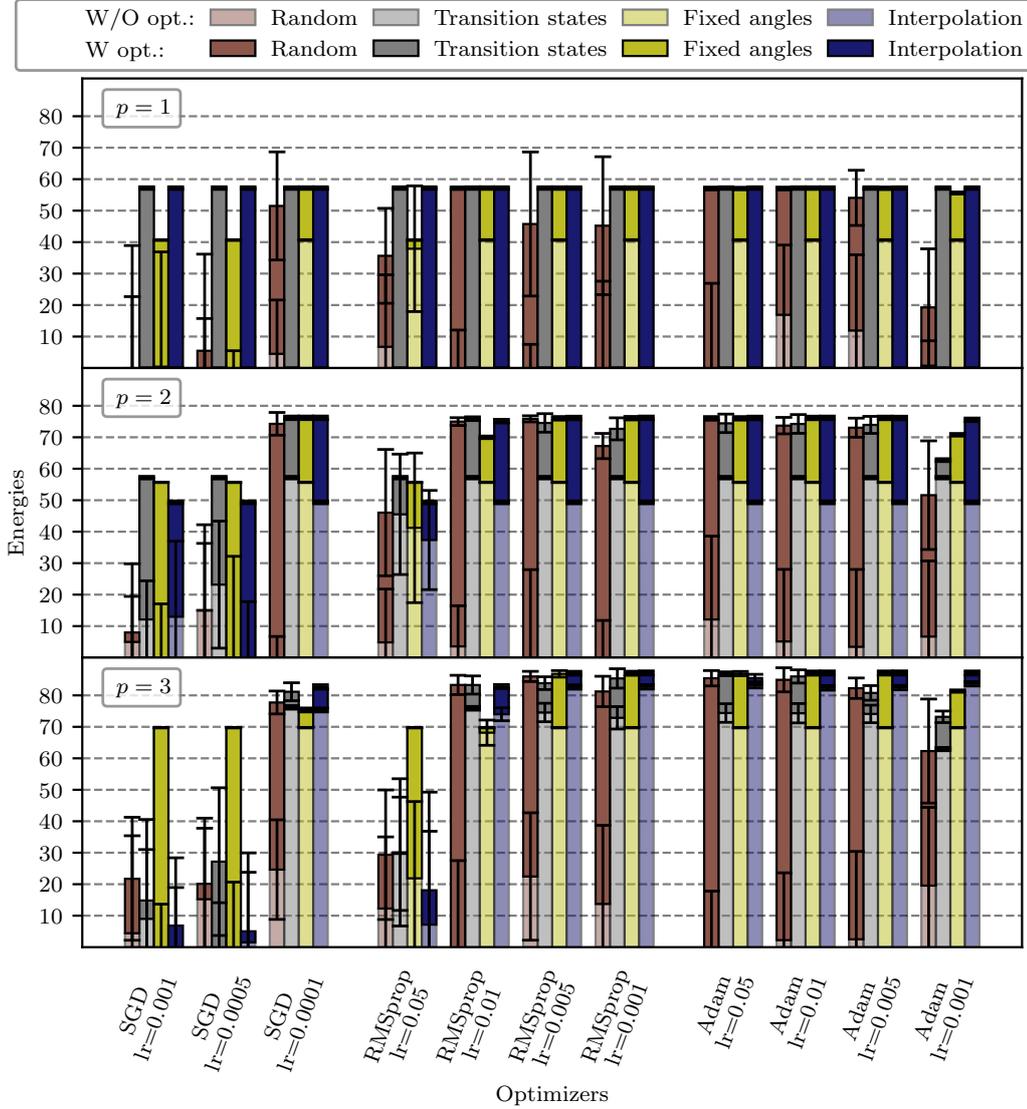


Figure B.1: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MAXCUT problems with $n = 100$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

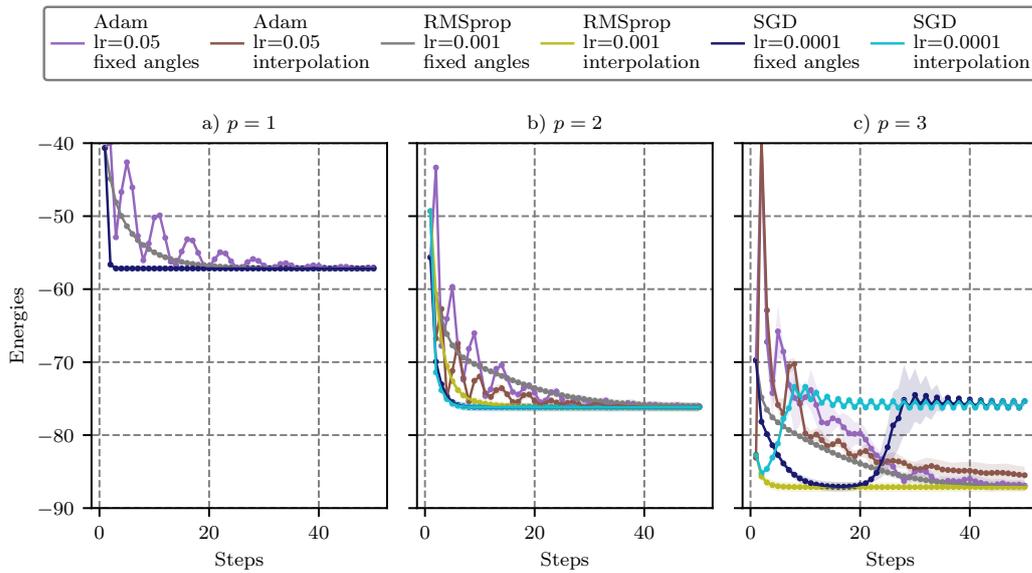


Figure B.2: Loss curves during QAOA parameter optimization of the most promising combinations of initialization, optimizer and learning rate (lr) are shown. The losses correspond to the energy results of Fig. B.1. Thus, mean and standard deviation of the losses during optimization of 20 different 3-regular MAXCUT graphs with $n = 100$ variables are pictured.

