# Anwendung und Optimierung eingeschränkter Hamiltonians

Christoph Roch



München 2024

# Anwendung und Optimierung eingeschränkter Hamiltonians

Christoph Roch

Dissertation an der Fakultät für Mathematik, Informatik und Statistik der Ludwig–Maximilians–Universität München

> vorgelegt von Christoph Roch

Tag der Einreichung: 24.10.2023

1. Berichterstatter/in:Prof. Dr. Claudia Linnhoff-Popien2. Berichterstatter/in:Prof. Dr. Dr. h. c. Frank LeymannTag der Einreichung:24.10.2023Tag der Disputation:22.05.2024

## Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, 24.10.2023 Christoph Roch

# Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei einigen Personen bedanken, die mich in meiner Zeit am Lehrstuhl für Mobile und Verteilte Systeme an der Ludwig-Maximilians-Universität München begleitet haben.

Zunächst möchte ich Frau Prof. Dr. Claudia Linnhoff-Popien einen ganz besonderen Dank aussprechen, für die stets vertrauensvolle Zusammenarbeit und gegenseitige Wertschätzung über viele Jahre hinweg. Ich konnte durch die Möglichkeiten und Chancen, die sie mir eröffnet hat meinen Forschungsinteressen folgen und stets neue Ziele erreichen. Ihr Lehrstuhl bot mir dabei ein außergewöhnlich inspirierendes, wertschätzendes Umfeld. Herzlichen Dank dafür, dass ich ein Teil dieses wunderbaren Teams sein konnte.

Des Weiteren gilt mein herzlicher Dank Herrn Prof. Dr. Dr. h. c. Frank Leymann, der sich freundlicherweise bereit erklärte, die Rolle des Zweitberichterstatters zu übernehmen. Auch bin ich für die jahrelange wertschätzende Zusammenarbeit im BMWK geförderten Projekt PlanQK sehr dankbar.

Ebenso danke ich Herrn Prof. Dr. Matthias Schubert für die Übernahme des Vorsitzes der Prüfungskommission, sowie Herrn Prof. Dr. Albrecht Schmidt für seine Verfügbarkeit als Ersatzprüfer.

Ein großes Dankeschön geht an alle meine Kollegen und Freunde am Lehrstuhl. Ein Team wie dieses ist etwas Besonderes und wird mir immer in Erinnerung bleiben. Namentlich erwähnen möchte ich Dr. Sebastian Feld, der mich an den Lehrstuhl herangeführt hat und als Mentor zu Beginn eine besondere Unterstützung war, egal ob beim Schreiben gemeinsamer Paper, oder bei langen Gesprächen zur Situation am Lehrstuhl.

Danken möchte ich auch meiner Familie, meinen Freunden und meiner Freundin für die jahrelange Unterstützung und die so entscheidend wertvolle Grundgelassenheit und Überzeugtheit, dass ich erreichen kann, was ich mir wünsche.

# Zusammenfassung

Quantum Computing Hardware ist noch immer begrenzt in der Anzahl der Qubits, deren Kohärenzeigenschaften und deren Konnektivität auf den Quantenprozessoren. Um daher bereits frühzeitig einen möglichen praktischen Quantenvorteil zum klassischen Computing zu erzielen, gilt es deshalb, sowohl relevante Anwendungsfälle mit großem Potenzial, als auch entsprechend umsetzbare Algorithmen für die aktuell verfügbare Noisy Intermediate-Scale Quantum (NISQ) Computer zu identifizieren. Viele relevante Problemstellungen der Industrie und Wissenschaft lassen sich auf Optimierungs- und Suchprobleme abbilden. Die Komplexität aus Anwendersicht ist die Transformation der Probleme in entsprechende Repräsentationen und Algorithmen für sowohl gatterbasierte als auch annealingbasierte Quantum Computing Hardware. Während Gatter-Architekturen das Problem in Form eines Quantenschaltkreis als Eingabe bekommen, werden zur Problembeschreibung für Quantum Annealing Hardware Ising-Hamiltonians oder Quadratic Unconstrained Binary Optimization (QUBO) Probleme verwendet. Viele Realwelt-Anwendungsfälle enthalten zudem Nebenbedingungen, die bei der Lösung berücksichtigt werden müssen. In Bezug auf die Ising-Hamiltonians werden diese Bedingungen in Penalty-Terme kodiert und mit Penalty-Parametern gewichtet, die bei einer Verletzung die entsprechende Lösung als invalide kennzeichnen. Diese Eigenheit beschreibt dann sogenannte *eingeschränkte* Ising-Hamiltonians.

Die vorliegende Arbeit beschäftigt sich mit der Anwendung und Optimierung eingeschränkter Ising-Hamiltonians. Zu Beginn werden hybride Algorithmen, die unter anderem auf einem eingeschränkten Ising-Hamiltonian basieren, für die Domäne der Spieltheorie vorgestellt. Die Ansätze werden sowohl auf gatterals auch annealingbasierter Hardware evaluiert und ihre Anwendbarkeit bestimmt. Anschließend wird eine adaptierte Kreuzentropie-Methode, zur Optimierung der Penalty-Parameter eingeschränkter Ising-Hamiltonians verschiedener akademischer kombinatorischer und graphbasierter Optimierungsprobleme, präsentiert. Die Evaluation der Hamiltonians mit den optimierten Penalty-Parametern zeigt eine signifikant gesteigerte Lösungsqualität, die mit einer größeren minimalen Spektrallücke in dem jeweiligen Eigenspektrum korreliert. Schließlich werden Methoden des maschinellen Lernens eingesetzt, um allgemeine Muster und Richtlinien zur Wahl der Penalty-Parameter für ausgewählte eingeschränkte Ising-Hamiltonians zu identifizieren und zu lernen.

Zusammenfassend hebt diese Arbeit die Bedeutung der Optimierung eingeschränkter Ising-Hamiltonians hervor.

## Abstract

Quantum computing hardware is still limited in the number of qubits, their coherence properties and their connectivity on quantum processors. Therefore, in order to achieve a possible practical quantum advantage over classical computing at an early stage, it is necessary to identify both relevant use cases with great potential and correspondingly implementable algorithms for currently available Noisy Intermediate-Scale Quantum (NISQ) computers. Many relevant problems from industry and science can be understood as optimization and search problems. The complexity from the user's perspective is the transformation of the problems into appropriate representations and algorithms for both gate-based and annealing-based quantum computing hardware. While quantum gate architectures receive the problem in the form of a quantum circuit as input, Ising-Hamiltonians or Quadratic Unconstrained Binary Optimization (QUBO) problems are used to describe the problem for quantum annealing hardware. Some real-world use cases also contain constraints that must be considered in the solution. With respect to Ising-Hamiltonians, these constraints are encoded in penalty-terms and weighted by penalty-parameters which, when violated, mark the corresponding solution as invalid. This characteristic then describes so-called *constrained* Ising-Hamiltonians.

The present work deals with the application and optimization of constrained Ising-Hamiltonians. At the beginning, hybrid algorithms based on a constrained Ising-Hamiltonian are presented for the domain of game theory. The approaches are evaluated on both gate- and annealing-based quantum hardware and their applicability is determined. Subsequently an adapted crossentropy method, for optimizing the penalty-parameters of constrained Ising-Hamiltonians of various academic combinatorial and graph-based optimization problems is presented. The evaluation of the Hamiltonians with the optimized penalty-parameters shows a significantly increased solution quality, which correlates with a larger minimum spectral gap in the respective eigenspectrum. Finally, machine learning methods are used to identify and learn general patterns and guidelines for choosing penalty-parameters for selected constrained Ising-Hamiltonians.

In summary, this work highlights the importance of optimizing constrained Ising-Hamiltonians.

# Inhaltsverzeichnis

1	Einleitung	1
1.1	Zugrundeliegende Vorarbeiten	3
1.2	Aufbau der Arbeit	5
2	Grundlagen des Quantum Computing	7
2.1	Qubits und ihre Eigenschaften	7
	2.1.1 Informationskodierung $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	7
	2.1.2 Multi-Qubit Systeme	9
	2.1.3 Quantengatter und Operatoren	0
	2.1.4 Eigenwerte und Eigenzustände	1
	2.1.5 Verschränkung	1
2.2	Adiabatisches Quantum Computing	2
	2.2.1 Zeitabhängiger Hamiltonian	3
	2.2.2 Adiabatisches Theorem	3
	2.2.3 Eigenspektrum	4
2.3	Quantum Annealing	5
	2.3.1 Optimierungsprobleme	7
	2.3.2 Ising-Modell	8
	2.3.3 Eingeschränkte Ising-Hamiltonians	0
	2.3.4 Implementierung in Hardware	5
2.4	Quantenschaltkreismodell 2'	7
2.1	2 4 1 Quantengatter des Quantenschaltkreismodells	8
	2.4.2 Algorithmen des Quantenschaltkreismodells	n
	2.4.3 Implementierung in Hardware	б
25	Kapitelzusammenfassung 39	8
2.0		5
3	Hybride Algorithmen für die Berechnung von reinen Nash-	
	Gleichgewichten in graphischen Spielen 41	1
3.1	Motivation und Zielsetzung	2
3.2	Grundlagen der Spieltheorie	3
	3.2.1 Spiele in Normalform	3
	3.2.2 Spiele in graphischer Form	4
	3.2.3 Reines Nash-Gleichgewicht	5
	3.2.4 Beste Antwortstrategien	6
3.3	Verwandte Arbeiten	6
3.4	Eingeschränkter Hamiltonian-Ansatz für das Finden von reinen	
	Nash-Gleichgewichten	8

	3.4.1 Konzept	48
	3.4.2 Aufbau der Experimente	51
	3.4.3 Ergebnisse und Diskussion	54
	3.4.4 Zusammenfassung und Ausblick	65
3.5	Grover-Ansatz zur Findung von reinen Nash-Gleichgewichten	66
	3.5.1 Konzept	67
	3.5.2 Aufbau der Experimente	70
	3.5.3 Ergebnisse und Diskussion	70
	3.5.4 Zusammenfassung und Ausblick	72
36	Vapitalzusammonfassung	72
0.0		12
4	Penalty-Parameter Optimierung eingeschränkter Hamiltonians	75
4.1	Motivation und Zielsetzung	76
42	Definition der Kreuzentropie-Methode	77
4.3	Verwandte Arbeiten	79
ч.0 Д Д	Konzept	80
1.1 1.5	Optimiorung der Penalty Parameter eingeschränkter Hamiltonians	00
ч.0	für getterbesierte Architekturen	82
	4.5.1 Aufbau der Experimente	82
	4.5.2 Freebrisse und Dickussion	81
	4.5.2 Zugemmenfergung und Auchlick	04 07
16	4.5.5 Zusammennassung und Ausblick	01
4.0	für annealinghagierte Architekturen	00
	107 anneanngDasierte Architekturen	00
	4.6.2 Englasized and Dislanging	00
	4.0.2 Ergebnisse und Diskussion	90
4 77	4.0.3 Zusammenrassung und Ausblick $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	95
4.1	Kapiteizusammentassung	95
5	Finfluss der Penalty-Parameter eingeschränkter Hamiltonians	
J	auf das Eigenspektrum	97
5.1	Motivation und Zielsetzung	98
5.2	Maschinelles Lernen	99
0.2	5.2.1 Clusteringanalyse	100
	5.2.2 Regressionsanalyse	101
53	Verwandte Arbeiten	104
5.0 5.4	Aufhau der Experimente	105
0.4	5 4 1 Detengeneriorung	105
	5.4.2 Methodon zur Evoluiorung	109
55	Freebnisse und Diskussion	100
0.0	5.5.1 Identifizieren von Trends im Figensnektrum von einge	109
	schrönkten Hemiltenians	100
	5.5.2 Lornon der Denalty Darameter für eingeschrönlich Uerzilteniene	119
	5.5.2 Anwondung vorhorsess stor und sufälligen Density Density	112
	5.5.5 Anwendung vornergesagter und zurähliger Penalty-Parameter	110
	aui Quantum Annealing Hardware	113

	5.5.4 Korrelation der Trajektorien-Cluster mit der Anzahl der glo- balen Optima und der Dichte der Probleminstanzen	116
	5.5.5 Korrelation skalierter Probleminstanzen mit der minimalen	110
	Spektrallücke	118
	5.5.6 Vergleich der minimalen Spektrallücken von logischen und	
	eingebetteten Instanzen	119
5.6	Kapitelzusammenfassung und Ausblick	120
6	Zusammenfassung und Ausblick	123
<b>6</b> 6.1	Zusammenfassung und Ausblick	<b>123</b> 123
<b>6</b> 6.1 6.2	Zusammenfassung	<b>123</b> 123 125
6 6.1 6.2 Abbi	Zusammenfassung       und       Ausblick         Zusammenfassung	<ol> <li>123</li> <li>123</li> <li>125</li> <li>132</li> </ol>

## 1 Einleitung

Die Untersuchung der physikalischen Grenzen klassischer Computersysteme auch in Hinblick auf das Mooresche Gesetz erfordert alternative Konzepte, um den stetig wachsenden Bedarf an Rechenleistung zu befriedigen [80]. Die Idee eines Computers, der auf den Prinzipien der Quantenmechanik beruht, kam bereits in den 1970er Jahren auf. Paul Benioff, Richard Feynman und Yuri Manin formulierten relativ zeitgleich ihre Theorien quantenmechanische Phänomene, wie die Überlagerung und Verschränkung, zu nutzen, um ein neues Modell zur Berechnung zu entwickeln und bildeten somit die theoretischen Grundlagen für das Quantum Computing (QC) [14, 63, 111]. Vermehrte Aufmerksamkeit generierten die Publikationen von Peter Shor [158] und Lov Grover [79] Mitte der 90er Jahre, in denen zwei Quantenalgorithmen vorgestellt wurden, die eine theoretische Beschleunigung für praktische Anwendungen gegenüber ihrer entsprechenden klassischen Algorithmen liefern. Während der Grover-Algorithmus eine polynomielle Beschleunigung bei der Durchsuchung einer unstrukturierten Datenbank liefert, kann beim Shor-Algorithmus eine exponentielle Beschleunigung bei der Primfaktorzerlegung gegenüber des jeweiligen klassischen Gegenstücks erzielt werden. Auch wenn diese Algorithmen potenziell viele Bereiche in der Wissenschaft und Wirtschaft revolutionieren werden, sind deren praktische Umsetzung für relevante Problemgrößen zum aktuellen Zeitpunkt zu ressourcenintensiv [73].

Aktuelle Quantum Computing Hardware ist noch immer sehr begrenzt in der Anzahl der Qubits, deren Kohärenzeigenschaften und deren Konnektivität auf den Quantenprozessoren (engl. Quantum Processing Units (QPUs)). Um daher bereits frühzeitig einen möglichen praktischen Quantenvorteil zum klassischen Computing zu erzielen, gilt es deshalb, sowohl relevante Anwendungsfälle mit großem Potenzial, als auch entsprechend umsetzbare Algorithmen für die aktuell verfügbare Noisy Intermediate-Scale Quantum (NISQ) Computer zu identifizieren und zu entwickeln. Für die Industrie sind vor allem die geschäftsrelevanten Anwendungsfälle von Interesse, die von Lager-, Routing-, Lieferkettenbis hin zur Portfoliooptimierung reichen. Aber auch komplexe Entscheidungsfindungsprozesse spielen meist eine wichtige Rolle in der Unternehmensplanung und fallen in diese Kategorie. Konventionelle Spieltheorie wird dabei oftmals als Modell verwendet, um diese strategischen Prozesse abzubilden und entsprechende Lösungskonzepte, wie beispielsweise Nash-Gleichgewichte, zu berechnen. Dabei wird aktuell auf klassische Heuristiken und Methoden, wie bestärkendes Lernen, ganzzahlige lineare Optimierung oder auch die zufällige Suche zurückgegriffen [153, 171]. Jedoch gibt es Konstellationen bei denen die Berechnung genannter Lösungskonzepte NP-schwer ist [76] und Quantum Computing einen Mehrwert bieten könnte.

Die erwähnten Anwendungsfälle lassen sich mathematisch als Optimierungsbeziehungsweise Suchprobleme auffassen, bei denen eine entsprechende Zielfunktion minimiert, maximiert oder im Falle des Suchproblems erfüllt werden muss. Es gibt verschiedene Möglichkeiten solche Problemstellungen auf Quantum Computing Hardware zu lösen [131]. Dabei wird grundsätzlich zwischen den gatterbasierten und annealingbasierten Hardware-Architekturen unterschieden, die verschiedene Problemformulierungen und Algorithmen unterstützen. Die Komplexität aus Anwendersicht ist dabei die entsprechende Transformation des zu lösenden Problems zu erarbeiten, sodass diese von Quantum Computern verstanden und gelöst werden können. Während Gatter-Architekturen das Problem in Form eines Quantenschaltkreis als Eingabe bekommen, wird zur Problembeschreibung für Quantum Annealing Hardware ein Ising-Hamiltonian oder das Quadratic Unconstrained Binary Optimization (QUBO) Problem verwendet. Viele theoretische, aber vor allem Realwelt-Anwendungsfälle, wie oben eingangs erwähnt, beinhalten zusätzliche Nebenbedingungen. Diese werden dann im Falle von Ising-Hamiltonians in zusätzlichen Penalty-Termen kodiert und mit Penalty-Parametern gewichtet, um Lösungen als invalide zu deklarieren, falls eine Bedingung nicht eingehalten wurde. Diese Besonderheit beschreibt dann sogenannte *eingeschränkte* Ising-Hamiltonians. In der vorliegenden Arbeit wird sich mit der Anwendung und Optimierung solcher eingeschränkter Ising-Hamiltonians beschäftigt. Die Anwendungsfälle reichen von der Domäne der Spieltheorie bis hin zu akademischen, kombinatorischen und graphbasierten Optimierungsproblemen, die die Grundlage für viele Industrieproblemstellungen darstellen. Im ersten Teil der Arbeit werden die ersten quanten-klassischen hybriden Ansätze zur Findung von reinen Nash-Gleichgewichten in graphbasierten Spielen, sowohl für die Gatter- als auch Annealing-Architekturen identifiziert, erarbeitet und empirisch gegenüber klassischen Methoden evaluiert. Darüber hinaus wird im zweiten Teil eine Hardwarearchitektur unabhängige unterstützende Methode zur Optimierung der Penalty-Parameter eingeschränkter Ising-Hamiltonians vorgestellt, um den aktuellen frühen Reifegrad der NISQ-Computer entgegenzuwirken und die Lösungsqualität bereits auf bestehender QC-Hardware zu steigern, ohne dabei auf ressourcenintensive Fehlerkorrektur zurückgreifen zu müssen. Abschließend wird eine auf maschinellen Lernens (ML) basierte Analyse der Penalty-Parameter verschiedener eingeschränkter Ising-Hamiltonians durchgeführt, um deren Einfluss auf das jeweilige Eigenspektrum zu bestimmen und somit zum einen die Lösbarkeit mittels Quantum Annealing Hardware zu beurteilen und zum anderen Richtlinien zur effizienteren Problemformulierung mittels der Penalty-Parameter zu liefern.

Der Schwerpunkt dieser Arbeit liegt auf der Ausarbeitung unterstützender Methoden, um Problemstellungen in Form von eingeschränkten Hamiltonians für aktuelle NISQ-Hardware zu optimieren. Dies ist motiviert aus der Beobachtung, dass ein beträchtlicher Teil der Forschung sich zwar mit der Findung neuer Formulierung für viele unterschiedliche Problemstellungen beschäftigt, jedoch weniger auf deren Lösbarkeit beziehungsweise Optimierungsmöglichkeiten eingegangen wird. Die Arbeit leistet somit ihren Beitrag, diese Lücke weiter zu schließen.

### 1.1 Zugrundeliegende Vorarbeiten

Ein Großteil der Inhalte dieser Arbeit wurde bereits im Rahmen internationaler Fachkonferenzen und Journals veröffentlicht. Nachfolgend wird der Eigenanteil des Autors der vorliegenden Arbeit an den für diese Arbeit relevanten und bereits publizierten Inhalten, im Einzelnen aufgeschlüsselt. Die jeweiligen Vorveröffentlichungen werden dabei in Reihenfolge ihrer ersten inhaltlichen Erwähnung aufgelistet. Zusätzlich zu dieser Übersicht wird zu Beginn der Inhaltskapitel 3, 4 und 5 noch einmal auf die bereits veröffentlichten Inhaltsteile eingegangen. Für alle im Folgenden aufgeführten Publikationen gilt, dass Prof. Dr. Claudia Linnhoff-Popien in ihrer Rolle als Doktormutter des Autors stets mit konstruktiver Kritik und wertvollen Gedanken mitgewirkt hat. Dies gilt insbesondere dann, wenn sie als Autorin aufgeführt ist.