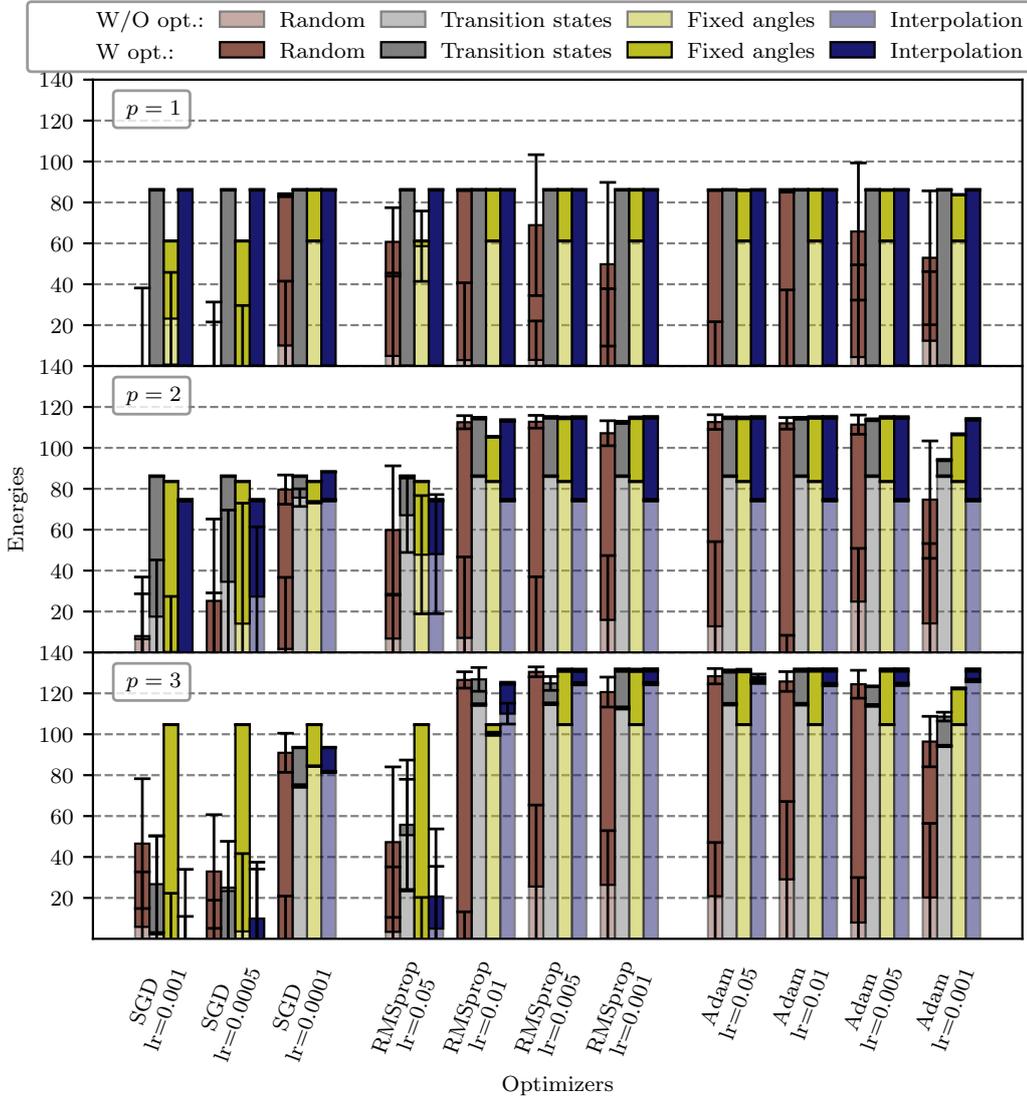


Figure B.3: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MAXCUT problems with $n = 150$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

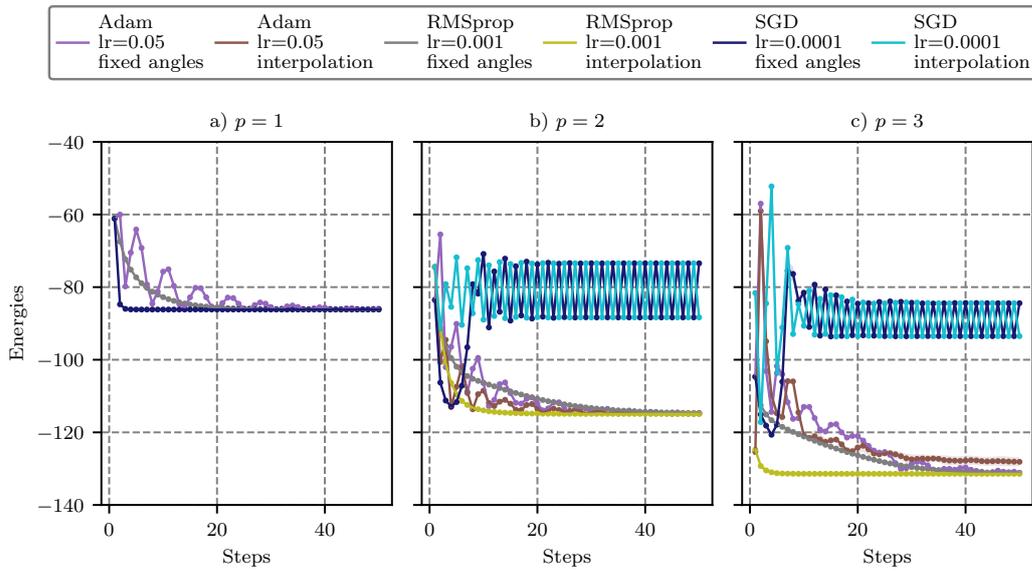


Figure B.4: Loss curves during QAOA parameter optimization of the most promising combinations of initialization, optimizer and learning rate (lr) are shown. The losses correspond to the energy results of Fig. B.3. Thus, mean and standard deviation of the losses during optimization of 20 different 3-regular MAXCUT graphs with $n = 150$ variables are pictured.

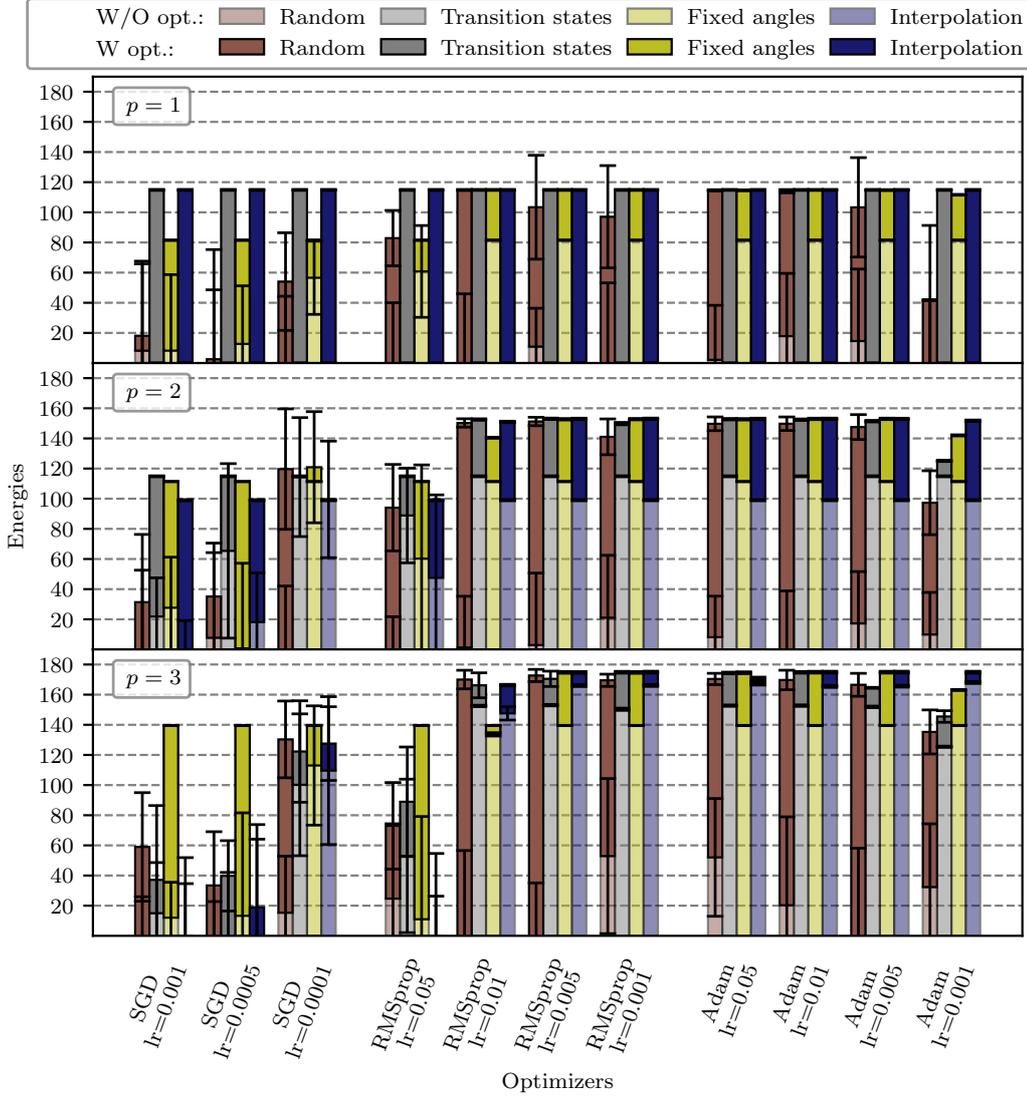


Figure B.5: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MAXCUT problems with $n = 200$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

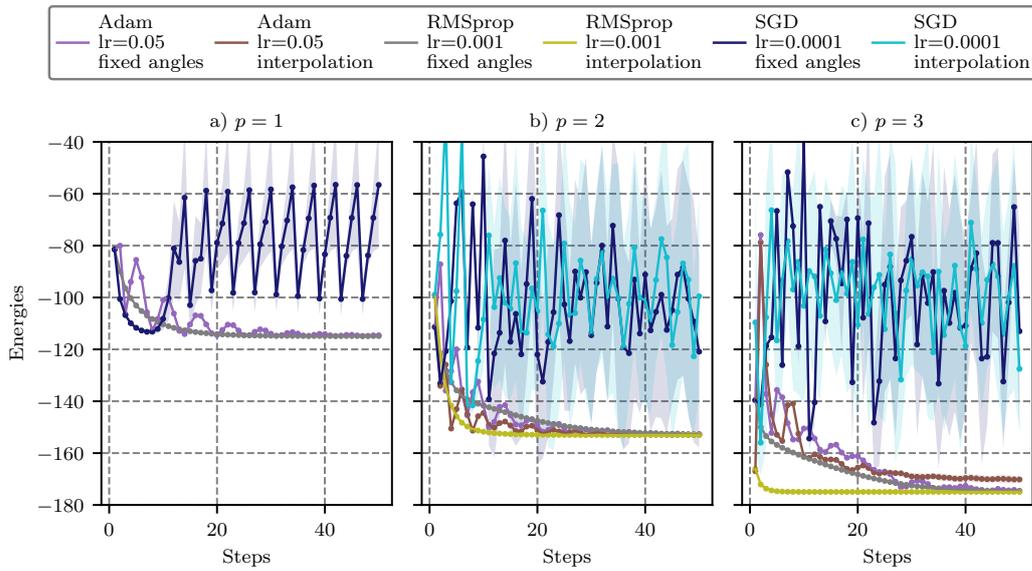


Figure B.6: Loss curves during QAOA parameter optimization of the most promising combinations of initialization, optimizer and learning rate (lr) are shown. The losses correspond to the energy results of Fig. B.5. Thus, mean and standard deviation of the losses during optimization of 20 different 3-regular MAXCUT graphs with $n = 200$ variables are pictured.