(i) A Quantum Annealing Algorithm for Finding Pure Nash Equilibria in Graphical Games [143] Diese Publikation stellt den ersten hybriden Quantum Annealing Algorithmus zum Finden reiner Nash-Gleichgewichte in graphischen Spielen vor. Der Ansatz wird auf Spielen unterschiedlicher Graphtopologien auf D-Wave Quantum Annealing Hardware evaluiert und mit klassischen Methoden verglichen. Die Ergebnisse zeigen, dass Q-Nash auf D-Wave Quantum Annealing Hardware konkurrenzfähig ist.

Diese Veröffentlichung bildet die Grundlage für Kapitel 3.4. Idee, Konzept, Implementierung, Evaluation und Ausarbeitung stammen vom Autor. Die Co-Autoren halfen bei der Einordnung und der Diskussion der Ergebnisse.

(ii) A Grover based Quantum Algorithm for Finding Pure Nash Equilibria in Graphical Games [142] Im Rahmen dieser Publikation wird ein auf dem Grover-Algorithmus basierender hybrider Ansatz zur Berechnung reiner Nash-Gleichgewichte in graphischen Spielen vorgestellt. Es wird die Transformation der Problemstellung in die Orakelfunktion des Grover-Algorithmus beschrieben und anschließend wird der Ansatz auf seine NISQ-Fähigkeit evaluiert.

Die Publikation dient als Grundlage für Kapitel 3.5. Die Idee, das Konzept und eine prototypische Implementierung stammen vom Autor. Santiago Londoño Castillo fertigte eine finale Implementierung an, unterstützte bei der Ausformulierung des Methodenteils und stand als Diskussionspartner zur Verfügung.

(iii) Cross Entropy Hyperparameter Optimization for Constrained Problem Hamiltonians Applied to QAOA [141] Kern dieser Arbeit ist eine Optimierungsmethode, die auf Kreuzentropie basiert und auf die Penalty-Parameter eingeschränkter Hamiltonians angewendet werden kann. Infolgedessen ist der klassische Optimierer des QAOAs in der Lage bessere Gatterparameter für den QAOA-Schaltkreis zu finden, was sich in einer gesteigerten Lösungsqualität widerspiegelt.

Die Publikation bildet die Grundlage für Kapitel 4.5. Die Idee der Änderung der Penalty-Parameter zur Optimierung der Hamiltonians stammt dabei vom Autor. Thomy Phan steuerte als Vorschlag die Kreuzentropie-Methode bei. Alexander Impertro unterstützte bei der Implementierung und Evaluierung des Ansatzes. Thomas Gabor und Sebastian Feld standen für äußert fruchtbare Diskussionen zu inhaltlichen Fragen der Veröffentlichung zur Verfügung.

(iv) Cross Entropy Optimization of Constrained Problem Hamiltonians for Quantum Annealing [140] Basierend auf Publikation [141] wurde die Kreuzentropie-Methode auf ein breiteres Spektrum an eingeschränkten Hamiltonians angewendet und auf der D-Wave Quantum Annealing Hardware evaluiert. Die numerische Berechnung des Hamiltonian-Eigenspektrums weist eine Korrelation der minimalen Spektrallücke, die als Metrik für die Anwendbarkeit von Hamiltonians auf adiabatischen Quantum Computern verwendet werden kann, und den optimierten Penalty-Parametern auf. Selbst bei der Anwendung auf aktueller D-Wave Quantum Annealing Hardware bewirkt die Optimierung der Penalty-Parameter eine signifikant gesteigerte Lösungsqualität im Vergleich zu zufällig gewählten Parametern.

Die Veröffentlichung bildet die Grundlage für Kapitel 4.6. Die Idee und das Konzept stammen vom Autor, der ebenfalls hauptsächlich für die Ausarbeitung der Veröffentlichung zuständig war. Alexander Impertro übernahm die Implementierung. Zudem unterstütze er bei der Darstellung der Experimente und Ergebnisse in der Publikation und stand dem Autor zudem beratend zur Seite.

(v) The Effect of Penalty Factors of Constrained Hamiltonians on the Eigenspectrum in Quantum Annealing [144] Die Erkenntnisse aus [141] und [140] führten in dieser Publikation zu einer umfangreichen Untersuchung der Penalty-Parameter eingeschränkter Hamiltonians und deren Einfluss auf das jeweilige Eigenspektrum mit Methoden des maschinellen Lernens. Mit Hilfe der ML-Methoden wird die Korrelation der Penalty-Parameter und der minimalen Spektrallücke für sechs ausgewählte eingeschränkte Hamiltonians analysiert, und es wird gezeigt, dass die Regression in Form eines neuronalen Netzes in der Lage ist, die besten Penalty-Parameter in einem vordefinierten Versuchsaufbau für verschiedene Probleminstanzen vorherzusagen.

Die Publikation bildet die Grundlage für Kapitel 5. Die Idee, Implementierung und Evaluation stammen dabei vom Autor selbst. Daniel Ratke unterstützte bei der Implementierung der Methoden des maschinellen Lernens und der Generierung des Datensatzes. Jonas Nüßlein, Thomas Gabor und Sebastian Feld standen über den gesamten Zeitraum für kritische Diskussionen zur Verfügung und unterstützten bei der Anfertigung der Publikation.

## 1.2 Aufbau der Arbeit

Die vorliegende Arbeit unterteilt sich in insgesamt sechs Kapitel. In Kapitel 2 werden zunächst die wichtigsten Quantum Computing Grundlagen für das weitere Verständnis der Arbeit dargelegt. Dabei wird sowohl auf die Eigenschaften der Qubits als auch auf die zwei fundamentalen Modelle des Quantum Computings, das adiabatische Quantum Computing Modell und das Quantenschaltkreismodell, eingegangen. Zudem werden die Grundlagen in Bezug auf Optimierungsprobleme, deren Repräsentation als Ising-Hamiltonians und die Quantum Annealing Heuristik beleuchtet. Weitere Definitionen und Grundlagen, die sich speziell auf nur eines der folgenden Inhaltskapitel beziehen, werden am Anfang des jeweiligen Kapitels erörtert.

In Kapitel 3 werden neben den spieltheoretischen Definitionen hybride Algorithmen vorgestellt, die für das Berechnen von reinen Nash-Gleichgewichten in graphischen Spielen entwickelt wurden. Die Algorithmen, die unter anderem auf einem eingeschränkten Hamiltonian basieren, werden auf realer Quantum Computing Hardware evaluiert und auf ihre Anwendbarkeit auf NISQ-Hardware beurteilt.

Anschließend wird in Kapitel 4 die Möglichkeit untersucht, eingeschränkte Hamiltonians zu optimieren, um eine gesteigerte Lösungsqualität auf aktueller Quantum Computing Hardware zu erzielen. Es wird die Kreuzentropie-Methode vorgestellt, mit der die Nebenbedingungen der Hamiltonians entsprechend effektiv mit den Penalty-Parametern gewichtet werden können. Dieses Verfahren wird für verschiedene ausgewählte eingeschränkte Hamiltonians sowohl auf gatter- als auch annealingbasierter Hardware evaluiert.

In Kapitel 5 wird die Wahl der Penalty-Parameter unterschiedlicher Hamiltonians und deren Einfluss auf das jeweilige Eigenspektrum auf allgemeine Muster und Trends mit Hilfe von Methoden des maschinellen Lernens untersucht.

Abschließend fasst Kapitel 6 die Ergebnisse der vorliegenden Arbeit zusammen. Weiterführende Aspekte werden als Ausblick für zukünftige Forschungsarbeiten diskutiert.

#### 1 Einleitung

# 2 Grundlagen des Quantum Computing

Dieses Kapitel beschäftigt sich zum einen mit den Grundlagen des Quantum Computings und dessen Informationskodierung, als auch mit den verschiedenen Modellen des Quantum Computings, die für das Verständnis der Arbeit notwendig sind. In Abschnitt 2.1.1 wird die quantenmechanische Informationskodierung so wie die Eigenschaften von Qubits erläutert. In Abschnitt 2.2 wird das adiabatische Quantum Computing und dessen Funktionsweise eingeführt. Im Anschluss daran wird die verwandte Metaheuristik Quantum Annealing in Abschnitt 2.3 vorgestellt, welche sowohl klassisch simuliert, als auch auf Quantenhardware ausgeführt werden kann. Zuletzt wird in Abschnitt 2.4 auf das Quantenschaltkreismodell eingegangen und verschiedene bekannte Quantenschaltkreisalgorithmen, wie der *Quantum Approximate Optimization Algorithm* (QAOA) und der *Grover-Algorithmus* erläutert. Diese beiden Algorithmen dienen neben der Quantum Annealing Heuristik als Basis für die in Kapitel 3 entwickelten hybriden Ansätze.

## 2.1 Qubits und ihre Eigenschaften

Das nachfolgende Unterkapitel befasst sich mit der Informationskodierung mittels Qubits und deren Eigenschaften. Zudem wird deren Zusammenschluss zu Multi-Qubit Quantensystemen aufgezeigt und die Möglichkeiten zur Zustandsänderung vorgestellt.

#### 2.1.1 Informationskodierung

Klassisches und Quantum Computing unterscheiden sich grundlegend in der Form der Zustandsbeschreibung. Der Zustand eines klassischen Algorithmus kann durch ein Register R mit klassischen Bits dargestellt werden, die jeweils den Wert 0 oder 1 annehmen können. Der Zustand eines Quantenalgorithmus wird hingegen durch ein Register Q dargestellt, das *Quantum-Bits* (Qubits) enthält.

Ein grundlegender Unterschied zwischen Bits und Qubits ist die Eigenschaft der Superposition. Das heißt, ein Qubit kann sich gleichzeitig in den Zuständen 0 und 1 befinden. Ein Superpositionszustand, der in der Dirac-Notation als  $|\psi\rangle$  beschrieben wird, lässt sich mathematisch als Linearkombination von zwei Basiszuständen, die konventionell als  $|0\rangle$  und  $|1\rangle$  gewählt werden, darstellen,

 $|\phi\rangle = \alpha |0\rangle + \beta |1\rangle$ . Mit  $\alpha$  und  $\beta$  als zwei komplexe Zahlen. Eine alternative Schreibweise neben der Dirac-Notation ist die folgende Matrix-Notation,

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \ |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ und } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Zu beachten ist, dass in der Quantenmechanik Superpositionszustände nicht direkt beobachtet werden können. Wenn ein Qubit in Bezug auf eine bestimmte Basis gemessen wird, kollabiert es augenblicklich auf einen der Basiszustände: Nämlich entweder  $|0\rangle$  (beobachtet als klassischer Zustand 0) mit der Wahrscheinlichkeit  $|\alpha^2|$  oder  $|1\rangle$  (beobachtet als klassischer Zustand 1) mit einer Wahrscheinlichkeit von  $|\beta^2|$ . Daraus ergibt sich  $|\alpha^2| + |\beta^2| = 1$  mit  $|\alpha|$  als Betrag der komplexen Zahl  $|\alpha| = |x + iy| = \sqrt{x^2 + y^2}$ . [112]

Eine andere Möglichkeit dieses schwer verständliche Phänomen zu betrachten, besteht darin zu sagen, dass jedes Messgerät ein Paar orthogonaler Basen definiert (beispielsweise  $|0\rangle$  und  $|1\rangle$ ). Beim Messen wird dann der Superpositionszustand des Qubits auf eines der Basen projiziert. Die Projektion beziehungsweise Messung ist ein irreversibler Vorgang, der  $\alpha$  und  $\beta$  ändert, sodass es unmöglich ist, denselben Qubit-Zustand zweimal zu lesen [112].



Abbildung 2.1: Die Bloch-Einheitskugel.

Graphisch lässt sich ein zweidimensionaler komplexer Zustandsvektor  $|\psi\rangle$  als Punkt auf der Oberfläche der dreidimensionalen Bloch-Einheitskugel visualisieren, wie in Abbildung 2.1 dargestellt. Dabei entsprechen der Nord- bzw. Südpol der Kugel den Standard-Basiszuständen  $|0\rangle$  bzw.  $|1\rangle$ . Die komplexen Amplituden  $\alpha$  und  $\beta$  eines Quantenzustands werden durch reelle Werte ausgedrückt und ein Term, der die relative Phase zwischen ihnen repräsentiert, hinzugefügt:

$$|\psi\rangle = \alpha |0\rangle + \beta e^{i\phi} |1\rangle \text{ mit } \alpha, \beta, \phi \in \mathbb{R}$$

Mit Hilfe der trigonometrischen Funktionen lässt sich  $|\alpha^2| + |\beta^2| = 1$  als  $\sqrt{\sin^2 x + \cos^2 x} = 1$  ausdrücken, sodass sich  $\alpha$  und  $\beta$  durch eine einzige Variable  $\theta$  ersetzen lassen, mit  $\alpha = \cos \frac{\theta}{2}$  und  $\beta = \sin \frac{\theta}{2}$ . Dadurch kann der Zustand des Qubits durch zwei Winkel  $\phi$  und  $\theta$  ausgedrückt werden,

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \text{ mit } \theta, \phi \in \mathbb{R}$$

Wenn nun  $\theta$  und  $\phi$  als Kugelkoordinaten interpretiert werden, kann jeder beliebige Ein-Qubit-Zustand auf der Einheitskugel abgebildet werden [86].

#### 2.1.2 Multi-Qubit Systeme

Quantenmechanische Systeme bestehen meist aus mehreren Qubits und deren Zustandsvektoren. Zwei Qubit-Zustände  $|\psi_1\rangle = (\alpha_1, \beta_1)^T$  und  $|\psi_2\rangle = (\alpha_2, \beta_2)^T$ können durch Anwendung des Tensorprodukts  $\otimes$  auf deren Zustandsvektoren zu einen gemeinsamen Zustand  $|\psi\rangle = |\psi_1, \psi_2\rangle$  kombiniert werden:

$$|\psi\rangle = (\alpha_1, \beta_1)^T \otimes (\alpha_2, \beta_2)^T = (\alpha_1 \alpha_2, \alpha_1 \beta_2, \beta_1 \alpha_2, \beta_1 \beta_2)^2.$$

Ein Zwei-Qubit-Zustandsraum hat vier Basiszustände, die in Matrix-Notation wie folgt dargestellt werden:

$$|00\rangle = \begin{pmatrix} 1\\0 \end{pmatrix} \otimes \begin{pmatrix} 1\\0 \end{pmatrix} = \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix}$$
$$|01\rangle = \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix}$$
$$|10\rangle = \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix}$$
$$|11\rangle = \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix}$$

Indem man  $\gamma_1 = \alpha_1 \alpha_2$ ,  $\gamma_2 = \alpha_1 \beta_2$ ,  $\gamma_3 = \beta_1 \alpha_2$  und  $\gamma_4 = \beta_1 \beta_2$  setzt, kann der Zustand  $|\psi\rangle$  folgendermaßen in Bezug auf seine Basis angegeben werden [112]:

$$\begin{aligned} |\psi\rangle &= |\psi_1, \psi_2\rangle = (\gamma_1, \gamma_2, \gamma_3, \gamma_4)^T \\ &= \gamma_1 |00\rangle + \gamma_2 |01\rangle + \gamma_3 |10\rangle + \gamma_4 |11\rangle, \text{mit} \\ 1 &= |\gamma_1|^2 + |\gamma_2|^2 + |\gamma_3|^2 + |\gamma_4|^2. \end{aligned}$$

Diese Werte stellen die Wahrscheinlichkeiten dar, dass das Qubit-Paar in jedem möglichen Zustand beobachtet beziehungsweise gemessen wird. D.h., eine Messung beider Qubits resultiert zu einen der vier möglichen Bitfolgen, die den vier Basisvektoren entsprechen. Anhand dieses Beispiels wird einer der bedeutendsten Vorteilen von Multi-Qubit Systemen ersichtlich. Die Dimension bzw. Informationskapazität des Zustandsraums wächst exponentiell mit der Anzahl der Qubits n.

#### 2.1.3 Quantengatter und Operatoren

Ein Quantengatter ist ein unitärer Operator der den quantenmechanischen Zustand eines Systems ändern kann. Diese Transformation wird mathematisch durch eine unitäre Matrix U mit komplexen Einträgen beschrieben. Dabei ist eine Matrix U unitär wenn gilt,

$$UU^{\dagger} = U^{\dagger}U = I,$$

mit  $U^{\dagger}$  als die komplex konjugierte und transponierte Matrix  $(U^*)^T$  und  $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  der Einheitsmatrix. Die Änderung eines Ein-Qubit-Zustands  $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$  durch eine 2×2-Matrix U wird dann folgendermaßen beschrieben,

$$|\psi\rangle \to U |\psi\rangle = \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} U_{00}\alpha + U_{01}\beta \\ U_{10}\alpha + U_{11}\beta \end{pmatrix} = |\psi'\rangle$$

Bildlich betrachtet, wäre im Falle des Ein-Qubit-Zustands  $|\psi\rangle$  die Transformation U eine entsprechende Rotation des Zustandsvektors auf der Bloch-Einheitskugel, zu dem neuen Zustand  $|\psi'\rangle$ .

Die drei grundlegenden unitären 2 × 2-Transformationen werden durch die Pauli-Operatoren  $\sigma^x, \sigma^y$  und  $\sigma^z$  ausgedrückt,

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ und } \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Beispielsweise lässt sich so überprüfen, dass die Anwendung des Pauli-Operators  $\sigma^x$  auf einen gegebenen Zustandsvektor  $|\psi\rangle = (\alpha, \beta)^T$  eine Spiegelung an der x, y-Ebene bewirkt, so dass sich die Amplituden  $\alpha, \beta$ vertauschen  $(\sigma^x | 1 \rangle \rightarrow | 0 \rangle$  und  $\sigma^x | 0 \rangle \rightarrow | 1 \rangle$ ). Somit stellt  $\sigma^x$  das Quantenäquivalent eines klassischen NOT-Gatters dar. Ein wesentlicher Unterschied zwischen klassischen und Quantengattern besteht darin, dass die Quantenlogik umkehrbar sein muss. Das heißt, dass es für jeden Quantenoperator möglich sein muss, die Eingangsbits durch Ablesen der Ausgangsbits zu bestimmen.[112]

Abschließend sei erwähnt, dass Operatoren auch auf mehrere Qubits angewendet werden können. Diese Operatoren werden dann mit dem Kronecker-Produkt kombiniert. Zum Beispiel, wenn  $U_1$  und  $U_2$  Operatoren sind, die auf zwei verschiedene Qubits wirken, dann ist der vollständige Operator U, der auf das kombinierte Zwei-Qubit System wirkt, durch  $U = U_1 \otimes U_2$  gegeben [92]. Neben den grundlegenden Pauli-Operatoren gibt es zahlreiche weitere Operatoren, die für die Berechnung im Quantenschaltkreismodell relevant sind und im entsprechenden Kapitel 2.4 vorgestellt werden.

#### 2.1.4 Eigenwerte und Eigenzustände

Gemäß der Matrix-Darstellung von Zuständen und Operatoren, ist ein Eigenvektor einer quadratischen Matrix A ein von Null verschiedener Vektor v, so dass die Multiplikation von A mit v ein konstantes Vielfaches von v ergibt, d.h.  $A \cdot v = \lambda \cdot v$  für einen Skalar  $\lambda$ . Der Koeffizient  $\lambda$  ist der Eigenwert von A, der dem Eigenvektor v zugeordnet ist. Die Pauli-Matrizen, die im vorherigen Abschnitt 2.1.3 beschrieben wurden, haben jeweils die Eigenwerte +1 und -1. Nachfolgenden sind zur Veranschaulichung die Eigenwerte und -vektoren von  $\sigma^z$  dargestellt [112].

$$\sigma^{z} |0\rangle = +1 |0\rangle$$
  
$$\sigma^{z} |1\rangle = -1 |1\rangle.$$

Mit Hilfe der Matrix-Notation können diese Gleichungen folgendermaßen verifiziert werden.