Appendix C

QAOA parameter analysis for MIS problems of sizes $n \in \{100, 150, 200\}$

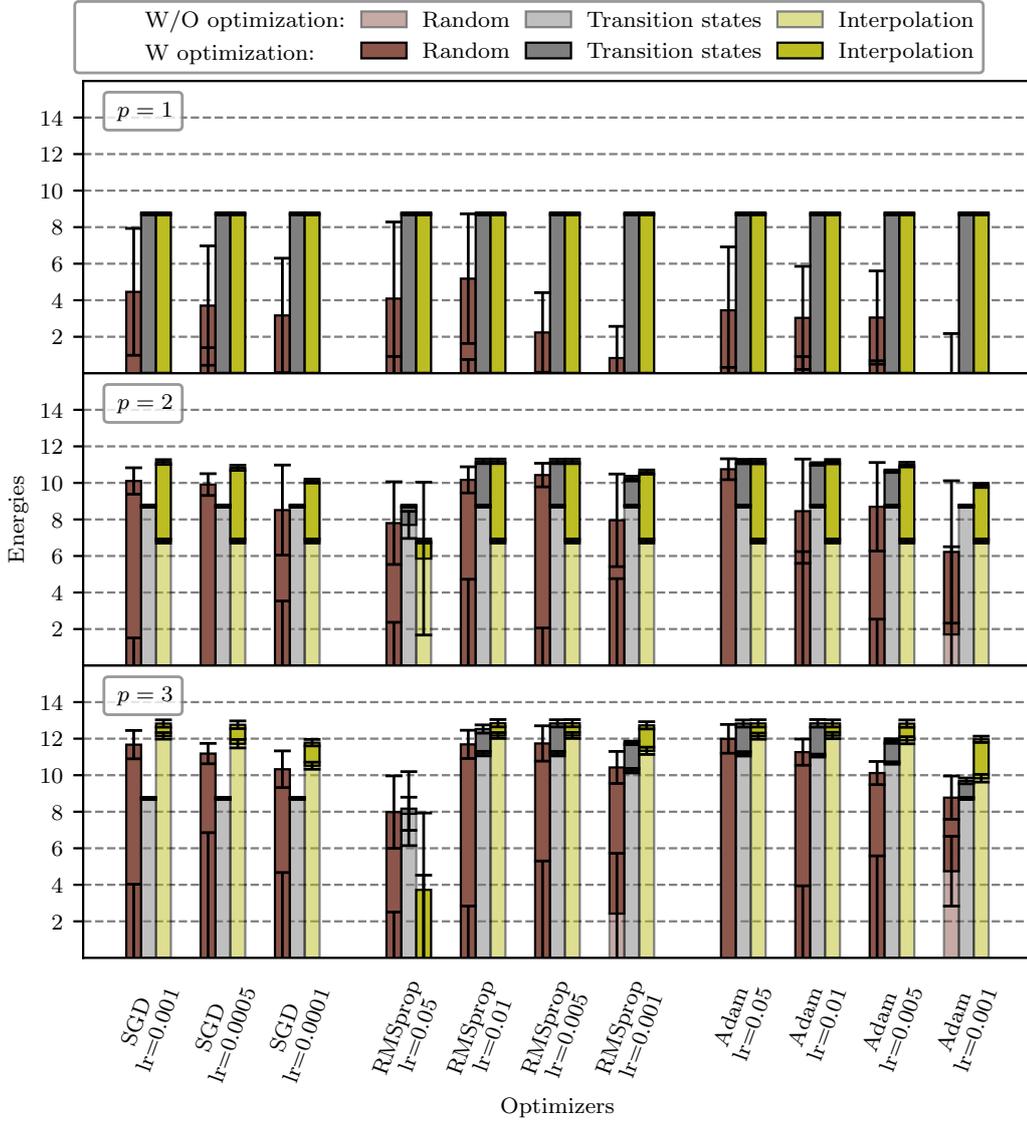


Figure C.1: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MIS problems with $n = 50$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

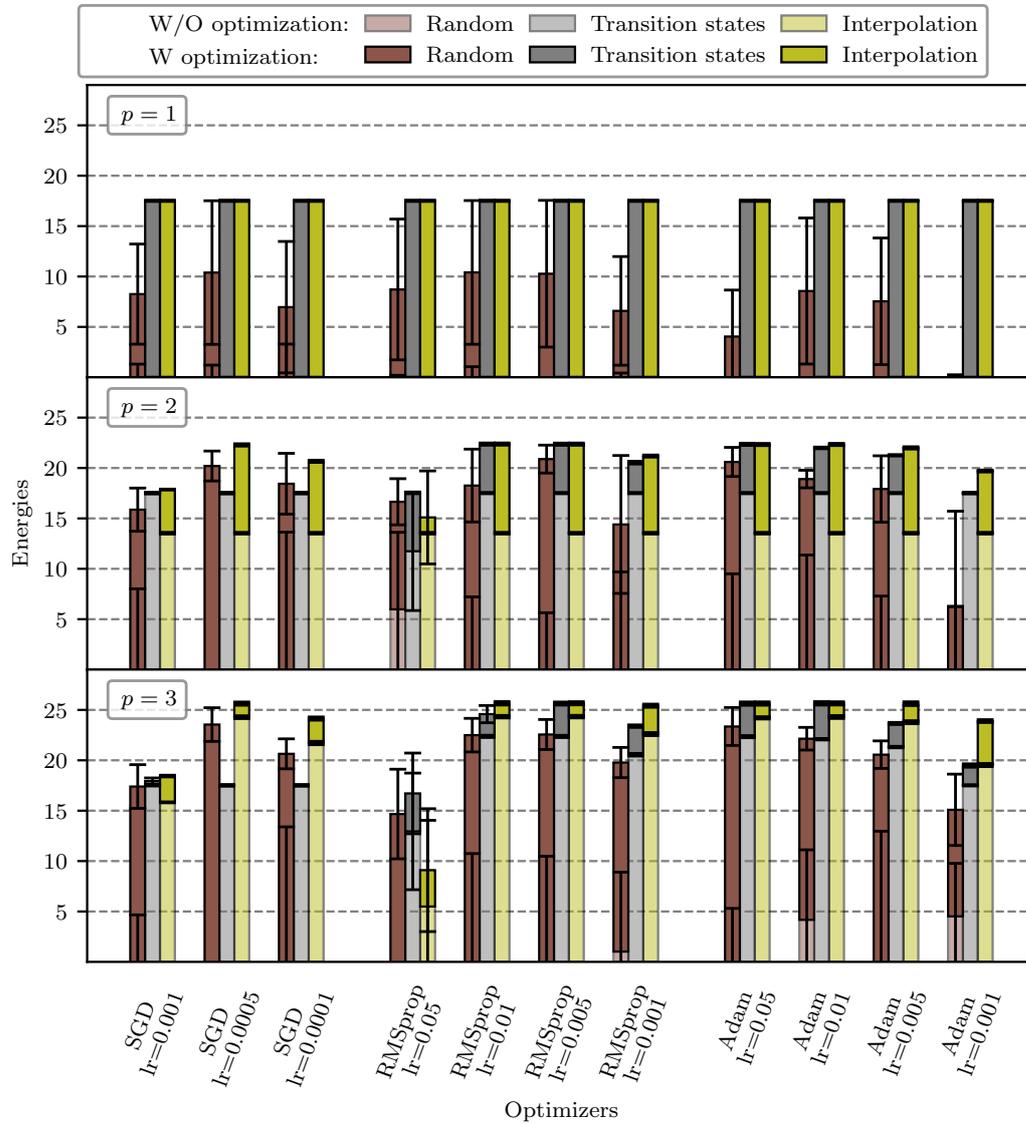


Figure C.2: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MIS problems with $n = 100$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

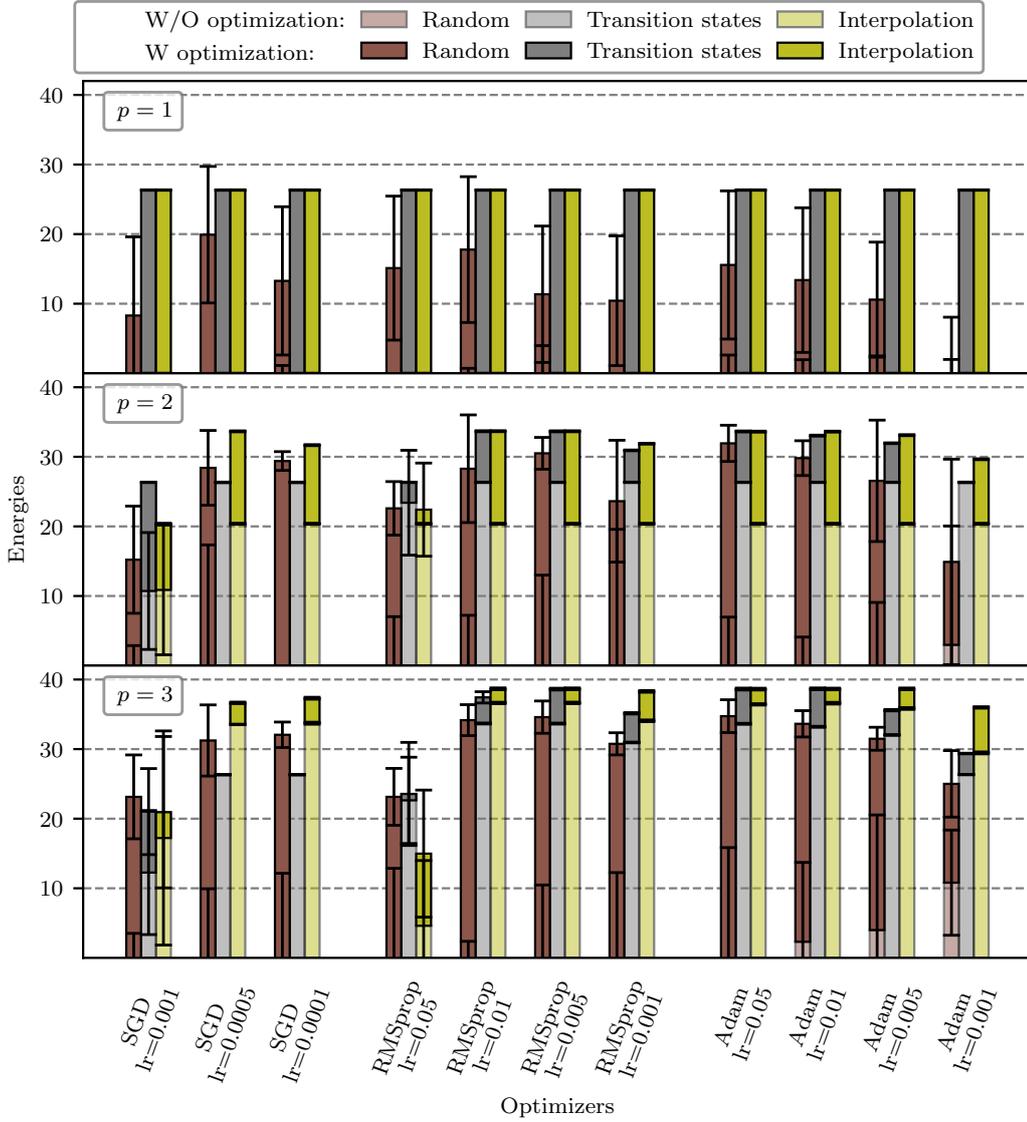


Figure C.3: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MIS problems with $n = 150$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

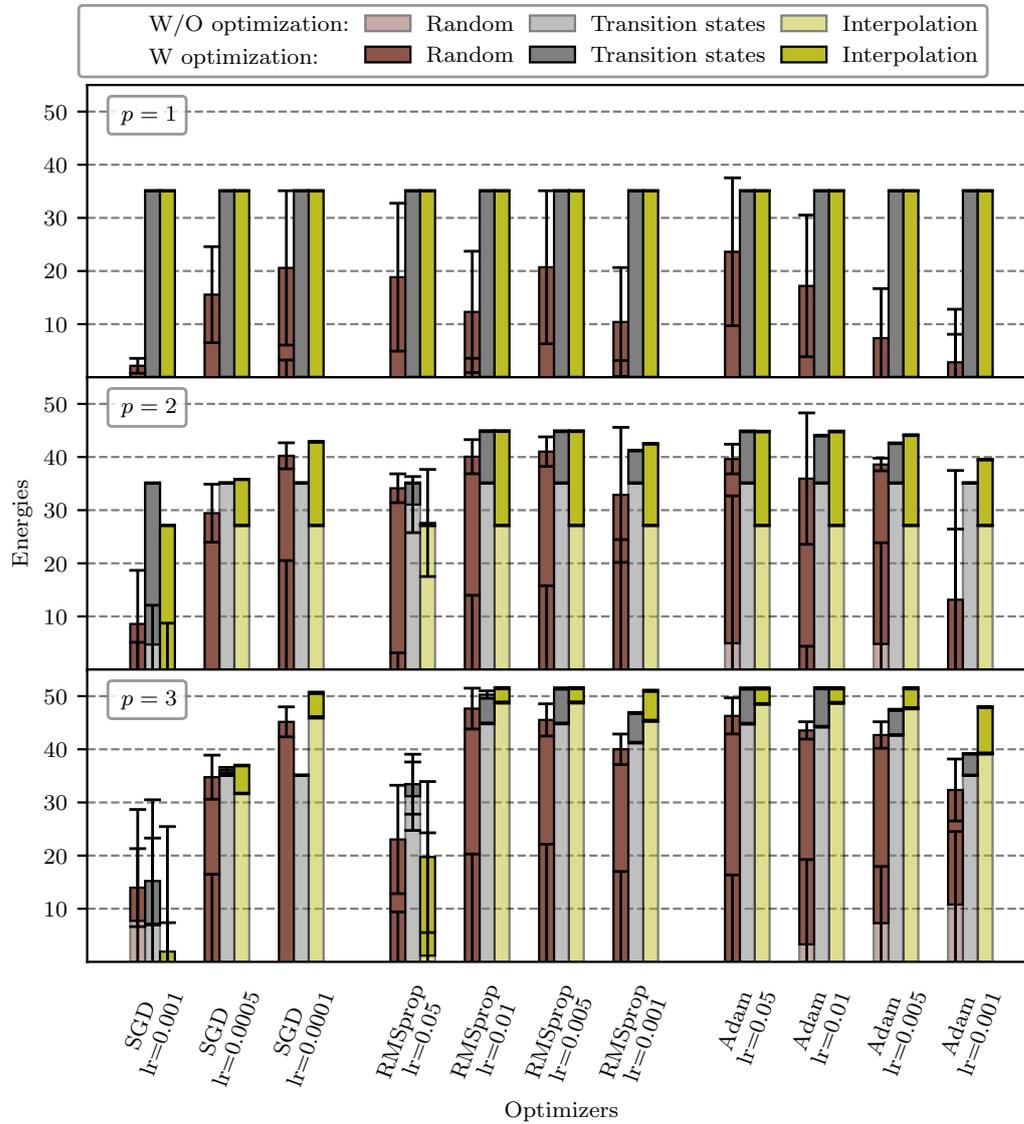


Figure C.4: Results of the analysis on different initialization methods and classical optimizers for finding low energy QAOA states are shown. For each combination of initialization, optimizer and corresponding learning rate (lr) the full QAOA routine was applied to 20 different random 3-regular MIS problems with $n = 200$ variables. The bars represent the negative mean value of the obtained energies and error bars the standard deviation. Light colored bars show the energy without optimization routine and full colored after optimization.