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = +1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Andere unitäre Matrizen können durchaus andere Eigenwerte und oder Eigenvektoren haben. Die Eigenvektoren und -werte bilden das Eigenspektrum einer entsprechenden Matrix, das in Abschnitt 2.2.3 näher erläutert wird und eine wichtige Grundlage für Kapitel 5 darstellt.

#### 2.1.5 Verschränkung

Eine weitere wesentliche Eigenschaft, die die Quantenberechnung von der klassischen Berechnung unterscheidet, ist die *Verschränkung*. Qubits bzw. Quantenteilchen können so miteinander interagieren, dass sie nach der Interaktion nicht mehr unabhängig voneinander sind. Mathematisch betrachtet sind ihre Amplituden  $\alpha$ 's und  $\beta$ 's miteinander verknüpft, so dass die Wahrscheinlichkeit,  $|\psi_1 = 1\rangle$  zu beobachten, mit der Wahrscheinlichkeit  $|\psi_2 = 1\rangle$  zu beobachten, korreliert [112].

Dieses Phänomen der Verschränkung kann anhand des Zwei-Qubit *Bell-Zustands* erläutert werden:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

In einem Bell-Zustand sind die zwei Teilchen so verschränkt, dass der Zustand jedes Teilchens mit dem des anderen korreliert, sogar dann, wenn sie durch eine physisch große Distanz getrennt sind. Dabei ist die Wahrscheinlichkeit,  $|01\rangle$  oder  $|10\rangle$  zu beobachten 0% und die Wahrscheinlichkeiten,  $|00\rangle$  oder  $|11\rangle$ zu beobachten jeweils 50%. Man beachte, dass jedes einzelne Qubit die gleiche Wahrscheinlichkeit hat, als  $|0\rangle$  oder  $|1\rangle$  gemessen zu werden. Aufgrund der Verschränkung können aber die Wahrscheinlichkeiten, die mit den Multi-Qubit-Zuständen wie  $|00\rangle$  oder  $|11\rangle$  verbunden sind, nicht in die Produkte der Einzelwahrscheinlichkeiten (vgl. Abschnitt 2.1.3) zerlegt werden.

Eine weitere interessante Eigenschaft der Verschränkung ist, dass bei der Messung des beispielsweise ersten verschränkten Qubits  $\psi_1$ , bei der sein Zustand zerstört wird, die Korrelation mit  $\psi_2$  solange bestehen bleiben, bis dieses ebenfalls gemessen wird.

### 2.2 Adiabatisches Quantum Computing

Adiabatisches Quantum Computing (AQC) ist zusammen mit dem Quantenschaltkreismodell eines der beiden grundlegenden Berechnungsmodelle des Quantum Computings. Während bei ersterem die Algorithmen durch eine Aneinanderreihung von unitären Quantengattern beschrieben werden (mehr dazu in Kapitel 2.4), verläuft die Berechnung beim AQC von einem *initialen Hamiltonian*, dessen Grundzustand leicht herzustellen ist, zu einem finalen *Problem-Hamiltonian*, dessen Grundzustand die Lösung des Problems kodiert. Das *adiabatische Theorem* das im nachfolgenden in Abschnitt 2.2.2 beschrieben wird, garantiert dabei, dass das System unter gewissen Voraussetzungen dem momentanen Grundzustand folgt [4].

Die Idee, die Lösung eines Problems in den Grundzustand eines Hamiltonians zu kodieren, erschien erstmals 1988 im Zusammenhang mit der Lösung klassischer kombinatorischer Optimierungsprobleme, die als quantenstochastische Optimierung bezeichnet wurde [134]. Daraufhin folgten weitere ähnliche Arbeiten, die auch die verwandte Quantum Annealing Heuristik beinhalteten [6, 7, 65, 98, 154], und in der Literatur oft fälschlicherweise synonymisch zum AQC verwendet wird. In den nachfolgenden Abschnitten wird das allgemeine Verfahren des AQC vorgestellt, bevor in Kapitel 2.3 auf die verwandte Quantum Annealing Heuristik eingegangen wird.

#### 2.2.1 Zeitabhängiger Hamiltonian

Der Hamiltonian eines quantenmechanischen Systems ist ein Operator, der die Gesamtenergie, einschließlich der kinetischen und der potenziellen Energie, dieses Systems beschreibt. Mathematisch werden Hamiltonians durch hermitesche Matrizen repräsentiert, deren Eigenwerte reellwertig sind. Die Menge aller möglichen Eigenwerte, die aus einer Messung der Gesamtenergie des Systems resultieren wird auch Eigenspektrum oder Energiespektrum des Hamiltonians genannt. Hamiltonians bilden die elementaren Bausteine der AQC-Algorithmen und werden im Folgenden beschrieben.

Ein AQC-Algorithmus arbeitet auf einem Quantenregister Q mit n Qubits. Der grundlegende Prozess des AQC ist dann die physikalische Interpolation zwischen einem initialen Hamiltonian  $\mathcal{H}_I$  mit einer leicht herzustellenden minimalen Energiekonfiguration (auch Grundzustand genannt) und einem Problem-Hamiltonian  $\mathcal{H}_P$ , der die Zielfunktion f(x) eines zu lösenden Problems P so kodiert, dass der Grundzustand von Q ein Eigenzustand von  $\mathcal{H}_P$  mit minimalem Eigenwert ist. Das heißt, der Grundzustand des Registers Q entspricht der optimalen Lösung von P. [112]

Dieses physikalische Prinzip kann dann durch einen zeitabhängigen Hamiltonian  $\mathcal{H}(t)$  beschrieben werden, wie in Gleichung (2.1) dargestellt:

$$\mathcal{H}(t) = s(t)\mathcal{H}_I + (1 - s(t))\mathcal{H}_P \tag{2.1}$$

Hierbei entspricht die Funktion s(t) dem adiabatischen Evolutionspfad, der über eine gewisse Evolutionszeit  $t_f$  stetig von 1 bis 0 abnimmt,  $t: 0 \to t_f$ . Im einfachsten Fall kann hier ein linearer Pfad  $s(t) = 1 - \frac{t}{t_f}$  verwendet werden [112].

Laut dem adiabatischen Theorem ist die Wahrscheinlichkeit, während der Evolution im Grundzustand zu verbleiben, nahe eins, wenn dieser Übergang hinreichend langsam erfolgt [4]. Eine entsprechende Definition des Theorems ist nachfolgend gegeben.

#### 2.2.2 Adiabatisches Theorem

Das adiabatische Theorem wurde ursprünglich von Born und Fock [22] formuliert und wurde seitdem mehrfach in der Literatur aufgegriffen [3, 49, 138]. Es bildet die Grundlage für das adiabatische Berechnungsmodell und kann anhand der Schrödinger-Gleichung beschrieben werden,

$$i\hbar \frac{d}{dt} |\phi_t\rangle = \mathcal{H}(t) |\phi_t\rangle.$$
 (2.2)

Hierbei beschreibt *i* die imaginäre Einheit und  $\hbar = \frac{h}{2\pi}$  mit *h* als die Planck-Konstante. Die Gleichung besagt, dass die momentane Änderungsrate des adiabatischen Prozesses proportional zur Energie des Prozesses ist, die durch den zeitabhängigen Hamiltonian zur Zeit *t* beschrieben wird [49, 112].

Um die Bedingungen des Theorems zu veranschaulichen, kann die Schrödinger-Gleichung (2.2) folgendermaßen verallgemeinert werden,

$$\frac{d}{ds} \left| \phi_s \right\rangle = -i\tau(s) \widetilde{\mathcal{H}}(s) \left| \phi_s \right\rangle.$$
(2.3)

Der Hamiltonian  $\mathcal{H}(t)$  aus (2.2), der sich über die Zeit  $t: 0 \to t_f$  entwickelt, kann durch einen äquivalenten Hamiltonian  $\widetilde{\mathcal{H}}(s)$  und einer streng von 1 bis 0 abnehmenden Funktion s(t), die die Zeitspanne  $t: 0 \to t_f$  abbildet, generalisiert werden. Die Funktion  $\tau(s)$  definiert die Rate mit der sich der Hamiltonian in Abhängigkeit von s ändert. Zudem sei  $\tau(s)$  durch,

$$\tau(s) \gg \frac{||\frac{d}{ds}\widetilde{\mathcal{H}}(s)||}{(\delta_{min})^2},\tag{2.4}$$

begrenzt, wobei  $||\cdot||$  die Matrixnorm der Euklidischen Distanz beschreibt und  $\delta_{min}$  die minimale Spektrallücke (engl. Minimal Spectral Gap (MSG)) darstellt. Diese ist als die (positive) Differenz zwischen den Eigenwerten des Grundzustands und des ersten angeregten Zustands zu einem beliebigen Zeitpunkt s definiert, unter der Annahme, dass  $\widetilde{\mathcal{H}}(s)$  während der gesamten Zeit s einen nicht entarteten Grundzustand hat.

Unter diesen Voraussetzungen besagt das adiabatische Theorem, dass,

- 1. falls sich das System im initialen Grundzustand zum Zeitpunkt s = 0 befindet,
- 2. und  $\delta_{min}$  zu jedem Zeitpunkt s strikt größer als 0 ist, und
- 3. sich der Prozess gemäß Gleichung (2.4) langsam genug entwickelt,

dieser Prozess mit ausreichend hoher Wahrscheinlichkeit im Grundzustand des Problem-Hamiltonians verbleiben wird. [22, 112]

#### 2.2.3 Eigenspektrum

Das Eigenspektrum beschreibt die Eigenzustände und deren Eigenwerte zu einem bestimmten Zeitpunkt oder Zeitspanne. Im AQC oft auch Energiespektrum genannt, setzt es die entsprechenden Energiezustände und Eigenenergien eines Hamiltonians zueinander in Verbindung. In Abbildung 2.2 ist ein beliebiges Eigenspektrum dargestellt, dessen Grundzustand durch die unterste Linie und die angeregten Zustände durch die oberen Linien beschrieben werden. Die rote Markierung kennzeichnet die minimale Spektrallücke. Diese wird durch den geringsten Abstand zwischen dem Grundzustand und dem ersten angeregten Zustand (derjenige mit der geringsten Energie im Vergleich zum Grundzustands) während des gesamten adiabatischen Prozesses definiert,

$$\delta_{\min} = \min_{0 \le t \le t_f; \ j \ne 0} \ [E_j(t) - E_0(t)].$$
(2.5)

 $E_j(t)$  ist die Energie eines beliebigen angeregten Zustands und  $E_0(t)$  die des Grundzustands zum Zeitpunkt t. Durch die Berechnung aller Energiezustände und ihrer entsprechenden Eigenenergien kann das Energiespektrum eines (relativ kleinen) Hamiltonians analysiert und seine minimale Spektrallücke abgeschätzt werden. Allerdings werden unterschiedliche Probleme auch durch verschiedene Hamiltonians beschrieben und haben daher ein anderes entsprechendes Eigenspektrum. Laut dem Hardwarehersteller D-Wave Systems sind die schwierigsten Probleme in Bezug auf das AQC und die davon inspirierte Quantum Annealing Heuristik diejenigen mit den kleinsten Spektrallücken [47].

Zu erwähnen ist, dass bei einem sogenannten Zustandsübergang, eine Situation bei der  $\delta_{min} = 0$ , das adiabatische Theorem nicht gilt. In diesem Falle kann es passieren, dass der Grundzustand verlassen wird und in einen angeregten Zustand springt, was sich beim Messen des Problem-Hamiltonians in einer nicht optimalen Lösung widerspiegelt. Um die Wahrscheinlichkeit des Auftretens eines Zustandsübergangs zu verringern, wird der initiale Hamiltonian so gewählt, dass dessen Basis orthogonal zu der des Problem-Hamiltonians ist und diese somit nicht kommutieren [112].

Das Eigenspektrum von Hamiltonians und deren minimale Spektrallücke bilden eine wichtige Grundlage für die Optimierung von Hamiltonians und sind somit für die inhaltlichen Kapitel 4 und 5 dieser Arbeit von großer Bedeutung.

## 2.3 Quantum Annealing

Quantum Annealing (QA) ist ein heuristischer Ansatz zur Lösung von kombinatorischen Optimierungsproblemen, der ursprünglich für die Implementierung auf konventionellen Computern entwickelt wurde. Er kann als eine Variante zu der bekannteren klassischen *Simulated Annealing* (SA) Metaheuristik betrachtet werden. Der Grundgedanke eher Quanten- als thermische Fluktuationen für einen Optimierungsalgorithmus zu verwenden wurde bereits 1988 in [7, 134] als quanten-inspirierter klassischer Algorithmus vorgestellt. Unabhängig davon wurden in den darauf folgenden Jahren ähnliche QA-Verfahren publiziert [65, 98, 181].

Nachfolgend wird der allgemeine Ablauf des Quantum Annealing Algorithmus nach [98] beschrieben. QA ist eine heuristische Suche, die sich sinnbildlich über die Lösungslandschaft eines Optimierungsproblems bewegt und nach tief liegenden Bereichen (lokalen beziehungsweise globalen Minima) sucht. Wie in Kapitel 2.2.1 erläutert, kann die Zielfunktion f(x) eines Problems Pin einen Problem-Hamiltonian  $\mathcal{H}_P$  kodiert werden. Mit einem weiteren so-



Abbildung 2.2: Vereinfachte Darstellung eines beliebigen Eigenspektrums mit vier Zuständen, dessen Grundzustand durch die unterste Linie beschrieben ist und die angeregten Zustände darüber. Der rote Kreis markiert die minimale Spektrallücke zwischen dem Grundzustand und dem ersten angeregten Zustand über den gesamten Evolutionspfad  $t: 0 \rightarrow t_f$ . Die Abbildung basiert auf [47].

genannten Transversalfeld-Hamiltonian (auch Disordering-Hamiltonian)  $\mathcal{H}_D$ wird dann durch die Lösungslandschaft von  $\mathcal{H}_P$  gesteuert. Der Transversalfeld-Hamiltonian wird dabei mit einem Transversalfeldkoeffizienten  $\Gamma(t)$  skaliert, der zu Beginn des Annealing Prozesses mit einem hohen Wert initialisiert wird. Über die Zeit wird  $\Gamma(t)$  dann schrittweise auf 0 reduziert. Dieser Prozess ist über den zeitabhängigen Hamiltonian,

$$\mathcal{H}(t) = \Gamma(t)\mathcal{H}_D + \mathcal{H}_P, \qquad (2.6)$$

definiert. [112]

Der Transversalfeld-Hamiltonian  $\mathcal{H}_D$  führt dem Annealing Prozess kinetische Energie in Form von Quantenfluktuationen zu. Eine stetige Verringerung von  $\Gamma(t)$  schwächt die Quantenfluktuationen ab und bringt das System näher an den Problem-Hamiltonian  $\mathcal{H}_P$  [112]. In Bezug auf die Lösungslandschaft, ermöglicht ein großes  $\Gamma(t)$  lokalen Minima zu Beginn des Annealing Prozesses zu entkommen, indem durch Hügel der Lösungslandschaft getunnelt werden kann. So wird letztendlich dem Algorithmus ermöglicht, größere Bereiche des Lösungsraums zu einem frühen Zeitpunkt im Prozess zu durchsuchen, bevor er zu einem lokalen oder globalen Minimum konvergiert.

Die Ähnlichkeit zum AQC (siehe Gleichung (2.1) ist offensichtlich. Beide Algorithmen verwenden einen Problem-Hamiltonian, um das zu lösende Problem zu kodieren. Der orthogonale Hamiltonian  $\mathcal{H}_D$  des QA führt dem System Quantenfluktuationen zu, um der heuristischen Suche zu ermöglichen, lokalen Minima zu entkommen; dies ist analog zu dem initialen Hamiltonian  $\mathcal{H}_I$  des AQC-Modells, der notwendig ist, um zu Beginn des adiabatischen Prozesses eine gleichverteilte Superposition zu erzeugen. Die Funktion  $\Gamma(t)$  des QA-Prozesses erzeugt zudem eine bestimmte Art von adiabatischer Evolution. Da Quantum Annealing sowohl mit Methoden des AQC als auch mit klassischen Computern realisiert bzw. simuliert werden kann, bildet es die Brücke zwischen klassischer Optimierung und adiabatischen Quantum Computing [112]. Fälschlicherweise wird QA dementsprechend oftmals als Synonym für das AQC verwendet. Doch während das AQC-Modell in der Lage ist, jeden Algorithmus des universellen Quantenschaltkreismodells (mit einem maximal polynomialen Anstieg in Rechenzeit) zu simulieren und somit ebenfalls universell ist [3], ist QA eher für die Anwendung auf kombinatorische Optimierungsprobleme konzipiert und lediglich vom theoretischen AQC-Modell inspiriert. Der größte Unterschied zum AQC ist, dass beim QA nicht unbedingt die gesamte Berechnung im Grundzustand stattfinden muss [112].

#### 2.3.1 Optimierungsprobleme

Wie bereits in den vorherigen Abschnitten angedeutet, ist das AQC bzw. die QA-Heuristik besonders für das Lösen von Optimierungsproblemen geeignet. Deshalb soll vorerst eine allgemeine Definition von Optimierungsproblemen gegeben werden, bevor im Anschluss deren Transformation zu Problem-Hamiltonians diskutiert wird.

Das Ziel eines Optimierungsproblems ist die beste Lösung aus einem gegebenen Lösungsraum zu finden. Es gibt drei Elemente die bei der Lösung eines Optimierungsproblems von wesentlicher Bedeutung sind: eine Zielfunktion, Variablen, und Nebenbedingungen. Jede Variable kann verschiedene Werte annehmen, und das Ziel ist es, den optimalen Wert für jede Variable zu finden, sodass die Zielfunktion minimiert wird. Optimierungsprobleme können weiterhin in Bezug auf ihre Variablen in diskrete oder kontinuierliche Probleme unterteilt werden. Während bei den Optimierungsproblemen mit diskreten Variablen meist Lösungen in Form von ganzen Zahlen oder Permutationen aus einer abzählbaren Menge gefunden werden müssen, wird bei den Problemen mit kontinuierlichen Variablen prinzipiell ein optimaler Wert zu einer kontinuierlichen Funktion gesucht. Zudem ist bei Optimierungsproblemen zu unterscheiden, ob diese Nebenbedingungen besitzen, d.h. bei denen Beschränkungen für die Variablen gelten, oder ob es sich um uneingeschränkte Probleme ohne Restriktionen auf die Variablen handelt.

In dieser Arbeit wird sich auf diskrete (kombinatorische) Optimierungsprobleme mit Nebenbedingung beschränkt, die mathematisch wie folgt definiert sind [19]:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.7}$$

gemäß

$$h(x) = 0 \tag{2.8}$$

$$g(x) \le 0 \tag{2.9}$$

Hierbei beschreibt Formel (2.7) die Zielfunktion  $f : \mathbb{D}^n \to \mathbb{R}$ , die auf n Variablen  $x = x_1, ..., x_n$  aus einer diskreten Domäne  $\mathbb{D}$  definiert ist. Ziel ist es, Werte für die Variablen x zu finden, sodass f(x) minimiert wird. <sup>1</sup> Zudem definieren die Gleichung und Ungleichung aus (2.8) und (2.9) jeweils Nebenbedingungen, die die jeweiligen zulässigen Werte für die Variablen x ebenfalls einschränken. Viele interessante und hoch komplexe Problemstellungen aus der Industrie und Wissenschaft lassen sich als eingeschränkte Optimierungsprobleme auffassen [117, 121, 137]. Oft sind die entsprechenden Lösungslandschaften nicht konvex und haben viele lokale Minima. In Abbildung 2.3 ist eine entsprechende nichtkonvexe Lösungslandschaft eines beliebigen Optimierungsproblems visualisiert.



Abbildung 2.3: Darstellung einer nicht-konvexen Lösungslandschaft eines beliebigen Optimierungsproblems mit einem globalen Minimum und mehreren lokalen Minima.

#### 2.3.2 Ising-Modell

Das Ising-Modell ist ein mathematisches Modell, das zur Beschreibung von Phasenübergängen und anderen energetischen Eigenschaften physikalischer

<sup>&</sup>lt;sup>1</sup>Entsprechend gilt, dass sich aufgrund des im Dualitätsprinzip der Optimierung [78] geforderte Zusammenhang min  $f(x) = -\min - f(x)$  jedes Minimierungs- in ein Maximierungsproblem umwandeln lässt und vice versa.

Systeme, die sich über die Zeit entwickeln, verwendet wird. Es beschreibt die Interaktion von n Teilchen, die auf den Knoten eines Graphen in Form eines ddimensionalen Gitters, angeordnet sind. Jedes Teilchen kann sich in einem von zwei Zuständen, sogenannten Spins  $\{s_i | 1 \leq i \leq n\}$  befinden, die als  $\pm 1$  dargestellt werden. Die externen und interagierenden Kräfte, die auf bzw. zwischen den benachbarten Teilchen in dem physikalischen System wirken, werden mit Variablen  $h_i$  und  $J_{ij}$  modelliert. Das Ising-Modell ermöglicht somit verschiedene Spinkonfigurationen  $s = s_1...s_n$  und deren energetischen Eigenschaften in einem physikalischen System auszudrücken. Mathematisch wird die Energie einer klassischen Ising-Spinkonfiguration folgendermaßen definiert,

$$H(s) = \sum_{i} h_{i} s_{i} + \sum_{i < j} J_{ij} s_{i} s_{j}.$$
 (2.10)

Ersetzt man die Spinvariablen  $s_i$ , die die Werte  $\pm 1$  annehmen können, durch die Pauli-z Operatoren, die die Eigenwerte  $\pm 1$  besitzen, erhält man folgenden quantenmechanischen Ising-Hamiltonian,

$$\mathcal{H} = \sum_{i} h_i \sigma_z^{(i)} + \sum_{i>j} J_{i,j} \sigma_z^{(i)} \sigma_z^{(j)}.$$
(2.11)

Ising-Hamiltonians spielen eine zentrale Rolle im Quantum Computing, da sie zur Problembeschreibung der Hamiltonians in den AQC- und QA-Algorithmen verwendet werden können. Die Pauli-z Operatoren  $\sigma_z^{(i)}$  des Ising-Hamiltonians aus Gleichung (2.11) werden gewissermaßen auf die einzelnen Qubits  $\{q_i|1 \leq i \leq n\}$  des AQC- oder QA-Algorithmus angewendet, die in diesem Fall ebenfalls in einem *d*-dimensionalen Gitter mit ihren direkt benachbarten Qubits energetisch interagieren können. Es ist bekannt, dass Ising-Hamiltonians NPschwer sind [10] und folglich Probleme, die sich als Ising-Hamiltonian abbilden lassen, ebenfalls komplexitätstheoretisch in die NP-schweren Probleme einzuordnen sind.

Der Vollständigkeit halber sei erwähnt, dass es eine alternative Formulierung zum Ising-Modell gibt, die abhängig von der Problemstellung häufig vorteilhafter für die entsprechende Problemformulierung sein kann. Die *Quadratic Un*constrained Binary Optimization (QUBO) ist mathematisch äquivalent zum Ising-Hamiltonian und kann durch eine einfache Transformation entwickelt werden, indem jeder Pauli-z Operator  $\sigma_z^{(i)}$  durch  $2x_i - 1$  ersetzt wird. Wobei  $x_i$  eine binäre Variable ist [23, 161]. QUBO-Probleme besitzen folgende Form [102],

$$E(x) = x^T Q x. (2.12)$$

Hier entspricht  $x \in \{0, 1\}^n$  einem Vektor der Größe n und  $Q \in \mathbb{R}^{n \times n}$  einer symmetrischen  $n \times n$  reellwertigen Matrix, die die Wechselwirkungen (analog zu den externen und interagierenden Kräfte des Ising-Modells) zwischen den binären Variablen des Vektors beschreibt. Bei gegebener Matrix Q wird für viele Problemstellungen die binäre Variablenzuordnung x gesucht, die den Term aus (2.12) minimiert.

In [23, 75, 108, 164] sind zahlreiche Ising-Hamiltonians und QUBO-Formulierungen für viele verschiedene Optimierungsprobleme definiert.

#### 2.3.3 Eingeschränkte Ising-Hamiltonians

Zahlreiche Problemstellungen aus der Industrie und Wissenschaft lassen sich als Optimierungs- bzw. Suchprobleme abbilden [40, 128, 177]. Meist müssen dabei verschiedene Bedingungen berücksichtigt werden, die den Lösungsraum zusätzlich begrenzen und die Variablen beschränken. Das in Kapitel 2.3.2 vorgestellte Ising-Modell, zur Formulierung von Optimierungsproblemen, enthält keine anderen Beschränkungen, als dass die Spins entsprechend die Werte  $\pm 1$  annehmen müssen. Um zusätzliche Nebenbedingungen der Domäne in ein Ising-Modell zu kodieren, können quadratische Penalty-Terme in die Zielfunktion eingebunden werden. Sollten diese Penalty-Terme bzw. Nebenbedingungen höhergradige (> 2) Interaktionen repräsentieren, müssen zusätzliche Spins eingeführt werden, um diese höhergradigen Interaktionen auf mehrere paarweise Interaktionen des klassischen Ising-Modells zu reduzieren [8, 18]. Die quadratischen Penalty-Terme sind dann so formuliert, dass sie für machbare Lösungen gleich Null und für nicht machbare Lösungen gleich einem positiven Penalty-Parameter sind. Wenn die Penalty-Terme eines Minimierungsproblems erfüllt und somit auf Null gebracht werden können, wird die ursprüngliche uneingeschänkte Zielfunktion minimiert, andernfalls würde der Penalty-Parameter das Energieniveau der entsprechenden Lösung anheben und diese als invalide beziehungsweise suboptimal kennzeichnen.

Beispiele für Penalty-Terme die häufig Verwendung finden, sind in Tabelle 2.1 aufgeführt. Dabei sind alle Variablen in der Tabelle binär. Im Allgemeinen wird in dieser Arbeit für die Spinvariable  $s_i$  der Ising-Hamiltonians deren komplementäre binäre Bitvariable  $x_i$  mit  $x_i \equiv \frac{s_i+1}{2}$  verwendet. In der Regel und vom Problem abhängig ist es oftmals intuitiver, die Terme eines Ising-Hamiltonians mit der entsprechenden binären Bitvariable zu formulieren. Der Penalty-Parameter A ist ein positiver Skalar. Dieser Wert muss ausreichend groß gewählt werden, um sicherzustellen, dass der Penalty-Term tatsächlich der klassischen Bedingung entspricht. Mehr zur Wahl der entsprechenden Penalty-Parameter wird in den Kapiteln 2.3.3.1-2.3.3.6 an verschiedenen eingeschränkten Ising-Hamiltonians beschrieben.

Um die grundlegende Idee einer Nebenbedingung zu veranschaulichen, wird folgendes eingeschränktes Problem der Form,

 $\min y = f(x)$ mit der Nebenbedingung  $x_1 + x_2 \le 1$ ,

betrachtet. Wobei  $x_1$  und  $x_2$  binäre Variablen sind. Die Bedingung definiert, dass entweder eine oder keine der beiden Variablen gewählt werden kann. Sie

Klassische Nebenbedingung	Penalty-Term
$x + y \le 1$	$A \cdot (x \cdot y)$
$x + y \ge 1$	$A \cdot (1 - x - y + x \cdot y)$
x + y = 1	$A \cdot (1 - x - y + 2 \cdot x \cdot y)$
$x \leq y$	$A \cdot (x - x \cdot y)$
$x + y + z \le 1$	$A \cdot (x \cdot y + x \cdot z + y \cdot z)$

Tabelle 2.1: Auflistung bekannter klassischer Nebenbedingungen und ihr entsprechender Penalty-Term. [75]

schließt ausdrücklich aus, dass beide gewählt werden können (d.h., beide können nicht auf 1 gesetzt werden). Aus der ersten Zeile der obigen Tabelle 2.1 geht hervor, dass der Penalty-Term, der der Nebenbedingung entspricht, wie folgt lauten muss  $A \cdot (x_1 \cdot x_2)$ . Wenn A ein ausreichend großer positiver Skalar ist, erhält man folgendes eingeschränktes Problem,

$$\min y = f(x) + A \cdot (x_1 \cdot x_2),$$

das die gleiche optimale Lösung wie das uneingeschränkte Problem besitzt. Wenn f(x) nun linear oder quadratisch ist, dann hat dieses uneingeschränkte Problem die Form eines QUBO-Modells. In dem vorliegenden Beispiel wird jeder Optimierer, der versucht, f(x) zu minimieren, versuchen, Lösungen zu vermeiden, bei denen sowohl  $x_1$  als auch  $x_2$  gleich 1 sind, da sonst der positive Penalty-Parameter A zur Zielfunktion hinzugefügt wird. D.h., die Lösungen bei denen die Bedingungen nicht eingehalten werden, werden durch den addierten Penalty-Parameter und die daraus resultierenden erhöhten Kosten für die Lösung, als invalide deklariert.[75]

In den nachfolgenden Unterkapiteln werden die in dieser Arbeit verwendeten Optimierungsprobleme und deren Formulierung als Hamiltonian vorgestellt. Um unterschiedlich große Probleminstanzen auf aktueller NISQ-Hardware ausführen zu können, wird sich in dieser Arbeit auf akademische theoretische Problemstellungen mit jeweils einer Nebenbedingung und deren Variablen meist linear zur Problemgröße skalieren, beschränkt.

#### 2.3.3.1 Minimum Exact Cover Problem

Das Minimum Exact Cover Problem (MECP) ist ein diskretes kombinatorisches Optimierungsproblem bei dem eine Menge  $U = \{1, ..., n\}$  und eine Menge V von nicht-leeren Teilmengen  $V_i \subseteq U$  gegeben ist. Mit i = 1, ..., N und  $U = \bigcup_i V_i$ . Die Aufgabe besteht darin, die kleinste Teilmenge R der Menge  $V_i$ zu finden, für die die Elemente von R disjunkte Mengen sind und die Vereinigung der Elemente von R gleich U ist. Der eingeschränkte Hamiltonian ist in [108] angegeben,

$$\min B \sum_{i} x_i + A \sum_{\alpha}^{n} \left( 1 - \sum_{i:\alpha \in V_i} x_i \right)^2.$$
(2.13)

Dabei bezeichnen die Indices  $\alpha$  die Elemente von U und i die Teilmengen  $V_i$ . Der zweite Term, der die Nebenbedingung des Optimierungsproblems darstellt, ist genau dann Null, wenn jedes Element von U genau einmal enthalten ist. Das bedeutet dann, dass die Teilmengen  $V_i$  von R disjunkt sind und ihre Vereinigung jedes Element von U enthält. Der erste Term, der die Zielfunktion darstellt, zählt die Anzahl der Teilmengen der jeweiligen Lösung. Der Eigenwert des Grundzustands dieses Hamiltonians beträgt  $m \cdot B$ , wobei m die kleinste Anzahl von Teilmengen darstellt, die für die vollständige Vereinigung erforderlich ist. Für die Wahl der Penalty-Parameter A und B muss der schlechteste Fall betrachtet werden, der dennoch eine valide Lösung für die Problemstellung darstellt. D.h., die Lösung bestünde aus eine kleinen Anzahl von Teilmengen mit jeweils einem einzigen nicht disjunkten Element, deren Vereinigung wiederum U ist. Um sicherzustellen, dass dies nicht eintritt, müssen die Penalty-Parameter folgendermaßen festlegt werden

$$n \cdot B < A. \tag{2.14}$$

Die Anzahl der Variablen bzw. logischen Qubits, die für den Hamiltonian benötigt werden, skaliert linear mit der Anzahl der Teilmengen |V|.

#### 2.3.3.2 Set Packing Problem

Im Rahmen des Set Packing Problems (SPP) sind wiederum eine Menge  $U = \{1, ..., n\}$  und Teilmengen  $V_i \subseteq U$  mit i = 1, ..., N gegeben. Das Ziel ist es nun die maximale Anzahl an Teilmengen  $V_i$  zu finden, die alle disjunkt sind, d.h. deren Teilmengen keine gemeinsamen Elemente besitzen. In [108] ist der folgende Hamiltonian gegeben,

$$\min -B\sum_{i} x_i + A\sum_{i,j:V_i \cap V_j \neq \emptyset} x_i x_j.$$
(2.15)

Der zweite Term, der die Nebenbedingung beinhaltet, wird nur minimiert, wenn alle Teilmengen disjunkt sind. D.h., wenn der Schnitt der paarweisen Teilmengen  $V_i \cap V_j \neq \emptyset$  ist. Der erste Term, der die Zielfunktion beschreibt, zählt die Anzahl der Teilmengen in der Lösung. Durch die Wahl der Penalty-Parameter,

$$B < A, \tag{2.16}$$

wird sicher gestellt, dass die Nebenbedingung nie verletzt wird. Das SPP erfordert, wie das MECP, ebenfalls |V| logische Variablen zur Beschreibung des Hamiltonians.
### 2.3.3.3 Minimum Vertex Cover Problem

Das Minimum Vertex Cover Problem (MVCP) ist ein graphbasiertes Optimierungsproblem. Gegeben ist ein Graph G = (V, E) mit Knoten V und Kanten E. Gesucht wird die kleinste Menge an Knoten, die mindestens einen Endpunkt jeder Kante des Graphen enthält. Nachfolgend ist der Hamiltonian für das MVCP gegeben [75],

$$\min B \sum_{i=0}^{n} x_i + A \sum_{(i,j)\in E} 1 - x_i - x_j + x_i x_j.$$
(2.17)

Die binäre Variable  $x_i = 1$ , wenn der Knoten  $v_i$  in der gesuchten minimalen Menge ist und andernfalls  $x_i = 0$ . Während der erste Term des Hamiltonians die Zielfunktion darstellt, d.h. die Anzahl der Knoten in der Lösung beziehungsweise Menge zählt, stellt der zweite Term die Nebenbedingung sicher, dass jede Kante im Graphen mit mindestens einem Knoten aus der minimalen Menge an Knoten verbunden ist. Die Anzahl der Variablen bzw. logischen Qubits skaliert linear mit |V|. Um gültige Lösungen zu gewährleisten, müssen die Penalty-Parameter folgendermaßen gewählt werden

$$B < A. \tag{2.18}$$

### 2.3.3.4 Maximum Clique Problem

Eine Clique ist eine Teilmenge von Knoten eines Graphen, bei der alle Knoten direkt miteinander verbunden sind. Das *Maximum Clique Problem* (MCP) sucht dementsprechend nach der größten Clique eines Graphen G = (V, E) mit Knoten V und Kanten E. In Anlehnung an [36] ist der Hamiltonian gegeben als,

$$\min -B\sum_{i=0}^{n} x_i + A\sum_{(i,j)\in \bar{E}} x_i x_j.$$
(2.19)

Dabei ist  $x_i = 1$ , wenn der Knoten  $v_i$  in der Clique enthalten ist und andernfalls  $x_i = 0$ . Während der erste Term die Zielfunktion darstellt, d.h. die Anzahl der Knoten in der Clique zählt, stellt der zweite Term die Nebenbedingung sicher, dass keine Kante der Komplementärmenge der Kanten  $\overline{E}$  in der Clique enthalten ist. Die Anzahl der benötigten Variablen skaliert linear mit der Anzahl der Knoten |V| im Graphen. Um gültige Lösungen zu garantieren, müssen die Penalty-Parameter entsprechend,

$$B < A \tag{2.20}$$

gewählt werden.

#### 2.3.3.5 Knapsack Problem

Das Knapsack Problem (KP) ist ein kombinatorische Optimierunsproblem, bei dem n Gegenstände, die jeweils ein Gewicht  $w_i \in \mathbb{N}$  und ein Wert  $c_i \in \mathbb{N}$  haben, gegeben sind. Ziel des Optimierungsproblems ist es, die Gegenstände so auszuwählen, dass deren Gesamtgewicht kleiner oder gleich einer Rucksackkapazität W ist und die Summe der entsprechenden Gegenstandswerte maximiert wird. Der Hamiltonian ist in [108] gegeben,

$$\min -B\sum_{i=0}^{n} c_i x_i + A\left(1 - \sum_{j=0}^{W} y_j\right)^2 + A\left(\sum_{j=0}^{W} j y_j - \sum_{i=0}^{n} w_i x_i\right)^2.$$
 (2.21)

Dabei ist die Binärvariable  $y_j = 1$  für  $1 \le j \le W$ , wenn das summierte Gewicht des Rucksacks j ist, und andernfalls  $y_j = 0$ . Außerdem ist die Variable  $x_i = 1$ , wenn der Gegenstand mit dem Index i Teil der Lösung ist, und andernfalls  $x_i = 0$ . Der zweite und der dritte Term erzwingen, dass das Gewicht nur genau einen Wert annehmen kann und dass das summierte Gewicht der Gegenstände im Rucksack dem angegebenen Wert aus dem zweiten Term entspricht. Der erste Term, der die Zielfunktion darstellt, summiert die Werte der Gegenstände im Rucksack auf. Die Penalty-Parameter A und B werden folgendermaßen gewählt,

$$B \cdot \sum_{i=0}^{n} c_i < A, \tag{2.22}$$

um Verstöße gegen die Gewichtsbeschränkung zu bestrafen. Die Anzahl der benötigten Variablen ist n + W. Zudem muss angemerkt werden, dass der hier angegebene Bereich der Penalty-Parameter für das KP von dem in [108] abweicht. In einer nachfolgenden Veröffentlichung wurde das Parameterintervall für das KP überholt [135].

### 2.3.3.6 Binary Integer Linear Problem

Im Rahmen des Binary Integer Linear Problem (BILP) ist ein Vektor x mit binären Variablen  $x = (x_1..., x_N)$  gegeben. Beim BILP ist dann das größte Skalarprodukt  $c \cdot x$  für einen ganzzahligen Vektor c unter der Bedingung  $S \cdot x = b$ gesucht, wobei S eine  $m \times N$  Matrix und b einen Vektor mit m Komponenten darstellt. Der entsprechende Hamiltonian ist wie in [108] gegeben,

$$\min - B \sum_{i=1}^{N} c_i x_i + A \sum_{j=1}^{m} \left( b_j - \sum_{i=1}^{N} S_{ji} x_i \right)^2.$$
 (2.23)

Während der zweite Term die Nebenbedingung  $S \cdot x = b$  erzwingt, maximiert der erste Term das Skalarprodukt der Vektoren c und x. In dem Fall, dass die Koeffizienten  $S_{ij}$  und  $c_i$  ganzzahlig sind, werden die Penalty-Parameter folgendermaßen gewählt,

$$B \cdot N < A. \tag{2.24}$$

Die Anzahl der benötigten Variablen skaliert linear mit der Größe der Vektoren c bzw. x. Weitere Details und Varianten des BILP, die in dieser Arbeit nicht betrachtet werden, sind in [108] zu finden.

# 2.3.4 Implementierung in Hardware

In den Kapiteln 2.2 und 2.3 wurden die theoretischen Grundlagen für das adiabatische Quantum Computing und die verwandte Quantum Annealing Heuristik geliefert. Seit 2011 vertreibt die kanadische Firma D-Wave Systems kommerzielle Quantum Computing Hardware, die auf dem Quantum Annealing Paradigma fundiert [48]. Mit dem Hintergrund, dass die verschiedenen Hamiltonians, die in dieser Arbeit entwickelt und optimiert werden, unter anderem auf den D-Wave Quantum Annealern angewendet und evaluiert werden, wird in diesem Abschnitt die grundlegende Funktionsweise der Hardware vorgestellt.

Die Quantenprozessoren (engl. Quantum Processing Unit (QPU)) von D-Wave sind so konzipiert, dass sie Instanzen des Ising-Modells, durch die physikalische Umsetzung eines Quantum Annealing Algorithmus lösen können. Das Ising-Modell, vgl. Gleichung (2.11), ist dabei direkt in der Hardware implementiert: Die Gewichte  $h_i$  und  $J_{ij}$  des Modells entsprechen den elektronischen Signalen, den sogenannten Biases, die auf die Quantenteilchen in Form von Qubits und deren Kopplungen (engl. Couplers) zwischen Qubit-Paaren, wirken. In Bezug auf die *D-Wave 2000Q* Hardware, die im Rahmen dieser Arbeit zur Evaluation der Algorithmen und Hamiltonians verwendet wurde, sind dessen 2048 Qubits und 6016 physischen Kopplungen als Chimera-Graph  $C_k$  mit k = 16 angeordnet. Dabei entspricht die Notation  $C_k$  einem Chimera-Graphen C der aus einem  $k \times k$  Gitter von Chimera-Zellen besteht. Die Chimera-Zellen umfassen wiederum jeweils 8 Qubits. In Abbildung 2.4 ist ein entsprechender Chimera-Graph  $C_3$  dargestellt.