Bibliography

- [1] Gang Yu. *Industrial applications of combinatorial optimization*. Vol. 16. Springer Science & Business Media, 2013.
- [2] Prabhakar Raghavan and Clark D. Tompson. »Randomized rounding: a technique for provably good algorithms and algorithmic proofs«. In: *Combinatorica* 7.4 (Dec. 1987), pp. 365–374. ISSN: 0209-9683. DOI: 10.1007/BF02579324. URL: <https://doi.org/10.1007/BF02579324>.
- [3] Michel X Goemans and David P Williamson. »Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming«. In: *Journal of the ACM (JACM)* 42.6 (1995), pp. 1115–1145.
- [4] Edward Farhi, Jeffrey Goldstone and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph].
- [5] Lennart Bittel and Martin Kliesch. »Training Variational Quantum Algorithms Is NP-Hard«. In: *Phys. Rev. Lett.* 127 (12 Sept. 2021), p. 120502. DOI: 10.1103/PhysRevLett.127.120502. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.127.120502>.
- [6] Daniel Stilck França and Raul García-Patrón. »Limitations of optimization algorithms on noisy quantum devices«. en. In: *Nature Physics* (Oct. 2021). ISSN: 1745-2473, 1745-2481. DOI: 10.1038/s41567-021-01356-3. URL: <https://www.nature.com/articles/s41567-021-01356-3> (visited on 22/10/2021).
- [7] Sergey Bravyi et al. »Obstacles to Variational Quantum Optimization from Symmetry Protection«. In: *Phys. Rev. Lett.* 125 (26 Dec. 2020), p. 260505. DOI: 10.1103/PhysRevLett.125.260505. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.125.260505>.
- [8] Edward Farhi, David Gamarnik and Sam Gutmann. »The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: Worst Case Examples«. In: (2020). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2005.08747. URL: <https://arxiv.org/abs/2005.08747> (visited on 04/04/2023).

- [9] Edward Farhi, David Gamarnik and Sam Gutmann. »The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: A Typical Case«. In: (2020). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2004.09002. URL: <https://arxiv.org/abs/2004.09002> (visited on 04/04/2023).
- [10] Chi-Ning Chou et al. *Limitations of Local Quantum Algorithms on Random Max-k-XOR and Beyond*. 2022. arXiv: 2108.06049 [quant-ph].
- [11] Sergey Bravyi et al. »Hybrid quantum-classical algorithms for approximate graph coloring«. In: *Quantum* 6 (Mar. 2022), p. 678. ISSN: 2521-327X. DOI: 10.22331/q-2022-03-30-678. URL: <https://doi.org/10.22331/q-2022-03-30-678>.
- [12] Jernej Rudi Fin žgar et al. »Quantum-Informed Recursive Optimization Algorithms«. In: *PRX Quantum* 5 (2 May 2024), p. 020327. DOI: 10.1103/PRXQuantum.5.020327. URL: <https://link.aps.org/doi/10.1103/PRXQuantum.5.020327>.
- [13] Friedrich Wagner, Jonas Nüßlein and Frauke Liers. *Enhancing Quantum Algorithms for Quadratic Unconstrained Binary Optimization via Integer Programming*. arXiv:2302.05493 [quant-ph]. May 2023. URL: <http://arxiv.org/abs/2302.05493> (visited on 12/02/2024).
- [14] Eunok Bae and Soojoon Lee. »Recursive QAOA outperforms the original QAOA for the MAX-CUT problem on complete graphs«. In: *Quantum Information Processing* 23.3 (Feb. 2024). ISSN: 1573-1332. DOI: 10.1007/s11128-024-04286-0. URL: <http://dx.doi.org/10.1007/s11128-024-04286-0>.
- [15] Sergey Bravyi et al. *Classical algorithms for Forrelation*. 2021. arXiv: 2102.06963 [quant-ph].
- [16] Lucas T. Brady and Stuart Hadfield. *Iterative Quantum Algorithms for Maximum Independent Set: A Tale of Low-Depth Quantum Algorithms*. 2023. arXiv: 2309.13110 [quant-ph].
- [17] Edd Gent. *Quantum Computing's Hard, Cold Reality Check*. 2024. URL: <https://spectrum.ieee.org/quantum-computing-skeptics> (visited on 03/07/2024).
- [18] Gian Giacomo Guerreschi. »Solving Quadratic Unconstrained Binary Optimization with divide-and-conquer and quantum algorithms«. In: *ArXiv abs/2101.07813* (2021). URL: <https://api.semanticscholar.org/CorpusID:231648106>.
- [19] Gary Kochenberger et al. »The unconstrained binary quadratic programming problem: A survey«. In: *Journal of Combinatorial Optimization* 28 (July 2014). DOI: 10.1007/s10878-014-9734-0.

- [20] Fred Glover, Gary Kochenberger and Yu Du. *A Tutorial on Formulating and Using QUBO Models*. 2019. arXiv: 1811.11538 [cs.DS].
- [21] Rodney J. Baxter. »Exactly solved models in statistical mechanics«. In: 1982. URL: <https://api.semanticscholar.org/CorpusID:117867044>.
- [22] Iain Dunning, Swati Gupta and John Silberholz. »What Works Best When? A Systematic Evaluation of Heuristics for Max-Cut and QUBO«. In: *INFORMS Journal on Computing* 30.3 (2018), pp. 608–624. DOI: 10.1287/ijoc.2017.0798. URL: <https://doi.org/10.1287/ijoc.2017.0798>.
- [23] Victor Fischer et al. *The role of quantum and classical correlations in shrinking algorithms for optimization*. 2024. arXiv: 2404.17242 [quant-ph].
- [24] F. Barahona, M. Jünger and G. Reinelt. »Experiments in quadratic 0–1 programming«. In: *Mathematical Programming* 44.1–3 (May 1989), pp. 127–137. ISSN: 1436-4646. DOI: 10.1007/bf01587084. URL: <http://dx.doi.org/10.1007/BF01587084>.
- [25] Caterina De Simone. »The cut polytope and the Boolean quadric polytope«. In: *Discrete Mathematics* 79.1 (1990), pp. 71–75. ISSN: 0012-365X. DOI: [https://doi.org/10.1016/0012-365X\(90\)90056-N](https://doi.org/10.1016/0012-365X(90)90056-N). URL: <https://www.sciencedirect.com/science/article/pii/0012365X9090056N>.
- [26] Peter L. Ivănescu. »Some Network Flow Problems Solved with Pseudo-Boolean Programming«. In: *Operations Research* 13.3 (1965), pp. 388–399. ISSN: 0030364X, 15265463. URL: <http://www.jstor.org/stable/167803> (visited on 28/02/2024).
- [27] Michael Jünger and Sven Mallach. »Exact Facetial Odd-Cycle Separation for Maximum Cut and Binary Quadratic Optimization«. In: *INFORMS Journal on Computing* (Feb. 2021). DOI: 10.1287/ijoc.2020.1008.
- [28] Jonas Charfreitag et al. »McSparse: Exact Solutions of Sparse Maximum Cut and Sparse Unconstrained Binary Quadratic Optimization Problems«. In: *2022 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pp. 54–66. DOI: 10.1137/1.9781611977042.5. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611977042.5>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611977042.5>.
- [29] Richard M. Karp. »Reducibility among Combinatorial Problems«. In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*.

- Ed. by Raymond E. Miller, James W. Thatcher and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9. URL: https://doi.org/10.1007/978-1-4684-2001-2_9.
- [30] Francisco Barahona et al. »An application of combinatorial optimization to statistical physics and circuit layout design«. In: *Operations Research* 36.3 (1988), pp. 493–513.
- [31] Francisco Barahona, Michael Jünger and Gerhard Reinelt. »Experiments in quadratic 0–1 programming«. In: *Mathematical Programming* 44.1-3 (1989), pp. 127–137.
- [32] Michael Jünger and Sven Mallach. »Odd-cycle separation for maximum cut and binary quadratic optimization«. In: *27th Annual European Symposium on Algorithms (ESA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019.
- [33] Daniel Rehfeldt, Thorsten Koch and Yuji Shinano. »Faster exact solution of sparse MaxCut and QUBO problems«. In: *Mathematical Programming Computation* (2023), pp. 1–26.
- [34] Martin S Andersen, Joachim Dahl, Lieven Vandenberghé et al. »CVXOPT: A Python package for convex optimization«. In: *Available at cvxopt.org* 54 (2013).
- [35] Makoto Yamashita, Katsuki Fujisawa and Masakazu Kojima. »Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0)«. In: *Optimization Methods and Software* 18.4 (2003), pp. 491–505.
- [36] Miguel F Anjos and Jean B Lasserre. *Handbook on semidefinite, conic and polynomial optimization*. Vol. 166. Springer Science & Business Media, 2011.
- [37] Jonathan Wurtz et al. *Industry applications of neutral-atom quantum computing solving independent set problems*. 2024. arXiv: 2205.08500 [quant-ph].
- [38] Brent N. Clark, Charles J. Colbourn and David S. Johnson. »Unit disk graphs«. English (US). In: *Discrete Mathematics* 86.1-3 (Dec. 1990). Funding Information: of Toronto for hospitality during the time this paper was written. The research of the second author is supported by NSERC Canada under grant A0579., pp. 165–177. ISSN: 0012-365X. DOI: 10.1016/0012-365X(90)90358-0.
- [39] Erik Jan van Leeuwen. »Approximation Algorithms for Unit Disk Graphs«. In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Dieter Kratsch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 351–361. ISBN: 978-3-540-31468-4.

- [40] Martin J. A. Schuetz, J. Kyle Brubaker and Helmut G. Katzgraber. »Combinatorial optimization with physics-inspired graph neural networks«. In: *Nature Machine Intelligence* 4.4 (Apr. 2022), pp. 367–377. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00468-6. URL: <http://dx.doi.org/10.1038/s42256-022-00468-6>.
- [41] Maria Chiara Angelini and Federico Ricci-Tersenghi. »Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set«. In: *Nature Machine Intelligence* 5.1 (Dec. 2022), pp. 29–31. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00589-y. URL: <http://dx.doi.org/10.1038/s42256-022-00589-y>.
- [42] Martin J. A. Schuetz, John Kyle Brubaker and Helmut G. Katzgraber. »Reply to: Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set«. In: *Nat. Mac. Intell.* 5.1 (2023), pp. 32–34. DOI: 10.1038/S42256-022-00590-5. URL: <https://doi.org/10.1038/s42256-022-00590-5>.
- [43] Magnús Halldórsson and Jaikumar Radhakrishnan. »Greed is good: approximating independent sets in sparse and bounded-degree graphs«. In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*. STOC '94. Montreal, Quebec, Canada: Association for Computing Machinery, 1994, pp. 439–448. ISBN: 0897916638. DOI: 10.1145/195058.195221. URL: <https://doi.org/10.1145/195058.195221>.
- [44] Andrew Lucas. »Ising formulations of many NP problems«. In: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005. URL: <http://dx.doi.org/10.3389/fphy.2014.00005>.
- [45] Jacob C Bridgeman and Christopher T Chubb. »Hand-waving and interpretive dance: an introductory course on tensor networks«. In: *Journal of Physics A: Mathematical and Theoretical* 50.22 (May 2017), p. 223001. ISSN: 1751-8121. DOI: 10.1088/1751-8121/aa6dc3. URL: <http://dx.doi.org/10.1088/1751-8121/aa6dc3>.
- [46] Román Orús. »Tensor networks for complex quantum systems«. In: *Nature Reviews Physics* 1.9 (Aug. 2019), pp. 538–550. ISSN: 2522-5820. DOI: 10.1038/s42254-019-0086-7. URL: <http://dx.doi.org/10.1038/s42254-019-0086-7>.
- [47] Ulrich Schollwöck. »The density-matrix renormalization group in the age of matrix product states«. In: *Annals of Physics* 326.1 (Jan. 2011), pp. 96–192. ISSN: 0003-4916. DOI: 10.1016/j.aop.2010.09.012. URL: <http://dx.doi.org/10.1016/j.aop.2010.09.012>.

- [48] F. Verstraete and J. I. Cirac. *Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions*. 2004. arXiv: cond-mat/0407066 [cond-mat.str-el].
- [49] Y.-Y. Shi, L.-M. Duan and G. Vidal. »Classical simulation of quantum many-body systems with a tree tensor network«. In: *Physical Review A* 74.2 (Aug. 2006). ISSN: 1094-1622. DOI: 10.1103/physreva.74.022320. URL: <http://dx.doi.org/10.1103/PhysRevA.74.022320>.
- [50] G. Vidal. »Entanglement Renormalization«. In: *Physical Review Letters* 99.22 (Nov. 2007). ISSN: 1079-7114. DOI: 10.1103/physrevlett.99.220405. URL: <http://dx.doi.org/10.1103/PhysRevLett.99.220405>.
- [51] Igor L. Markov and Yaoyun Shi. »Simulating Quantum Computation by Contracting Tensor Networks«. In: *SIAM Journal on Computing* 38.3 (Jan. 2008), pp. 963–981. ISSN: 1095-7111. DOI: 10.1137/050644756. URL: <http://dx.doi.org/10.1137/050644756>.
- [52] Guifré Vidal. »Efficient Classical Simulation of Slightly Entangled Quantum Computations«. In: *Physical Review Letters* 91.14 (Oct. 2003). ISSN: 1079-7114. DOI: 10.1103/physrevlett.91.147902. URL: <http://dx.doi.org/10.1103/PhysRevLett.91.147902>.
- [53] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [54] Jaeho Choi and Joongheon Kim. »A Tutorial on Quantum Approximate Optimization Algorithm (QAOA): Fundamentals and Applications«. In: *2019 International Conference on Information and Communication Technology Convergence (ICTC) (2019)*, pp. 138–142. URL: <https://api.semanticscholar.org/CorpusID:209497481>.
- [55] Leo Zhou et al. »Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices«. In: *Phys. Rev. X* 10 (2 June 2020), p. 021067. DOI: 10.1103/PhysRevX.10.021067. URL: <https://link.aps.org/doi/10.1103/PhysRevX.10.021067>.
- [56] Michael Streif et al. »Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm«. In: *Phys. Rev. A* 104 (1 July 2021), p. 012403. DOI: 10.1103/PhysRevA.104.012403. URL: <https://link.aps.org/doi/10.1103/PhysRevA.104.012403>.
- [57] Ajinkya Borle, Vincent E. Elfving and Samuel J. Lomonaco. »Quantum approximate optimization for hard problems in linear algebra«. In: *SciPost Phys. Core* 4 (2021), p. 031. DOI: 10.21468/SciPostPhysCore.4.4.031. URL: <https://scipost.org/10.21468/SciPostPhysCore.4.4.031>.