Die Architektur und Konnektivität der Qubits spielen eine wesentliche Rolle bei der Lösung der Ising-Instanzen, da die Gewichte des Ising-Modells sich auf die physikalischen Qubits und deren Kopplungen auf der QPU abbilden lassen müssen. Das heißt, der logische Graph einer Ising-Instanz muss auf den physischen Chimera-Graphen der QPU abbildbar sein. Die zugrundelegende Problemstellung, ob ein beliebiger Graph A einen anderen beliebigen Graph B als Minor enthält ist NP-schwer [70]. D-Wave liefert dem Anwender ihrer Hardware hierfür eine Heuristik, die dieses sogenannte *Minor-Embedding* übernimmt und sich für spärliche Graphen mit vielen Knoten als effektiv erwiesen hat [29]. Für Ising-Instanzen, deren inhärente Graphen sich aufgrund der Konnektivität nicht direkt in den Hardware Graphen abbilden lassen, werden Qubit-Ketten zur Hilfe genommen. Das heißt, eine logische Variable des Ising-Modells wird durch mehrere auf der Hardware gekoppelte Qubits repräsentiert.



Abbildung 2.4: Visualisierung eines  $3 \times 3$  Chimera-Graphen  $C_3$ , in dessen 9 Chimera-Zellen jeweils 8 Qubits als bipartiter Graph angeordnet sind. Abbildung basiert auf [32].

Diese Qubits innerhalb einer Kette müssen entsprechend gewichtet<sup>2</sup> werden, damit sie bei anschließender Messung in die gleiche Basis kollabieren. D-Wave verwendet zusätzliche klassische Postprocessing-Methoden, um Fehler in den Ketten zu mitigieren und Lösungen, bei denen Qubits einer Kette nach der Messung unterschiedliche Werte aufweisen, zu reparieren. Die standardmäßig voreingestellte Methode ist die Mehrheitsentscheidung [104].

Der in Kapitel 2.3 beschriebene Quantum Annealing Prozess wird in der D-Wave Hardware durch den zeitabhängigen Hamiltonian  $\mathcal{H}(t)$  in Gleichung (2.25) realisiert.

$$\mathcal{H}(t) = \underbrace{-\frac{\mathcal{A}(t)}{2} \left(\sum_{i} \sigma_{x}^{(i)}\right)}_{\mathcal{H}_{I}} + \underbrace{\frac{\mathcal{B}(t)}{2} \left(\sum_{i} h_{i} \sigma_{z}^{(i)} + \sum_{i>j} J_{i,j} \sigma_{z}^{(i)} \sigma_{z}^{(j)}\right)}_{\mathcal{H}_{P}}.$$
 (2.25)

 $\mathcal{A}(t)$  und  $\mathcal{B}(t)$  sind die Annealing-Funktionen von D-Wave Systemen, wobei  $\mathcal{A}(t)$  die Transversalfeldenergie und  $\mathcal{B}(t)$  die Energie des Problem-Hamiltonians zum Zeitpunkt t in Joule beschreibt. Die Annealing-Funktionen müssen  $\mathcal{B}(t=0) = 0$  und  $\mathcal{A}(t=t_f) = 0$  erfüllen, wobei  $t_f$  die gesamte Ent-

<sup>&</sup>lt;sup>2</sup>D-Wave Systems liefert hierfür einen *Chain Strength* Parameter der größer als der Betrag des maximalen Koeffizienten des Ising-Modell gesetzt werden sollte, um Qubits einer Kette gleichzuschalten. Da jedoch Qubits und Kopplungen nur eine gewisse Hardwarespezifische Präzision haben, kann ein zu groß gewählter *Chain Strength* Parameter das Intervall der möglichen Problemkoeffizienten einschränken, was letztendlich bei der Ausführung der Ising-Modell Instanzen zu Präzisionsfehlern führen kann [44]. Diese Thematik wird in Kapitel 5 nochmal aufgegriffen.



Abbildung 2.5: D-Wave 2000Q Systems Annealing Funktionen  $\mathcal{A}(t)$  und  $\mathcal{B}(t)$ . Der Annealing Prozess beginnt zum Zeitpunkt t = 0 mit  $\mathcal{A}(t) \gg \mathcal{B}(t)$  und endet zum Zeitpunkt  $t = t_f = 1$  mit  $\mathcal{A}(t) \ll \mathcal{B}(t)$ . Die Abbildung basiert auf [129].

wicklungszeit darstellt. Wenn sich die Zustandsentwicklung von t = 0 zu  $t = t_f$ ändert, führt der durch  $\mathcal{H}(t)$  beschriebene Annealing-Prozess zu der endgültigen Form des Problem-Hamiltonians, der dem zu minimierenden Ising-Problem entspricht. Daher entwickelt sich der Grundzustand des initialen Hamiltonians  $\mathcal{H}(0) = \mathcal{H}_I$  zum Grundzustand des Problem-Hamiltonians  $\mathcal{H}(t_f) = \mathcal{H}_P$ . Die zum Zeitpunkt  $t_f$  durchgeführten Messungen liefern dann energetisch niedrige Zustände des Ising-Hamiltonians, wie in Gleichung (2.11) angegeben. In Abbildung 2.5 sind zur Veranschaulichung des Quantum Annealing Prozesses die Annealing-Funktionen der D-Wave 2000Q Hardware abgebildet. Abschließend ist zu erwähnen, dass D-Wave Quantum Annealing Hardware sowohl QUBO-Formulierungen, als auch Ising-Hamiltonians als Eingabe akzeptiert. Im Falle einer QUBO-Problemformulierung wird dieser, vor der Ausführung auf der Hardware, automatisch durch die D-Wave API, gemäß der in Kapitel 2.3.2 beschriebenen Transformation, in einen Ising-Hamiltonian überführt.

# 2.4 Quantenschaltkreismodell

In den nachfolgenden Unterkapiteln werden die Grundlagen und bekannten Quantum Computing Ansätze des universellen Quantenschaltkreismodells (engl. Quantum Circuit Model (QCM)), die für das Verständnis dieser Arbeit notwendig sind, vorgestellt. In Kapitel 2.1.1 wurde bereits gezeigt, wie Quanteninformation mittels Qubits kodiert werden kann und wie einzelne quantenmechanische Zustände zu einem Quantensystem zusammengeschalten werden können. Während beim adiabatischen Quantum Computing die Berechnung über einen zeitabhängigen Hamiltonian beschrieben wird, wird im Quantenschaltkreismodell die Berechnung mit Hilfe verschiedener Quantengatter durchgeführt.

## 2.4.1 Quantengatter des Quantenschaltkreismodells

In Kapitel 2.1.3 wurden bereits sowohl die grundlegenden Eigenschaften von Quantengattern, als auch die drei Pauli-Operatoren vorgestellt, die sich sowohl im AQC- als auch im QCM-Modell wiederfinden. Neben diesen grundlegenden Quantengattern gibt es zahlreiche weitere Ein- und Zwei-Qubit-Gatter [162], die Verwendung im QCM finden. Im nachfolgenden wird eine Auswahl an entsprechenden Ein- und Zwei-Qubit-Gattern vorgestellt, die für das Verständnis verschiedener Algorithmen in dieser Arbeit notwendig sind.

### 2.4.1.1 Rotations-Gatter

Die konventionellen Pauli-Operatoren bilden eine  $180^{\circ}$  Rotation um die jeweilige Achse auf der Bloch-Einheitskugel ab. Die Operatoren können aber auch mit beliebigen Rotationen  $\theta$  parametrisiert werden. Solche parametrisierten Gatter werden allgemein als *Rotations-Gatter* bezeichnet und haben folgende Form,

$$R_A(\theta) = e^{-iA\frac{\theta}{2}} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}A$$
(2.26)

wobei A der Pauli-Matrix x, y, z entspricht und I der Einheitsmatrix. Mit den in Kapitel 2.1.1 bereits erwähnten trigonometrischen Funktionen lassen sich folgende drei parametrisierbare Rotations-Gatter darstellen,

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

### 2.4.1.2 Hadamard-Gatter

Das *Hadamard-Gatter* (H-Gatter) ist eines der wichtigsten Ein-Qubit-Gatter. Es wird verwendet, um ein Qubit in eine Superposition zu bringen. Die dazugehörige unitäre Matrix lautet:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1\\ 1 & -1 \end{pmatrix} \tag{2.27}$$

Wird das Hadamard-Gatter auf die Standardbasis angewendet, so wird deutlich, dass sich der Folgezustand zwischen den beiden Basiszuständen  $|0\rangle$  und  $|1\rangle$ befindet, beziehungsweise beide Zustände annimmt [122]. Das Qubit befindet sich dann in einer sogenannten gleichverteilten Superposition.

$$H |0\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = |+\rangle$$
$$H |1\rangle = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = |-\rangle$$

Die Anwendung des Hadamard-Gatters kann auf der Bloch-Einheitskugel visualisiert werden als eine 90° Rotation um die y-Achse mit einer anschließenden 180° Drehung um die x-Achse. Wendet man ein Hadarmard-Gatter auf ein Qubit im Zustand  $|+\rangle$  an, folgt der Zustand  $|0\rangle$  (siehe Beispiel 2.4.1.2). Eine weitere Besonderheit ist, dass zwei nacheinander angewendete Hadamard-Gatter den Zustand nicht verändern, da HH = I gilt [122].

### 2.4.1.3 CNOT-Gatter

Das kontrollierte Nicht-Gatter (engl. Controlled NOT-Gate (CNOT-Gatter)) ist ein Zwei-Qubit-Gatter, das ein Qubit als Kontroll-Qubit und ein anderes als Ziel-Qubit als Input bekommt. Ist das kontrollierende Qubit im Zustand  $|1\rangle$ , so wird der Operator den Zustand des Ziel-Qubits umkehren [86, 122]. Die entsprechende unitäre Matrix des CNOT-Gatters lautet,

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$
 (2.28)

Im nachfolgende Anwendungsbeispiel wird das linke Qubit als Kontroll-Qubit und das rechte als Ziel-Qubit betrachtet.

$$CNOT |\psi\rangle = CNOT(\gamma_1 |00\rangle + \gamma_2 |01\rangle + \gamma_3 |10\rangle + \gamma_4 |11\rangle)$$
  
=  $\gamma_1 |00\rangle + \gamma_2 |01\rangle + \gamma_3 |11\rangle + \gamma_4 |10\rangle$ 

Die CNOT-Operation tauscht die Amplituden der Zustandsvektoren von  $|10\rangle$  und  $|11\rangle$ . Die entsprechende Matrix wird deswegen auch Permutationsmatrix genannt [86].

### 2.4.1.4 Toffoli-Gatter

Das Toffoli-Gatter (CCNOT-Gatter) ist ein universelles umkehrbares Drei-Qubit-Gatter [67]. So lässt sich jedes klassische Gatter (AND, OR, NOT) und deren möglichen Zusammensetzungen aus Toffoli-Gattern aufbauen [86]. Das Gatter ist auf drei Qubits anwendbar, zwei Kontroll- und ein Ziel-Qubit. Nur wenn sich beide Kontroll-Qubits im Zustand  $|1\rangle$  befinden, wird das NOT-Gatter auf das Ziel-Qubit ausgeführt.

Im nachfolgende Anwendungsbeispiel werden die ersten beiden Qubits als

Kontroll-Qubits und das letzte als Ziel-Qubit betrachtet.

$$CCNOT |\psi\rangle = CCNOT(\gamma_1 |000\rangle + \gamma_2 |001\rangle + \gamma_3 |010\rangle + \gamma_4 |011\rangle$$
$$\gamma_5 |100\rangle + \gamma_6 |101\rangle + \gamma_7 |110\rangle + \gamma_8 |111\rangle)$$
$$= (\gamma_1 |000\rangle + \gamma_2 |001\rangle + \gamma_3 |010\rangle + \gamma_4 |011\rangle$$
$$\gamma_5 |100\rangle + \gamma_6 |101\rangle + \gamma_8 |110\rangle + \gamma_7 |111\rangle)$$

Die CCNOT-Operation tauscht die Amplituden der Zustandsvektoren von  $|110\rangle$  und  $|111\rangle$ .

Wie bereits in Kapitel 2.1.3 erwähnt und am Toffoli-Gatter exemplarisch ersichtlich, kann jedes höherdimensionale Gatter aus einer Kombination von Einund Zwei-Qubit-Gattern mit Hilfe des Kronecker-Produkts erstellt werden. Somit lassen sich auch beliebige quantenmechanische Zustände durch eine Reihe von grundlegenden Quantengattern erzeugen. Diese Gatter werden als universelle Gatter bezeichnet. Das CNOT-Gatter zusammen mit allen Ein-Qubit-Gattern bildet eine solche universelle Menge [174].

# 2.4.2 Algorithmen des Quantenschaltkreismodells

In diesem Abschnitt werden zwei grundlegende QCM-Algorithmen vorgestellt, die als Basis für die Entwicklung neuer hybrider Ansätze in Kapitel 3 dienen. Im Allgemeinen kann der Ablauf eines Quantenalgorithmus standardmäßig in drei Schritte unterteilt werden. Zu Beginn des Algorithmus werden Qubits in einen initialen Zustand versetzt [86]. Daraufhin werden die Qubits mit Quantengattern manipuliert. Die Quantengatter können selbst komplexere Berechnungen sein und auf mehrere Qubits angewendet werden. Sie können jedoch auch auf Ein-Qubit-Operationen und CNOT-Operationen reduziert werden, wie im vorherigen Abschnitt erläutert [139]. Der Abschluss eines Algorithmus bildet die Messung eines oder mehrerer Qubits zu einer definierten Basis [122]. Quantenalgorithmen des QCMs werden grundsätzlich mit Quantenschaltkreisen visualisiert. Ein beispielhafter Quantenschaltkreis ist in Abbildung 2.6 dargestellt. Jede Leitung entspricht bei dieser Veranschaulichung einem Qubit [86]. Gatter werden durch Rechtecke gekennzeichnet. Da alle Berechnungen umkehrbar sein müssen, ist sichergestellt, dass die Anzahl der Eingabe-Qubits der der Ausgabe-Qubits entspricht.

Der Endzustand  $|x', y', z'\rangle$  in Abbildung 2.6 ergibt sich aus dem Tensorprodukt der einzelnen Gatter [86]. Für dieses Beispiel ist das,

$$|x',y',z'\rangle = (I_2 \otimes W \otimes I_2)(U \otimes V) |x,y,z\rangle.$$
(2.29)

Die Abbildung 2.7 zeigt eine Auflistung von Operationen auf Qubits und deren Darstellung in Quantenschaltkreisen.



Abbildung 2.6: Quantenschaltkreis mit drei Qubits und drei Gattern. Die horizontalen Linien repräsentieren die Qubits und die Rechtecke die Quantengatter, die auf die entsprechenden Qubits wirken. Die Abbildung basiert auf [86].



Abbildung 2.7: Darstellung von Operationen und der entsprechenden Gatter in einem Quantenschaltkreis. In Bezug auf das  $R_Z(\theta)$ -Gatter beschreibt Z die Rotationsachse und  $\theta$  den Rotationsparameter. Analog kann die X- oder Y-Achse als Rotationsachse verwendet werden. Das CNOT- und Toffoli-Gatter besitzen beide ein Ziel-Qubit ( $\oplus$ ) und ein bzw. zwei Kontroll-Qubits ( $\bullet$ ). Das  $\sqrt{X}$ -Gatter entspricht dem  $R_X(\frac{\pi}{2})$ -Gatter mit einer anderen globalen Phase und ist zudem ein physikalisch implementierbares Gatter der IBM Quantum Hardware.

## 2.4.2.1 Quantum Approximate Optimization Algorithm

Der Quantum Approximate Optimization Algorithm (QAOA) ist ein hybrider quanten-klassischer Algorithmus des Quantenschaltkreismodells, der vor allem für das Lösen beziehungsweise Approximieren von kombinatorischen Optimierungsproblemen bekannt ist [60]. Der Algorithmus fällt unter die Kategorie der variationellen Quantenalgorithmen (VQA) [34], die einen klassischen Optimierer in Kombination mit einem parametrisierbaren Quantenschaltkreis verwenden.

In Bezug auf den QAOA wird der initiale Zustand des Systems in eine gleichverteilte Superposition der Basiszustände gebracht und anschließend durch Anwendung der algorithmusspezifischen Gatter weiterentwickelt, mit dem Ziel den Zustand herzustellen, der die Kosten einer Funktion oder eines Problems minimiert. Diese Quantenevolution ist durch einen Quantenschaltkreis mit geringer Tiefe gekennzeichnet, der auch als *Ansatz* bezeichnet wird und die Ausführung auf NISQ-Hardware ermöglicht. Die geringe Quantenschaltkreistiefe wird durch parametrisierte Gatter erreicht, deren Parameter als Rotationswinkel aufgefasst werden können. Die optimale Quantenevolution kann dann durch das Variieren der Gatter-Parameter gefunden werden. In der variationellen Architektur des QAOA werden dabei die Parameter der Gatter mit Hilfe einer klassischen Methode über mehrere Iterationen optimiert.

In Abbildung 2.8 ist der Aufbau des QAOA beschrieben. Zu Beginn wird das Optimierungsproblem in einen Problem-Hamiltonian  $H_P$ , wie in Gleichung (2.11) beschrieben, kodiert und ein zusätzlicher Transversalfeld-Hamiltonian  $H_D = \frac{1}{2} \sum_i \sigma_i^x$  definiert, wobei  $\sigma_i^x$  den Pauli-Operator, der auf Qubit *i* wirkt, darstellt. Mit p ( $p \ge 1$ ) werden die Anzahl an Wiederholungen bzw. Runden definiert für die die Hamiltonians  $H_P$  und  $H_D$  abwechselnd innerhalb des Quantenschaltkreises angewendet werden. Zudem werden zufällig jeweils zwei Rotationswinkel,  $0 \le \beta \le \pi$  und  $0 \le \gamma \le 2\pi$ , pro Runde gewählt [92].

Das System wird dann zunächst mit Hilfe des Hadamard-Gatters in eine gleichverteilte Superposition der Basiszustände  $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$  transformiert und die Hamiltonians  $H_P$  und  $H_D$  werden alternierend mit den Parametern  $\gamma_p$  und  $\beta_p$  angewendet, um den Zustand

$$\left|\psi(\beta,\gamma)\right\rangle = V_p U_p \dots V_2 U_2 V_1 U_1 \left|\psi\right\rangle$$

zu präparieren. Dabei sind  $U_p = e^{-i\gamma_p H_P}$  und  $V_p = e^{-i\beta_p H_D}$  unitäre Operatoren des Problem- bzw. Transversalfeld-Hamiltonians. Diese Technik zur Zustandstransformation wird auch *Trotterisierung* bezeichnet und ist eine bekannte Methode zur Simulation von nicht kommutierenden Hamiltonians auf Quantencomputern [163]. Mit dieser Methode wird der Problem-Hamiltonian  $H_P$  für eine gewisse Zeit  $\gamma_p$  und der Transversalfeld-Hamiltonian  $H_D$  für einige Zeit  $\beta_p$  entwickelt. Anschließend wird der Zustand mehrere Male in der z-Basis gemessen, um einen Erwartungswert zu generieren. Dieser wird dann von einem klassischen Optimierer als Input verwendet, um die Parameter  $\beta, \gamma$  anzupassen, mit dem Ziel, die Grundzustandsenergie des Problem-Hamiltonians

$$\min \langle \psi(\beta,\gamma) | H_P | \psi(\beta,\gamma) \rangle$$

zu finden. Der Grundzustand entspricht dann dem globalen Optimum des zu lösenden Optimierungsproblems. Dieser iterative Prozess wird solange fortgesetzt, bis der klassische Optimierer konvergiert oder eine Lösung von akzeptabler Qualität gefunden wurde [157].



Abbildung 2.8: Übersicht des hybriden quanten-klassischen QAOA: Die Quantenevolution startet mit einer gleichverteilten Superposition der Basiszustände  $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$ , die durch die Anwendung von Hadamard-Gattern realisiert wird. Anschließend werden die Operatoren der Problem- und Transversalfeld-Hamiltonians  $H_P$  und  $H_D$  abwechselnd mit den entprechenden Rotationswinkeln  $\beta$  und  $\gamma$  p Runden auf den Quantenzustand  $|\psi\rangle$  angewendet, um den neuen Zustand  $|\psi(\beta,\gamma)\rangle$ zu präparieren. Abschließend werden mehrere Messungen  $\langle \psi(\beta,\gamma)| H_P |\psi(\beta,\gamma) \rangle$  genommen, deren Erwartungswert dann als Input für den klassischen Optimierer dient, um die Winkelparameter  $\beta, \gamma$  anzupassen. Dieses Verfahren wird über mehrere Iterationen ausgeführt, bis der Optimierer konvergiert oder eine Lösung mit ausreichender Qualität erzielt wurde.

Die Ähnlichkeit des QAOA zum AQC ist ersichtlich. Im Prinzip nutzt der QAOA die Trotterisierung, um die Evolution des AQC, vgl. Gleichung (2.1), zu approximieren. Es ist theoretisch bewiesen, dass mit zunehmender Anzahl an Runden p, d.h. bei  $p \to \infty$  und optimalen Rotationsparametern  $\beta, \gamma$  der Erwartungswert immer gegen den bestmöglichen Wert konvergiert,  $lim_{p\to\infty}min_{\beta,\gamma} \langle \psi(\beta,\gamma) | H_P | \psi(\beta,\gamma) \rangle$  [60]. Jedoch ist in der Praxis eine der größten Schwierigkeiten, genau diese optimalen Rotationsparameter, in einer meist hochdimensionalen, nicht-konvexen Lösungslandschaft mit vielen nicht entarteten lokalen Optima, zu finden [184].

## 2.4.2.2 Grover-Algorithmus

Der Grover-Algorithmus ist neben dem Shor-Algorithmus einer der bedeutendsten anwendungsbezogenen Algorithmen des Quantum Computings. Lov Grover zeigte mit seinem Verfahren, dass ein bestimmtes Element in einer unstrukturierten Datenbank mit N Elementen mit  $O(\sqrt{N})$  Operationen gefunden werden kann [79]. Im Gegensatz dazu würde ein klassischer Computer O(N) Operationen benötigen. Daher bietet der Grover-Algorithmus eine quadratische Beschleunigung gegenüber dem optimalen klassischen Algorithmus. Es wurde zudem gezeigt, dass keine Quanten-Turing-Maschine dies in weniger als  $O(\sqrt{N})$  Operationen erreichen kann und der Grover-Algorithmus in diesem Sinne optimal ist [15].

Während der ursprüngliche Grover-Algorithmus für die Suche in einer Datenbank vorgestellt wurde, können die grundlegenden Ideen und Konzepte des Algorithmus in einem viel breiteren Kontext angewendet werden. Der Algorithmus bietet die Basis für eine beschleunigte Suche für verschiedene Problemstellungen, deren Lösungseigenschaften in Form einer Orakelfunktion  $\mathcal{O}$ beschrieben sind. In Kapitel 3 wird der Grover-Algorithmus als Basis für die Suche von reinen Nash-Gleichgewichten in graphischen Spielen vorgestellt. Der grundlegende Ablauf des Algorithmus ist folgendermaßen:

1. Initialisierung des quantenmechanischen Zustands von n Qubits als gleichverteilte Superposition

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \,.$$

- 2. Wiederhole die nächsten Schritte $\frac{\pi}{4}\sqrt{2^n}$ mal:<sup>3</sup>
  - Anwendung einer Orakelfunktion  $\mathcal{O}$ , die den gesuchten Zustand markiert und seine Amplitude mit -1 multipliziert,

$$\mathcal{O} \left| i \right\rangle_{n} = \begin{cases} & \left| i \right\rangle_{n}, \, i \neq w \\ & \left| i \right\rangle_{n}, \, i = w \end{cases}$$

mit  $|w\rangle$  als der gesuchte Zustand.

<sup>&</sup>lt;sup>3</sup>Zu beachten ist, dass die Anzahl der Iterationen zum einen von dem initialen Zustand und zum anderen von der Anzahl der Lösungen abhängt. Im allgemeinen Fall, in dem der initiale Zustand eine gleichverteilte Superposition der Basiszustände ist und die Anzahl der Lösungen eins beträgt, werden ungefähr  $\frac{\pi}{4}\sqrt{2^n}$  Iterationen benötigt [24].

• Anwendung eines sogenannten *Grover-Diffusionsoperator D*, der eine Spiegelung aller Amplituden um den Mittelwert durchführt.

$$D = 2 \left| s \right\rangle \left\langle s \right| - I$$

Die Negation des gesuchten Elements und die Spiegelung der Amplituden um den Mittelwert wird als *Grover-Iteration* bezeichnet, siehe Schritt 2. Sie ist der zentrale Baustein des Algorithmus. Durch die wiederholte Ausführung dieser Schritte wird die Amplitude des gesuchten Zustands erhöht, während die der anderen Zustände verringert wird, damit am Ende der gesuchte Zustand mit einer hohen Wahrscheinlichkeit gemessen wird.

Der ursprüngliche Grover-Algorithmus wurde später generalisiert, sodass man zum einen nicht mehr unbedingt von einer gleichverteilten Superposition starten muss, und zum anderen man auch mehrere gesuchte Zustände haben kann.



Abbildung 2.9: Darstellung der Amplitudenverstärung als Balkendiagramm: Links sind die Amplituden des gleichverteilten Superpositionszustands  $|s\rangle$  als Balken abgebildet. Der blaue Balken repräsentiert die Amplitude des gesuchten Zustands  $|w\rangle$ . Nach Anwendung der Orakelfunktion  $\mathcal{O}$  wird die Amplitude des gesuchten Zustands negiert, was zudem dazu führt, dass die durchschnittliche Amplitude (gekennzeichnet durch die gestrichelte Linie) verringert wird, siehe Abbildung Mitte. Abschließend wird der Diffusionsoperator angewendet, der die Spiegelung der Amplituden um deren Mittelwert verursacht, siehe Abbildung rechts. Da die durchschnittliche Amplitude durch die erste Spiegelung gesenkt wurde, erhöht die zweite Transformation bedingt durch den Diffusionsoperator die negative Amplitude von  $|w\rangle$  auf etwa das Dreifache ihres ursprünglichen Wertes, während sie die anderen Amplituden verringert. Durch mehrmaliges Ausführen dieser beiden Transformationen wird die Wahrscheinlichkeit, den gesuchten Zustand  $|w\rangle$  zu messen, schrittweise erhöht.

Diese Generalisierung wird als Amplitudenverstärkung bezeichnet [26]. In Abbildung 2.9 ist die Amplitudenverstärkung für den Standardfall mit einem gesuchten Zustand und einem initialen gleichverteilten Superpositionszustand visualisiert.

Zu erwähnen ist, dass die Anzahl der Grover-Iterationen von der Anzahl der Lösungen im Suchraum abhängt. Während im oben beschriebenen allgemeinen Fall  $\approx \frac{\pi}{4}\sqrt{2^n}$  Iterationen nötig sind, um den gesuchten Zustand zu finden, reichen im Fall von mehreren Lösungen  $\approx \frac{\pi}{4}\sqrt{2^n/M}$  Rotationen aus, wobei Mdie Anzahl der Lösungen bzw. gesuchten Zustände ist. Da in den meisten realen Anwendungen des Grover-Algorithmus die Anzahl der Lösungen unbekannt ist, wird M entweder zufällig ausgewählt oder mit dem *Quantum Counting* Algorithm geschätzt [1, 24].

# 2.4.3 Implementierung in Hardware

Das Quantenschaltkreismodell wird von einigen Quantum Computing Hardwareherstellern implementiert. Während sich die Hardware oft in der Beschaffenheit der Qubits und der Architektur [5, 72, 97, 100] unterscheidet, sind die theoretischen Konzepte und Vorgehensweisen bei der Algorithmenentwicklung ähnlich. Grundsätzlich müssen verschiedene Faktoren bei der Implementierung von QCM-Algorithmen auf realer Quantum Computing Hardware berücksichtigt werden. Da für die Evaluation der QCM-Algorithmen in dieser Arbeit IBM Hardware und Simulatoren verwendet wurden, werden im Nachfolgenden diese Kernaspekte in Bezug auf die IBM Hardware beschrieben.

### 2.4.3.1 Transpilierung

Bei der Beschreibung von Quantenalgorithmen stehen dem Entwickler viele verschiedene logische Quantengatter zur Verfügung. Jedoch muss beachtet werden, dass die physikalischen Gatter, die von den Hardwareherstellern direkt implementiert werden, meist sehr limitiert sind. Das heißt, vor der Ausführung des Algorithmus werden die logischen Gatter in die nativen physikalischen Gatter transpiliert, was zu einer größeren Schaltkreistiefe führen kann. Daher sollte bereits im besten Falle bei der Algorithmenentwicklung die nativen Gatter der Hardware berücksichtigt werden. Eine Übersicht aller logischen Quantengatter, die bei IBM zur Verfügung stehen, ist in [5] zu finden. Der in dieser Arbeit verwendete 27 Qubit *IBM Quantum Systeme One* Rechner mit Standort in Ehningen besitzt folgende native Gattermenge, {CX, ID, RZ, SX, X}, wobei das CNOT-Gatter (hier als CX-Gatter gekennzeichnet) das einzige physikalische Zwei-Qubit Gatter auf der IBM Hardware ist.

Zudem gibt es ein von IBM zur Verfügung gestelltes universelles logisches Ein-Qubit Gatter, das sogenannte *U-Gatter*. Es bewirkt eine Rotation um die Bloch-Einheitskugel mit drei Eulerschen Winkeln. Die drei Winkel, die zur Durchführung der Rotation verwendet werden, sind  $\theta, \phi$  und  $\lambda$  und bilden die folgende Matrix-Notation [5]:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\lambda+\phi)}\cos(\theta/2) \end{pmatrix}$$
(2.30)

Das U-Gatter mit seinen drei Rotations-Parametern, kann somit jeden beliebigen Ein-Qubit Zustand erzeugen. Somit lassen sich auch die bereits vorgestellten Gatter durch die oben beschriebene Form abbilden. Jedoch werden in entsprechenden Darstellungen von Quantenschaltkreisen meist die ursprüngliche Formulierungen beibehalten, um eine bessere Lesbarkeit zu wahren. Ein Beispiel zur Formulierung des Hadamard-Gatters durch das U-Gatter ist nachfolgend beschrieben:

$$U(\frac{\pi}{2}, 0, \pi) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1\\ 1 & -1 \end{pmatrix} = H$$
 (2.31)

### 2.4.3.2 Architektur

Die Architektur, beziehungsweise Konnektivität der Qubits auf der QPU ist ein weiterer wichtiger Aspekt, der bei der Formulierung von Algorithmen berücksichtigt werden sollte. Grundsätzlich werden Algorithmen in der Theorie auf vollvermaschten Architekturen entwickelt, was wiederum bedeutet, dass Zwei-Qubit-Gatter beliebig zwischen zwei Qubits angewendet werden können. In der Praxis sind aktuelle Prozessoren von Quantenschaltkreismodellen jedoch nicht vollvermascht. In Abbildung 2.10 ist ein entsprechender Topologiegraph G = (V, E) eines 27 Qubit IBM Rechners dargestellt. Die Knoten V repräsentieren die Qubits  $\{q_i | 1 \leq i \leq n\}$  der Hardware und die Kanten E die direkte physikalische Verbindung  $\{\{q,q'\}|q,q' \in E\}$  zwischen zwei Qubits q und q' [105]. Es ist ersichtlich, dass aufgrund der spärlichen Topologie nur bedingt Zwei-Qubit-Operationen angewendet werden können. So müssen die von einer Zwei-Qubit-Operation betroffenen Qubits innerhalb des Graphen durch eine Folge von sogenannten SWAP-Gattern verschoben werden, bis sie im Graphen benachbart sind [105]. Die spärliche Konnektivität der Qubits führt demnach zu zusätzlichen Gattern in Form von SWAPs oder erfordern andernfalls Algorithmen, die eine geeignete Abbildung der logischen Qubits und Operationen auf die physikalischen Qubits der Hardware berücksichtigen.

### 2.4.3.3 Rauschquellen

Schließlich ist es bei der Implementierung eines Quantenalgorithmus wichtig, die Fehler- beziehungsweise Rauschquellen des Quantum Computers zu berücksichtigen. Die beiden größten Rauschquellen sind in der Regel Gatterungenauigkeit und Dekohärenz. Die Gatterungenauigkeit bezieht sich auf die Tatsache, dass die vom Benutzer spezifizierten Gatter nicht genau mit den physikalisch implementierten Gattern übereinstimmen. Da die Gatterungenauigkeit bei Multi-Qubit-Gattern in der Regel höher ist als bei Ein-Qubit-Gattern,



Abbildung 2.10: Topologiegraph eines 27 Qubit Quantum Computers von IBM. Die Knoten repräsentieren die Qubits, und die Kanten die physikalischen Verbindungen zwischen den Qubits. Abbildung aus [91].

möchte man insbesondere die Anzahl der Multi-Qubit-Gatter im Algorithmus minimieren.

Dekohärenz beschreibt zudem ein Phänomen, das auftritt, wenn ein Quantensystem mit der Umwelt interagiert und über die Zeit seine Quanteneffekte verliert und sich somit immer mehr wie ein klassisches System verhält. Dieser Prozess führt zu immer mehr Rauschen und ungewollten Berechnungsfehlern bei der Ausführung des Quantenalgorithmus. Letztendlich begrenzt dieses Phänomen die Schaltkreistiefe, die auf Quantencomputern implementiert, bzw. effektiv ausgeführt werden kann. Zudem sollte bei der Implementierung berücksichtigt werden, dass verschiedene Qubits unterschiedliche Qualitäten aufweisen, bzw. mehr oder weniger von Dekohärenz betroffen sein können. Diese Information sollte grundsätzlich genutzt werden, um den jeweiligen Algorithmus für eine bestimmte Hardware zu konzipieren. Die Fehlerraten der einzelnen Qubits in den IBM Prozessoren sind auf der IBM Webseite aufgeführt [91].

# 2.5 Kapitelzusammenfassung

In diesem Kapitel 2 wurden die wesentlichen Grundlagen des Quantum Computing, die für das Verständnis dieser Arbeit notwendig sind, dargelegt. Dazu wurde zunächst zum einen die wichtigsten Definitionen zur Informationskodierung mittels Qubits und deren Eigenschaften erläutert und zum anderen die Grundlagen zur Zustandstransformation mittels Quantengatter dargelegt. Anschließend wurde sowohl das adiabatische Quantum Computing und dessen Funktionsweise, als auch die verwandte Quantum Annealing Heuristik erläutert. Die Umsetzung der Quantum Annealing Heuristik in Hardware, sowie die Implementierung und Funktionsweise des Quantenschaltkreismodell wurde ebenfalls behandelt. In den nachfolgenden Kapiteln werden die vorgestellten Grundlagen wie folgt angewandt: In Kapitel 3 werden hybride Algorithmen für das Finden von reinen Nash-Gleichgewichten in graphischen Spielen, sowohl für annealingbasierte (Abschnitt 2.3.4) als auch gatterbasierte Quantum Computing Hardware (Abschnitt 2.4.3) vorgestellt. Die Algorithmen basieren zum einen auf Ising-Hamiltonians (Abschnitt 2.3.3), die gleichermaßen mit Quantum Annealing (Abschnitt 2.3) und dem QAOA (Abschnitt 2.4.2.1) ausgeführt werden können und zum anderen auf dem Grover-Algorithmus (Abschnitt 2.4.2.2). Die Evaluation und Anwendung wird auf D-Wave und IBM Quantum Computer durchgeführt.

Anschließend befasst sich Kapitel 4 mit Optimierungsmöglichkeiten verschiedener eingeschränkter Ising-Hamiltonians. Es wird die Kreuzentropie-Methode zur Optimierung der Penalty-Parameter der Hamiltonians (Abschnitt 2.3.3) vorgestellt und die Auswirkung auf die Lösungsqualität auf realer QC-Hardware ermittelt.

Abschließend wird in Kapitel 5 der Einfluss der Penalty-Parameteroptimierung der Hamiltonians auf das jeweilige Eigenspektrum und deren minimale Spektrallücke (Abschnitt 2.2.3) empirisch untersucht.

Weitere Grundlagen, die sich nur auf eines der drei Inhaltskapitel 3-5 beziehen, werden am Anfang des jeweiligen Kapitels behandelt.

# 3 Hybride Algorithmen für die Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen

In diesem Kapitel soll die Anwendbarkeit verschiedener hybrider Algorithmen für kombinatorische eingeschränkte Optimierungs- bzw. Suchprobleme speziell an einem ausgewählten Realwelt-Anwendungsfall evaluiert werden. Die Berechnung von *reinen Nash-Gleichgewichten* (engl. Pure Nash Equilibria (PNE)) in graphischen Spielen ist eine wichtige, NP-vollständige und oft untersuchte Problemstellung der Spieltheorie [51, 76, 95, 160], für die aktuell keine praktischen Quantum Computing Ansätze existieren. Daher wird grundsätzlich zum einen die Fragestellung untersucht, welche Möglichkeiten zur Eingliederung von annealingbasierter, als auch gatterbasierter Quantum Computing Hardware zur Lösung dieser Problemstellung bestehen und zum anderen die sinnvolle Strukturierung der Problemstellung in Teile für das klassische und das Quantum Computing.

Zunächst wird in Abschnitt 3.1 die Motivation und Zielsetzung dieses Kapitels dargelegt bevor dann in Abschnitt 3.2 die Grundlagen der Spieltheorie, die zum Verständnis der Problemstellung erforderlich sind, vorgestellt werden. Insbesondere wird dabei sowohl auf Spiele in Normalform und graphischer Form als auch auf die Definition eines reinen Nash-Gleichgewichts eingegangen, welches das weitaus prominenteste Lösungskonzept für entsprechende Spiele darstellt. Anschließend werden in Abschnitt 3.3 verwandte klassische Algorithmen für diese Problemstellung vorgestellt und auf mögliche Quantenalgorithmen eingegangen, die als Basis zur Lösung verwendet werden können.

Die Abschnitte 3.4 und 3.5 umfassen die jeweiligen hybriden Ansätzen und basieren in gleicher Reihenfolge auf den Publikationen [143] und [142]. In Abschnitt 3.4 wird ein hybrider Ansatz, der die Problemstellung in einen eingeschränkten Hamiltonian kodiert, vorgestellt. Dieser Ansatz kann sowohl auf annealingbasierten, als auch gatterbasierten Hardwarearchitekturen mittels Quantum Annealing beziehungsweise dem QAOA ausgeführt werden. Da zum Zeitpunkt dieser Arbeit kein anderer anwendungsbezogener Quantenalgorithmus für diese Problemstellung bekannt ist, wird der Ansatz zuerst mit vergleichsweise kleinen Spielinstanzen auf unterschiedlicher NISQ-Hardware (D-Wave und IBM) evaluiert, bevor mittels einer quanten-klassischen Heuristik (QBSolv) der Hamiltonian-Ansatz mit größeren und relevanteren Spielinstanzen gegenüber zwei klassischen Methoden verglichen wird.

Der darauf folgende hybride Ansatz in Abschnitt 3.5 macht sich den Grover-Algorithmus und dessen Amplitudenverstärkung zu Nutze. Hier wird die Problemstellung in eine Orakelfunktion kodiert, die es ermöglicht, Zustände, die reinen Nash-Gleichgewichten entsprechen, zu identifizieren und deren Amplitude iterativ zu verstärken, sodass diese mit hoher Wahrscheinlichkeit gemessen werden können. Von besonderem Interesse bei diesem Quantenschaltkreisalgorithmus ist die Abschätzung der Skalierbarkeit für Spiele unterschiedlicher Größe und die damit verbundene Einordnung der Anwendbarkeit in realen Szenarien.

Abschnitt 3.6 fasst das Kapitel und dessen wichtigste Erkenntnisse kurz zusammen. Eine detailliertere Zusammenfassung der vorgestellten hybriden Algorithmen sowie ein Ausblick auf mögliche Ansatzpunkte für zukünftige Arbeiten, ist am Ende eines jeden Unterkapitels zu finden.