- [58] Abhishek Awasthi et al. »Quantum Computing Techniques for Multi-knapsack Problems«. In: *Intelligent Computing*. Springer Nature Switzerland, 2023, pp. 264–284. ISBN: 9783031379635. DOI: 10.1007/978-3-031-37963-5_19. URL: http://dx.doi.org/10.1007/978-3-031-37963-5_19.
- [59] Mark Hodson et al. *Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz*. 2019. arXiv: 1911.05296 [quant-ph].
- [60] Jack S. Baker and Santosh Kumar Radha. *Wasserstein Solution Quality and the Quantum Approximate Optimization Algorithm: A Portfolio Optimization Case Study*. 2022. arXiv: 2202.06782 [quant-ph].
- [61] Junde Li et al. »Hierarchical Improvement of Quantum Approximate Optimization Algorithm for Object Detection: (Invited Paper)«. In: *2020 21st International Symposium on Quality Electronic Design (ISQED)*. 2020, pp. 335–340. DOI: 10.1109/ISQED48828.2020.9136973.
- [62] Pradeep Niroula et al. »Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer«. In: *Scientific Reports* 12.1 (Oct. 2022). ISSN: 2045-2322. DOI: 10.1038/s41598-022-20853-w. URL: <http://dx.doi.org/10.1038/s41598-022-20853-w>.
- [63] Hasan Mustafa et al. *Variational Quantum Algorithms for Chemical Simulation and Drug Discovery*. 2022. arXiv: 2211.07854 [quant-ph].
- [64] Jaeho Choi, Seunghyeok Oh and Joongheon Kim. *Quantum Approximation for Wireless Scheduling*. 2020. arXiv: 2004.11229 [cs.OH].
- [65] Asier Ozaeta, Wim Van Dam and Peter L McMahon. »Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems«. In: *Quantum Science and Technology* 7.4 (Oct. 2022), p. 045036. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac9013. URL: <https://iopscience.iop.org/article/10.1088/2058-9565/ac9013> (visited on 12/02/2024).
- [66] Daniel Stilck França and Raul García-Patrón. »Limitations of optimization algorithms on noisy quantum devices«. In: *Nature Physics* 17.11 (Oct. 2021), pp. 1221–1227. ISSN: 1745-2481. DOI: 10.1038/s41567-021-01356-3. URL: <http://dx.doi.org/10.1038/s41567-021-01356-3>.
- [67] Danylo Lykov et al. *Tensor Network Quantum Simulator With Step-Dependent Parallelization*. 2022. arXiv: 2012.02430 [quant-ph].
- [68] Danylo Lykov and Yuri Alexeev. *Importance of Diagonal Gates in Tensor Network Simulations*. 2021. arXiv: 2106.15740 [quant-ph].
- [69] Cameron Ibrahim et al. »Constructing Optimal Contraction Trees for Tensor Network Quantum Circuit Simulation«. In: *2022 IEEE High Performance Extreme Computing Conference (HPEC)*. 2022, pp. 1–8. DOI: 10.1109/HPEC55821.2022.9926353.

- [70] Sergio Boixo et al. *Simulation of low-depth quantum circuits as complex undirected graphical models*. 2018. arXiv: 1712.05384 [quant-ph].
- [71] R. Dechter. »Bucket Elimination: A Unifying Framework for Probabilistic Inference«. In: *Learning in Graphical Models*. Ed. by Michael I. Jordan. Dordrecht: Springer Netherlands, 1998, pp. 75–104. ISBN: 978-94-011-5014-9. DOI: 10.1007/978-94-011-5014-9_4. URL: https://doi.org/10.1007/978-94-011-5014-9_4.
- [72] Roman Schutski, Danil Lykov and Ivan Oseledets. »Adaptive algorithm for quantum circuit simulation«. In: *Phys. Rev. A* 101 (4 Apr. 2020), p. 042335. DOI: 10.1103/PhysRevA.101.042335. URL: <https://link.aps.org/doi/10.1103/PhysRevA.101.042335>.
- [73] Wikipedia contributors. *Tree decomposition — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-July-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Tree_decomposition&oldid=1217459734.
- [74] Hisao Tamaki. *Positive-instance driven dynamic programming for treewidth*. 2018. arXiv: 1704.05286 [cs.DS]. URL: <https://arxiv.org/abs/1704.05286>.
- [75] Vibhav Gogate and Rina Dechter. *A Complete Anytime Algorithm for Treewidth*. 2012. arXiv: 1207.4109 [cs.DS]. URL: <https://arxiv.org/abs/1207.4109>.
- [76] David Wierichs, Christian Gogolin and Michael Kastoryano. »Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer«. In: *Physical Review Research* 2.4 (Nov. 2020). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.2.043246. URL: <http://dx.doi.org/10.1103/PhysRevResearch.2.043246>.
- [77] Stefan H. Sack and Maksym Serbyn. »Quantum annealing initialization of the quantum approximate optimization algorithm«. In: *Quantum* 5 (July 2021), p. 491. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-01-491. URL: <http://dx.doi.org/10.22331/q-2021-07-01-491>.
- [78] Stefan H. Sack et al. »Recursive greedy initialization of the quantum approximate optimization algorithm with guaranteed improvement«. In: *Phys. Rev. A* 107 (6 June 2023), p. 062404. DOI: 10.1103/PhysRevA.107.062404. URL: <https://link.aps.org/doi/10.1103/PhysRevA.107.062404>.
- [79] Jonathan Wurtz and Danylo Lykov. *The fixed angle conjecture for QAOA on regular MaxCut graphs*. 2021. arXiv: 2107.00677.

- [80] Jonathan Wurtz and Peter Love. »MaxCut quantum approximate optimization algorithm performance guarantees for *pqt;1*«. In: *Phys. Rev. A* 103 (4 Apr. 2021), p. 042612. DOI: 10.1103/PhysRevA.103.042612. URL: <https://link.aps.org/doi/10.1103/PhysRevA.103.042612>.
- [81] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747.
- [82] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. URL: <https://www.gurobi.com>.
- [83] Joao Basso et al. »The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model«. en. In: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. DOI: 10.4230/LIPICS.TQC.2022.7. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPICS.TQC.2022.7>.
- [84] Tadashi Kadowaki and Hidetoshi Nishimori. »Quantum annealing in the transverse Ising model«. en. In: *Physical Review E* 58.5 (Nov. 1998), pp. 5355–5363. ISSN: 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.58.5355. URL: <https://link.aps.org/doi/10.1103/PhysRevE.58.5355> (visited on 23/06/2023).
- [85] Edward Farhi et al. »A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem«. en. In: *Science* 292.5516 (Apr. 2001), pp. 472–475. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1057726. URL: <https://www.science.org/doi/10.1126/science.1057726> (visited on 25/05/2023).