# 3.1 Motivation und Zielsetzung

Die Anwendung der konventionellen Spieltheorie spielt eine wichtige Rolle in vielen modernen strategischen Entscheidungsprozessen der Wirtschaft, Diplomatie und nationalen Sicherheit [33, 136, 148]. Die Spieltheorie ist ein mathematisches Modell, in dem entsprechende domänenspezifischen Entscheidungssituationen abgebildet werden und mehrere Spieler miteinander interagieren können, um gemeinsam eine Aufgabe zu erfüllen oder ihre Interessen durchzusetzen [68]. Im klassischen Modell der Spieltheorie [120] wählen alle Spieler gleichzeitig eine Aktion und erhalten eine bestimmte Auszahlung, die von den Aktionen der anderen Spieler abhängt. Das weitaus prominenteste Lösungskonzept für ein solches Entscheidungsproblem ist das Nash-Gleichgewicht [118], bei dem kein Spieler in der Lage ist, seine Auszahlung einseitig zu verbessern, indem er seine gewählte Aktion im Nachhinein ändert.

Es gibt viele verschiedene Darstellungen für solche statischen Spiele. Die populärste ist die strategische Normalformdarstellung, die häufig für Spiele mit zwei Spielern, wie das *Gefangenendilemma* oder der *Kampf der Geschlechter*, verwendet wird. Aufgrund der exponentiell wachsenden Darstellungsgröße in Bezug auf die Anzahl der Spieler [99] wird jedoch zunehmend eine kompaktere Version, die so genannten graphischen Spiele, zur Modellierung von Szenarien mit mehreren Spielern verwendet [103, 107, 116, 127]. Dabei hängt die Aktion und deren Auszahlung eines Spielers nur von einer bestimmten Anzahl von Aktionen anderer Spieler ab. Diese Spieler werden als sogenannte *Spielernachbarschaft* bezeichnet und durch einen Graphen mit Spielern als Knoten und den Abhängigkeiten als Kanten repräsentiert.

Während Spielformen mit *reinen Strategien*, bei denen sich jeder Spieler eindeutig für eine bestimmte Aktion entscheidet, konzeptionell einfacher sind als mit *gemischten Strategien*, scheinen die damit verbundenen Berechnungsprobleme schwieriger zu sein [76]. Dies gilt auch für die kompakte Darstellung von graphischen Spielen, für die die Komplexität für das Finden von reinen Nash-Gleichgewichten als NP-vollständig bewiesen wurde, selbst im eingeschränkten Fall von Nachbarschaften mit einer maximalen Größe von 3 Spielern mit höchstens 3 Aktionen bzw. Strategien pro Spieler [76].

Obwohl viele Algorithmen existieren, die gemischte oder approximierte NE in graphischen Spielen berechnen [17, 99, 125, 160], gibt es nur eine Handvoll klassischer Algorithmen, die sich mit dem NP-vollständigen Problem der Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen beschäftigen. Mit dem Aufkommen der Quantum Computing Technologie besteht nun die Möglichkeit, diese Problemstellung mit neuen Konzepten und Lösungsansätzen anzugehen. Da zum aktuellen Zeitpunkt keine Ansätze bekannt sind, die diese Möglichkeiten zur Findung von reinen Nash-Gleichgewichten in graphischen Spielen einbeziehen, ist es das Ziel dieses Kapitels, neue Algorithmen für diese Problemstellung zu erarbeiten und deren Anwendbarkeit auf aktueller NISQ-Hardware zu evaluieren.

# 3.2 Grundlagen der Spieltheorie

Die Spieltheorie wurde von Neumann und Morgenstern begründet [120] und beschreibt die wissenschaftliche Analyse von Situationen in denen Individuen miteinander interagieren und Entscheidungen treffen. Diese Individuen können Menschen, Unternehmen oder Zusammenschlüsse wie etwa Parteien sein. Voraussetzung der Spieltheorie ist jedoch, dass die Individuen rational sind und entsprechend rationale Entscheidungen treffen. In der Spieltheorie werden dann diese Echtweltsituationen mit Hilfe von Spielen abstrahiert und mathematisch formalisiert. Das erlaubt es, diese Situationen zu evaluieren und Annahmen beziehungsweise Verhaltensmuster zu erforschen [126]. Nachfolgend werden die wichtigsten Spielformen und Konzepte der Spieltheorie für das Verständnis der hybriden Algorithmen dargelegt.

# 3.2.1 Spiele in Normalform

Die Normalform eines Spiels ist eine Darstellungsform der Spieltheorie, die die Entscheidungssituation mehrerer Spieler modelliert und sich dabei auf deren im vornherein festgelegten Strategiemengen und Auszahlungsfunktion beschränkt. Diese Form von Spielen wird meist in Spielsituationen verwendet, wenn die Spieler gleichzeitig eine ihrer Strategien bzw. Aktionen wählen, ohne dabei Kenntnis von der Strategiewahl der Gegenspieler zu haben [126]. Obwohl diese Darstellungsform auf eine Vielzahl von einfachen Situationen angewandt werden kann, hat sie vergleichsweise weniger Relevanz für Echtweltproblemstellungen. Sie bildet allerdings die Grundlage für viele andere komplexere Spielmodellierungen, wie beispielsweise die später diskutierten graphischen Spiele (siehe Abschnitt 3.2.2). Formal sind Spiele in Normalform als Tupel  $\langle P, Act, U \rangle$  definiert. P beschreibt die - in dieser Arbeit endliche und nicht leere - Menge, die alle n Spieler des Spiels beinhalten. Act ist eine Funktion, die für einen Spieler p eine Menge von möglichen Aktionen zurückgibt,  $Act : P \to S_p$ . Diese Menge wird Strategiemenge genannt und als  $S_p$  mit  $|S_p| \geq 2$  bezeichnet. Ein Element einer Strategiemenge eines Spielers p wird Strategie oder Aktion genannt. Das Resultat der Strategiewahlen aller Spieler ist das Strategieprofil  $s \in S$  mit  $S = \prod_{i=1}^{n} S_i$  [16, 85]. U ist eine Auszahlungsfunktion, die einen Wert für eine gespielte Aktion zurückgibt:  $u_p : s \to \mathbb{R}$ . Dieser Wert repräsentiert die Spielerpräferenz, beziehungsweise den Nutzen des Spielers p, wenn das Strategieprofil s gespielt wird [85].

Spiele in Normalform mit zwei Spielern werden oft in Tabellen dargestellt. Die Tabelle 3.1 versinnbildlicht die Funktionsweise. Die Strategiemenge des ersten Spielers ist  $\{I, J\}$ , die Strategiemenge des zweiten Spielers ist  $\{L, K\}$ . Die möglichen Aktionen des ersten Spielers werden in der Reihe notiert, während die Aktionen des zweiten Spielers spaltenweise abgebildet werden. [126]



Tabelle 3.1: Beispielhafte Darstellung eines Spiels in Normalform nach [126].

Zwei Ganzzahlen beschreiben den Nutzen einer Aktion. Sollte beispielsweise der erste Spieler I als Aktion wählen und sein Gegenspieler L, dann wird Spieler 1  $w_1$  und Spieler 2  $w_2$  als Auszahlung erhalten. Anders ausgedrückt: Sei das Strategieprofil  $s = \{I, L\}$ , so gilt  $u_1(s) = w_1$  und  $u_2(s) = w_2$ . Die Spieler wählen gleichzeitig und unabhängig, ohne gegenseitige Absprache ihre Strategie. Sie kennen vor der Wahl die Gewinne aller möglichen Aktionen, auch die der Gegenspieler. Daraufhin bekommen die Spieler den Gewinn nach der jeweiligen Auszahlungsfunktion in Abhängigkeit des Strategieprofils. Eine Lösung des Spiels ist ein rationales Strategieprofil. Das bedeutet, dass alle Spieler rational handeln und eine möglichst gute Strategie wählen. Die Annahme ist, dass jeder Spieler versucht den eigenen Gewinn zu maximieren, beziehungsweise den eigenen Verlust zu minimieren [126].

# 3.2.2 Spiele in graphischer Form

Spiele in Normalform beschreiben die Auszahlungsfunktion mit einer Tabelle, die für jede Strategiekombination aller Spieler einen Eintrag besitzt. Diese Darstellungsform ist jedoch sehr speicherintensiv und die darzustellenden Parameter für n Spieler mit jeweils s Strategien steigen exponentiell mit der Anzahl der Spieler an,  $ns^n$  [99]. Dies ist vor allem bei Spielen unvorteilhaft, in denen alle Spieler miteinander interagieren [76]. Allerdings kommt dieser Fall in den wenigsten Realwelt-Spielen vor, insbesondere wenn die Spieleranzahl hoch ist. Oft interagieren die Spieler nur mit einer begrenzten Anzahl an anderen Spielern, was eine wesentlich effizientere Darstellung ermöglicht. Bei solchen Spielen kann dann die Struktur beziehungsweise die Topologie des Spiels helfen, ein besseres Verständnis für das Spiel zu erlangen.

Spiele in graphischer Form lösen sowohl das Ressourcen-, als auch das Strukturproblem der Normalform. Formal sind graphische Spiele als Tupel  $\langle G, M \rangle$ definiert. G repräsentiert einen ungerichteten Graphen, dessen n Knoten die n Spieler des Spiels darstellen. M ist eine Menge von n Tabellen beziehungsweise Matrizen.  $M_p$  stellt die jeweilige Auszahlungsfunktion des Spielers p dar und wird als lokale Spielmatrix bezeichnet. Die direkten Nachbarn des Spielers p im Graphen G werden als die Nachbarschaft  $N_G(p) \subseteq \{1, ..., n\}$  definiert, d.h. die Knoten q die eine Kante (p, q) im Graphen G haben. Konventionell beinhaltet  $N_G(p)$  immer auch den Spieler p selbst. Folglich wird die Nachbarschaft so interpretiert, dass jeder Spieler nur mit seinen direkten Nachbarn im Graphen G spielt und entsprechend nur diese einen Einfluss auf die jeweilige Auszahlungsfunktion haben.

Die Größe der graphischen Spieldarstellung ist daher, im Gegensatz zur Normalformdarstellung, nur exponentiell im Knotengrad d des Graphen,  $ns^d$ . Wenn  $|N_G(p)| = k$  und  $s \in \prod_{i=1}^k S_i$ , dann definiert  $M_p(s)$  die Auszahlung an den Spieler p, wenn seine k Nachbarn (einschließlich seiner selbst) das Strategieprofil s spielen.

In Bezug auf die graphischen Spiele in dieser Arbeit mit n Spielern und den Strategiemengen  $S_1, ..., S_n$  gelten die folgenden Definitionen. Für jedes Strategieprofil  $s \in S$ , ist die Strategie von Spieler p als  $s_p$  definiert, während  $s_{-p}$  alle (n-1)-Strategietupel der anderen Spieler entspricht. Für jedes  $s'_p \in S_p$  und  $s_{-p} \in S_{-p}$  bestimmt  $(s_{-p}; s'_p)$  das Strategieprofil, in dem Spieler p entsprechend  $s'_p$  und alle anderen Spieler gemäß  $s_{-p}$  spielen.

Abschließend sei erwähnt, dass ein Strategieprofil s als global betitelt wird, wenn alle n Spieler dazu beitragen, d.h. ein globales Strategieprofil s besteht darin, dass jeder Spieler eine seiner Aktionen spielt.

# 3.2.3 Reines Nash-Gleichgewicht

Das Nash-Gleichgewicht ist ein elementares Lösungskonzept, das 1950 von John Nash im Zusammenhang von nicht-kooperativen Spielen entwickelt wurde [118]. Das reine Nash-Gleichgewicht bezieht sich dabei auf Spiele, in denen sich die Spieler eindeutig für eine ihrer Aktionen entscheiden und somit reine Strategien spielen. Ein globales Strategieprofil stellt ein reines Nash-Gleichgewicht dar, falls für jeden Spieler p mit seiner Strategie  $s'_p \in S_p$ ,

$$M_p(s) \ge M_p(s_{-p}; s'_p),$$
 (3.1)

gilt. Das bedeutet, dass ein reines Nash-Gleichgewicht besteht, sobald ein einseitiges Abweichen von einer Aktion für keinen Spieler profitabel ist.

# 3.2.4 Beste Antwortstrategien

Die Idee der *besten Antwortstrategien* bildet die Überlegungen und optimalen Reaktionen eines jeden Spielers p auf die erwarteten Strategien der Gegenspieler ab. Diese Reaktionen werden beste Antwortstrategien genannt, da sie den Nutzen des Spielers für die erwartete Strategie der Gegenspieler  $s_{-p}$  maximiert. Die beste Antwortstrategien eines Spielers p ist folgendermaßen definiert [51]:

$$BR_{M_p}(s_{-p}) \triangleq \{s_p \mid s_p \in S_p \text{ und } \forall s'_p \in S_p : M_p(s_{-p}; s_p) \ge M_p(s_{-p}; s'_p)\} \quad (3.2)$$

Dabei ist  $BR_{M_p}(s_{-p})$  die Menge der Strategien in  $S_p$ , die die Auszahlung von Spieler p maximieren, wenn die anderen Spieler gemäß  $s_{-p}$  spielen. Die Menge kombinierter Strategien, die jeweils eine beste Antwortstrategie auf die gespielten Strategien der anderen Spieler darstellt, wird als  $\mathcal{B} = \{BR_{M_p} \mid (1 \leq p \leq n)\}$  bezeichnet und hat die Kardinalität  $C_{\mathcal{B}} = \sum_{p=1}^{n} |BR_{M_p}|$ . Wenn jeder Spieler eine für sich optimale Aktion auswählt und die optimalen Strategien der anderen Mitspieler erwartetet, spricht man von wechselseitigen besten Antworten. Ist dies der Fall, hat kein Spieler einen Anreiz seine Aktion einseitig zu verändern, da der Nutzen bzw. die Auszahlung für jeden Spieler bereits maximiert ist. Liegen wechselseitig beste Antworten vor, also ein globales Strategieprofil, das ausschließlich aus besten Antworten besteht, liegt auch ein Nash-Gleichgewicht vor [85].

# 3.3 Verwandte Arbeiten

Die Berechnung von Nash-Gleichgewichten ist eine viel untersuchte Problemstellung. Es gibt mehrere klassiche Algorithmen, die Nash-Gleichgewichte in graphischen Spielen berechnen. Die meisten konzentrieren sich jedoch auf das Finden von NE in *gemischten* Strategien oder dem Approximieren entsprechende NE [17, 57, 99, 107, 125, 160]. Es wurden aber auch einige Untersuchungen zur Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen und ähnlichen Varianten davon durchgeführt, die für diese Arbeit von Bedeutung sind.

Ein Algorithmus, der reine Nash-Gleichgewichte in graphischen Spielen findet, wurde von Daskalakis und Papadimitriou vorgestellt [51]. Die Autoren zeigen, dass ein beliebiges graphisches Spiel auf ein Markov-Netzwerk (engl. Markov Random Field (MRF)) abgebildet werden kann, sodass das Finden einer maximalen a posteriori-Konfiguration des MRF die Frage beantwortet, ob in diesem Spiel ein reines Nash-Gleichgewicht existiert [51]. Zudem können durch die Berechnung der Randverteilungen der Cliquen des Markov-Netzwerks alle reinen Nash-Gleichgewichte im Spiel beschrieben werden [51]. Darüber hinaus belegen sie, dass durch die Anwendung des *Junction-Tree*-Algorithmus auf das MRF reine NE für eine große Anzahl an Spielen in polynomieller Laufzeit berechnet werden können. Dies gilt dann, wenn die Spiele eine Baumweite von  $O(\log n)$  mit n als die Knoten- bzw. Spieleranzahl, beschränkter Nachbarschaftsgröße und beschränkter Kardinalität der Strategieprofile aufweisen. Jedoch sind diese Bedingungen bei vielen Realwelt-Spielsituation nicht gegeben, sodass die Berechnung des MRFs weiterhin ein NP-schweres Problem bleibt [51].

Ferner gibt es Algorithmen, die graphische Spiele auf ein Bedingungserfüllungsproblem (engl. Constraint-Satisfaction-Problem (CSP)) abbilden. Ein CSP-Problem ist ein Tupel CSP = (X, D, C), bestehend aus einer Menge von Variablen  $X = \{X_1, ..., X_n\}$ , einen dazugehörigen Definitionsbereich  $D = \{D_1, ..., D_n\}$  der Variablen und einer Menge an Beschränkungen  $C = \{C_1, ..., C_m\}$ . Jede Beschränkung beinhaltet eine Teilmenge der Variablen und bestimmt die zulässigen Werte dieser Variablen. Ziel ist es dann, eine Belegungen von Variablen zu finden, die alle aufgestellten Bedingungen erfüllt. Es gibt mehrere Arbeiten, welche die Reduktion des Nash-Gleichgewichtsproblems auf ein CSP untersucht haben. Dazu zählen Gottlob et al. [76], Soni et al. [160], und Jiang [95]. Gottlob et al. stellten fest, dass die Berechnung von reinen Nash-Gleichgewichten in Form eines CSP in polynomieller Zeit durchzuführen ist, wenn die Breite des entsprechenden Hypergraphen des Spiels begrenzt ist. Dies ist beispielsweise der Fall, wenn der Hypergraph azyklisch ist. Diese restriktive Anforderung hat jedoch einen geringen Praxisbezug [76]. Für die meisten Spiele ist die Berechnung von reinen Nash-Gleichgewichten in einem graphischen Spiel in CSP Form ein NP-vollständiges Problem. Dies gilt auch, wenn die direkte Nachbarschaft auf drei Spieler beschränkt ist und eine festgesetzte Anzahl an Aktionen vorliegt [76].

Jiang und Leyton-Brown verfolgen einen ähnlichen Ansatz. In [93] stellen die Autoren einen Algorithmus für sogenannte *Aktionsgraphen-Spiele* (engl. Action Graph Games (AGG)), eine weitere komprimierte Darstellung von Spielen, vor. Dieser Algorithmus berechnet Gleichgewichte für symmetrische AGGs mit beschränkter Baumweite in polynomieller Laufzeit mit Hilfe der dynamischen Programmierung [93]. Jedes graphische Spiel kann auf ein nicht-symmetrisches AGG abgebildet werden. Ist die Menge der Strategien jedoch beschränkt, bleibt bei der Abbildung auf das AGG auch die Baumweite beschränkt. Allerdings ist die Berechnung von reinen Nash-Gleichgewichten für nicht-symmetrische Aktionsgraphen-Spiele NP-vollständig, selbst wenn die Baumweite eins beträgt [165].

Anhand dieser zahlreichen Veröffentlichungen ist die Bedeutsamkeit der Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen offensichtlich. Mit der stetigen Weiterentwicklung von Quantum Computing Hardware, besteht zudem nun die Möglichkeit, solche Problemstellungen mit neuen Ansätzen zu bewältigen. Das Finden von PNE in graphischen Spielen lässt sich auf ein kombinatorisches Optimierungsproblem beziehungsweise Suchproblem reduzieren. In [182] geben die Autoren einen Überblick über die verschiedenen Ansätze zur Lösung kombinatorischer Optimierungsprobleme mittels Quantum Computing. Darunter fällt in Bezug auf das Quantenschaltkreismodell die Simulation von Hamiltonians [3, 50], der Grover-Algorithmus in Form einer globalen Suche [12], als auch der QAOA zur Approximierung von Optimierungsproblemen in Form von Ising-Hamiltonians [60]. Zum aktuellen Zeitpunkt ist noch kein praktischer QC-Ansatz zur Berechnung von PNE in graphischen Spielen bekannt, weshalb in den nachfolgenden Abschnitten zwei entsprechende Konzepte sowohl für gatterbasierte als auch annealingbasierte Quantum Computing Hardwarearchitekturen vorgestellt und evaluiert werden. Während im ersten Ansatz die Problemstellung als Hamiltonian definiert wird und somit sowohl auf Quantum Annealing Hardware, als auch mit dem QAOA auf Hardware des Quantenschaltkreismodell ausgeführt werden kann, wird im zweiten Ansatz das Problem in Form einer Orakelfunktion kodiert, die im Zusammenspiel mit der Amplitudenverstärkung des Grover-Algorithmus auf gatterbasierter Hardware ausgeführt werden kann.

# 3.4 Eingeschränkter Hamiltonian-Ansatz für das Finden von reinen Nash-Gleichgewichten

Im Nachfolgenden wird die Formulierung des Optimierungs- bzw. Suchproblems, in Bezug auf das Finden von reinen Nash-Gleichgewichten, in graphischen Spielen als eingeschränkter Ising-Hamiltonian erläutert. Anschließend wird der Hamiltonian-Ansatz sowohl auf D-Wave Quantum Annealing Hardware als auch mittels QAOA auf IBM Hardware ausgeführt und evaluiert. Zudem wird dessen Anwendbarkeit in Hinblick auf die benötigten Hardwareressourcen auf aktueller QC-Hardware bestimmt.

# 3.4.1 Konzept

Der hybride Hamiltonian-Ansatz zur Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen wird im Folgenden *Q-Nash* bezeichnet und besteht aus zwei Phasen, die im Folgenden beschrieben werden.

Bestimmung der besten Antwortstrategien In der ersten Phase wird für jeden Spieler die beste Antwortstrategie auf das, was die anderen Spieler tun

könnten, ermittelt. Das heißt, es wird für jedes Strategieprofil s die Strategie (oder Strategien) mit der maximalen Auszahlung  $M_p(s)$  des Spielers p gesucht. Dabei werden die lokalen Spielmatrizen der einzelnen Spieler der Reihe nach durchlaufen und ihre optimalen Strategien ermittelt (siehe Beispiel 1). Dies ist in polynomieller Zeit machbar. Der erste Teil wird daher auf einem klassischen Computer ausgeführt, der als Pre-Processing des Q-Nash Algorithmus angesehen werden kann. Danach erhält man die Menge kombinierter Strategien, die jeweils eine beste Antwortstrategie auf die gespielten Strategien der anderen Spieler darstellen. Wie bereits in Abschnitt 3.2.4 beschrieben, wird diese Menge als  $\mathcal{B} = \{BR_{M_p} \mid (1 \le p \le n)\}$  und deren Kardinalität als  $C_{\mathcal{B}} = \sum_{p=1}^{n} |BR_{M_p}|$  bezeichnet.

## Beispiel 1:

In Tabelle 3.2 wird die lokale Auszahlungsmatrix  $M_A$  des Spielers A visualisiert. Nehmen wir an, Spieler B spielt Aktion 2 und Spieler C wählt Aktion 1. In diesem Fall ist eine beste Antwortstrategie für Spieler A die Aktion 0, da er in dieser Situation den höchsten Auszahlungsbetrag erhält. Dies führt zu einem besten Antwortstrategieprofil für Spieler A, die durch die Menge  $\{A0, B2, C1\}$ dargestellt wird.

A	B0 C0	B0 C1	B1 C0	B1 C1	B2 C0	B2 C1
0	4	1	<b>2</b>	<b>2</b>	1	4
1	1	3	<b>2</b>	1	<b>2</b>	2

Tabelle 3.2: Lokale Auszahlungsmatrix  $M_A$  des Spielers A, dessen Auszahlung von Spieler B und C abhängig ist. Die fett markierten Auszahlungen entsprechen den beste Antwortstrategien des Spielers A.

**Eingeschränkter Hamiltonian für das Finden von PNE** In der zweiten Phase des Algorithmus wird ein reines NE identifiziert, wenn alle Spieler gleichzeitig eine ihrer besten Antwortstrategien spielen. Mit der klassisch berechneten Menge  $\mathcal{B}$  aus der ersten Phase stellt sich die folgende Frage:

"Gibt es eine Vereinigung der kombinierten besten Antwortstrategien von  $\mathcal{B}$ , die zu einem globalen Strategieprofil führt, unter der Prämisse, dass jeder Spieler eine seiner besten Antwortstrategien spielt?" — Wie in Definition (3.2) angegeben, würde dies zu einem reinen NE führen.

In der Vorveröffentlichung [143] wurde ein Hamiltonian basierend auf dem Set*Cover Problem* entwickelt, der in der Lage ist, reine NE aus den besten Antwortstrategieprofilen der Spieler zu bestimmen. In der Zwischenzeit wurde im Rahmen dieser Arbeit ein neuer ressourceneffizienterer Hamiltonian identifiziert, dessen benötigte Variablen linear mit der Anzahl der besten Antwortstrategieprofilen skaliert. Im Vergleich dazu skalieren die Variablen des auf dem Set Cover Problem basierten Hamiltonians polynomiell, vgl. [108].

Der neu entwickelte Q-Nash Hamiltonian beruht ebenfalls auf einem Mengenproblem. Gesucht werden Teilmengen  $V_i \subseteq U$  mit  $1 \leq i \leq N$ , deren Vereinigung ein globales Strategieprofil ergibt, in der jeder Spieler genau eine seiner besten Antwortstrategien spielt. U entspricht dabei der Menge aller besten Antwortstrategien  $\mathcal{B}$  mit der Kardinalität  $C_{\mathcal{B}}$  und einer Teilmenge  $V_i$ mit  $1 \leq i \leq C_{\mathcal{B}}$  einer einzelnen beste Antwortstrategiekombination aus  $\mathcal{B}$ . Wie aus dem Beispiel 1 aus der ersten Phase des Algorithmus ersichtlich, bestehen die Elemente einer Teilmenge bzw. einer besten Antwortstrategiekombination  $V_i$  immer aus einer Zusammensetzung aus Spieler und seiner Strategie.  $\mathcal{V}_i$  sei nun die gleiche Teilmenge  $V_i$  reduziert auf die Spieler. Beispielsweise reduziert sich dann  $V_1 = \{A0, B2, C1\}$  auf  $\mathcal{V}_1 = \{A, B, C\}$ . Folgender eingeschränkter Hamiltonian beschreibt nun die Berechnung reiner NE,

$$H = A \sum_{\substack{i,j:|V_i \cup V_j| \\ \neq |\mathcal{V}_i| + |\mathcal{V}_j|}} x_i x_j + A \left( n - \sum_{i=1}^{C_{\mathcal{B}}} x_i \right)^2,$$
(3.3)

mit  $x_i \in \{0, 1\}$  als Binärvariable und n gleich der Anzahl der Spieler. Der erste Term gewährleistet, dass die Kardinalität der Vereinigung von zwei besten Antwortstrategiekombinationen gleich der Kardinalität der gleichen, auf die Spieler reduzierten, beste Antwortstrategiekombinationen ist. Im Falle einer Verletzung der Bedingung, d.h., wenn der Term ungleich 0 ist, wird ein Penalty-Parameter A auf die Lösungsenergie des Hamiltonian addiert. Durch das Hinzufügen eines Penalty-Parameters wird die entsprechende Lösung somit als ungültig, beziehungsweise ungünstig aufgrund der erhöhten Energie, deklariert.

Des Weiteren muss dem Hamiltonian ein zusätzlicher Term als Nebenbedingung hinzugefügt werden. Dieser zweite Term stellt sicher, dass genau n Teilmengen von  $\mathcal{B}$  in der Lösung enthalten sein müssen, um ein globales Strategieprofil zu bilden. Diese Einschränkung stellt implizit sicher, dass jeder Spieler genau eine seiner besten Antwortstrategien spielt. Eine Veranschaulichung ist in Beispiel 2 gegeben.

### Beispiel 2:

Um die Funktion des letzten Penalty-Terms von Gleichung 3.3 zu demonstrieren, wird in Abbildung 3.1 ein Auszug von beste Antwortstrategiekombinationen<sup>1</sup> aus  $\mathcal{B}$  für ein beliebiges 4-Spieler-Spiel visualisiert. Die grüne Vereinigung

<sup>&</sup>lt;sup>1</sup>Die beste Antwortstrategiekombinationen ist in Form von Mengen dargestellt, wobei die fettgedruckte Spieler-Aktionskombination die Zugehörigkeit der Strategie zu dem entsprechenden Spieler definiert.

von vier beste Antwortstrategiemengen führt zu einem PNE, bei der jeder Spieler eine seiner besten Antwortstrategien spielt. Obwohl die rote Vereinigung von drei Mengen ebenfalls zu einer globalen kombinierten Strategiemenge führt, in der jeder Spieler eine seiner Aktionen spielt, handelt es sich nicht um ein PNE, da Spieler B keine seiner besten Antwortstrategien spielt.



Abbildung 3.1: Auf der linken Seite ist ein Ausschnitt der Obermenge  $\mathcal{B}$  gegeben. Auf der rechten Seite sind die Mengen vereinigt, wobei jede Vereinigung (1) und (2) eine globale Strategiekombination darstellt, aber nur (1) ein reines Nash-Gleichgewicht beinhaltet.

# 3.4.2 Aufbau der Experimente

Um den hybriden Hamiltonian-Ansatz Q-Nash zu evaluieren und dessen Anwendbarkeit auf aktueller NISQ-Hardware aufzuzeigen, sind die Experimente zweiteilig aufgebaut. Im ersten Teil der Experimente wird Q-Nash für vergleichsweise kleine Probleme sowohl auf D-Wave Hardware mittels Quantum Annealing als auch auf IBM Hardware mittels dem QAOA ausgeführt. Im zweiten Teil wird der Ansatz auf größeren Probleminstanzen mittels einer quanten-klassischen Heuristik getestet, die es ermöglicht, ressourcenüberschreitende Ising-Hamiltonians auf aktueller Quantum Annealing Hardware auszuführen. In den nächsten beiden Abschnitten werden die Referenzmethoden und die Topologien der graphischen Spielinstanzen der Experimente vorgestellt.

# 3.4.2.1 Referenzmethoden

**Quantum Annealing** Für die Durchführung der Quantum Annealing Experimente wurde D-Wave Systems 2000Q Hardware verwendet. Die Funktionsweise und deren in Hardware implementierte Heuristik wurde bereits in den Abschnitten 2.3 und 2.3.4 erläutert. Aufgrund der vollvermaschten logischen Q-Nash Ising-Problemformulierung muss für das Hardware-Embedding ebenfalls eine vollvermaschte Struktur im Chimera-Graphen der D-Wave 2000Q Hardware gefunden werden. Hierfür wurde D-Wave Systems Software *Minorminer* verwendet, die heuristisch versucht, eine Abbildung eines Minorgraphen (logische Ising-Probleminstanz) in einen Zielgraphen (Chimera-Graphen) zu finden [30]. Die Qubit-Ketten, die durch die spärliche Konnektivität der Hardware-Qubits resultieren, wurden gemäß den D-Wave Systems Richtlinien mit dem *Chain Strength* Parameter um 1 höher als der maximale Betrag des Ising-Modell-Koeffizienten gesetzt, um das Brechen von Qubit-Ketten zu vermeiden [44]. Nach jeder Ausführung wurde 1024 mal gemessen. Die Annealing-Zeit wurde auf  $3\mu$ s und  $20\mu$ s (D-Waves voreingestellte Annealing-Zeit) gesetzt.  $3\mu$ s entspricht ungefähr der Zeit, die die IBM Hardware benötigt, um ein Sample für den QAOA für p = 1 zurückzuliefern [175].

**QAOA** Um Q-Nash mittels QAOA auch auf aktueller gatterbasierter Hardware zu testen, wurde, neben dem IBM QASM Simulator, der IBMQ-27 Rechner in Ehningen eingesetzt. Der allgemeine Ablauf des QAOA und die IBM Hardware wurden bereits in den Abschnitten 2.4.2.1 und 2.4.3 vorgestellt. Für die klassische Optimierung der Gatterparameter wurde die *Constrained Optimization By Linear Approximation* (COBYLA) Methode verwendet [133]. Nach jeder Iteration wurde 1024 mal gemessen. Entsprechend zu den Annealing-Zeiten der D-Wave Experimente wurde für die Länge der Trotterisierung neben p = 1 noch p = 6 gewählt, um etwa den 20 $\mu$ s der voreingestellten D-Wave Annealing-Zeit gerecht zu werden.

**QBSolv** Da die aktuelle Quantum Computing Hardware in Bezug auf die Anzahl der Qubits und deren Konnektivität begrenzt ist, muss man für die größeren Probleminstanzen auf die hybride Heuristik QBSolv zurückgreifen. QBSolv ist eine von D-Wave Systems entwickelte Software<sup>2</sup>, die Ising- und QUBO-Instanzen automatisch in Teilprobleme aufteilt, die dann auf der D-Wave Hardware ausgeführt werden können. Iterativ wird bei QBSolv eine klassische Tabu-Suche angewendet, die alle Teilergebnisse des D-Wave Quantum Annealers nachbearbeitet und zu einem Gesamtergebnis für das ursprüngliche Problem zusammensetzt. Die Anzahl der Iterationen für die Tabu-Suche kann über den Parameter Num Repeats voreingestellt werden, um das Gesamtergebnis schrittweise zu verbessern. Die Größe des Teilproblems (Anzahl der Variablen) kann über einen weiteren Hyperparameter angegeben werden, der zur Aufteilung der ursprünglichen Probleminstanz verwendet werden kann. Die Software kümmert sich dann ebenfalls um das Hardware-Embedding der Ising-Instanzen. Für weitere Informationen zu der inhärenten Tabu-Suche und dem Verfahren im Detail, siehe [20].

<sup>&</sup>lt;sup>2</sup>https://github.com/dwavesystems/qbsolv

**Vollständige Suche** Um die Effektivität von Q-Nash zu bewerten, wurde zum Vergleich eine *vollständige Suche* (engl. Brute Force (BF)) implementiert. Der Algorithmus ermittelt auf die gleiche Weise wie Q-Nash die besten Antwortstrategien aller Spieler und probiert anschließend alle möglichen Kombinationen dieser Strategien aus, um ein gültiges globales Strategieprofil zu bilden, das dann einem reinen Nash-Gleichgewicht entspricht. Die Anzahl der Kombinationen ist exponentiell in Bezug auf die Anzahl der Spieler des Spiels,  $\prod_{p=1}^{n} BR_{M_p}$ .

**Iterierende beste Antwortstrategie** Zusätzlich wurde der Algorithmus für die *iterierende beste Antwortstrategie* (engl. Iterated Best Response (IBR)) implementiert [147]. Bei diesem Verfahren ändert in jeder Iteration ein Spieler seine Aktion auf die Aktion, die die beste Antwort auf die Aktion der anderen Spieler ist. Dies wird so lange wiederholt, bis ein reines Nash-Gleichgewicht erzielt wird. Ist dies der Fall beginnt der Algorithmus erneut mit einem zufällig generierten globalen Strategieprofil, um weitere PNE zu finden. Der Algorithmus terminiert, sobald er eine gewisse Zeitspanne, die als Parameter übergeben werden kann, überschreitet.

## 3.4.2.2 Graphtopologien der Spiele

Für die Evaluierung des hybriden Q-Nash Algorithmus wurde ein Spielgenerator implementiert. Er erzeugt graphische Spielinstanzen von drei verschiedenen populären Graphtopologien, die in der Literatur häufig verwendet werden [93, 103, 127, 169. Diese Topologien sind in Abbildung 3.2 dargestellt. Als Eingabe für den Spielgenerator kann man die Anzahl der Spieler, die Graphtopologie und damit die Abhängigkeiten zwischen den Spielern sowie die Anzahl der Aktionen für jeden Spieler individuell einstellen. Die entsprechenden Auszahlungen werden zufällig aus dem Intervall [0, 15] gewählt. Für die Experimente wurden Spiele mit drei Aktionen pro Spieler betrachtet. Das klassische Nash-Theorem besagt, dass es für jedes Spiel ein Nash-Gleichgewicht im Raum der gemischten Strategien gibt [118]. Da in dieser Arbeit jedoch nur reine Strategien betrachtet werden, können auch (graphische) Spiele ohne PNE generiert werden [126]. In diesem Fall wurde die Spielinstanz verworfen und die zufällige Generierung eines Spiels unter Berücksichtigung der Parameter wiederholt. Zusätzlich ist zu erwähnen, dass Q-Nash, aufgrund der klassischen ersten Phase unabhängig von der graphischen Spielstruktur ist, da die Topologie der graphischen Spiele in ein Mengenüberdeckungsproblem (vgl. Abschnitt 3.4.1) aufgebrochen wird, das dann wiederum durch die paarweisen Abhängigkeiten innerhalb der Hamiltonian-Formulierung vereinfacht dargestellt werden kann. D.h., Q-Nash kann beispielsweise auch auf Spielen deren Abhängigkeitsgraph bzw. Topologie nicht zusammenhängend ist, ausgeführt werden. Das macht es möglich zudem kleinere topologieunabhängige graphische Spielinstanzen zu generieren, die zwar nicht unbedingt einer realen Spielsituation gleichen, aber auf aktueller NISQ-Hardware mittels Q-Nash ausführbar sind.



Abbildung 3.2: Verschiedene graphische Spieltopologien, die die Abhängigkeiten zwischen den Spielern eines Spiels anzeigen. (a) Baumtopologie mit n = 10, (b) Kreistopologie mit n = 6 und (c) Straßentopologie mit n = 6.

# 3.4.3 Ergebnisse und Diskussion

Die Evaluierung des hybriden Hamiltonian-Ansatzes ist in zwei Bereiche geteilt. Zuerst wird Q-Nash für die vergleichsweise kleinen Probleminstanzen sowohl auf gatterbasierter, als auch annealingbasierter Hardware verglichen und die Anwendbarkeit beurteilt, bevor Q-Nash mittels der Software QBSolv für größere und relevantere Spielinstanzen gegenüber der klassischen IBR auf dem D-Wave 2000Q System bewertet wird.

## 3.4.3.1 Q-Nash Algorithmus - QA und QAOA

In Tabelle 3.3 sind die Ergebnisse von Q-Nash auf unterschiedlich großen Spielinstanzen aufgelistet. Die Spielinstanzen variieren von 9 bis 24 beste Antwortstrategien und wurden so generiert, dass sie jeweils genau ein reines Nash-Gleichgewicht beziehungsweise globales Optimum besitzen. Die Hamiltonians wurde mittels QAOA für p = 1 und p = 6 auf dem IBMQ QASM Simulator<sup>3</sup> ausgeführt und mit Quantum Annealing auf der D-Wave Systems 2000Q Maschine. Dabei wurde die Annealing-Zeit auf  $3\mu s$  (das entspricht ungefähr der Zeit, die IBMQ benötigt, um ein Sample für den p = 1QAOA-Quantenschaltkreis zurückzuliefern [175]) und  $20\mu s$  (entspricht ungefähr QAOA mit p = 6 und der voreingestellten D-Wave Systems Annealing-Zeit) gesetzt. Für QAOA wurde zur klassischen Optimierung der Gatterparameter COBYLA eingesetzt, der in [133] im Detail beschrieben wird. Zur Bewertung der Lösungsqualität wurden zwei Metriken gewählt. Die erste involviert die Wahrscheinlichkeit den Grundzustand (GZ) nach der Ausführung

<sup>&</sup>lt;sup>3</sup>Da im nachfolgenden Abschnitt 3.4.3.1 gezeigt wird, dass die Wahrscheinlichkeitsverteilung der Messungen des IBMQ-27 Rechners und des IBMQ QASM Simulators relativ ähnlich sind und der IBMQ-27 Rechner nur bis Spiele von 21 logischen Qubits ausführbar war, wurde für die Ergebnisse in Tabelle 3.3 auf den Simulator zurückgegriffen.

des jeweiligen Algorithmus zu messen. Diese wird folgendermaßen berechnet,

$$GZ-Wahrscheinlichkeit = \frac{\#GZ-Messungen}{\#Messungen},$$
(3.4)

mit #Messungen gleich 1024. Als weitere Metrik wurde das Näherungsverhältnis r verwendet, mit

$$r = \frac{E(\gamma^*, \beta^*) - E_{max}}{E_{min} - E_{max}}.$$
(3.5)

Das Verhältnis r gibt an, wie nah der Erwartungswert  $E(\gamma *, \beta *)$  am Optimum ist und sollte daher möglichst den Maximalwert 1 erreichen. Bei den Spielinstanzen dieser Arbeit können die Eigenwerte des Hamiltonian negative und positive Werte annehmen. Der kleinste und größte Eigenwert wird als  $E_{min}$ bzw.  $E_{max}$  bezeichnet. Folglich kann das Verhältnis  $\frac{E(\gamma *, \beta *)}{E_{min}}$  negative und positive Werte annehmen. Durch Subtraktion des größten Eigenwerts  $E_{max}$  wird das Spektrum so verschoben, dass es nicht positiv ist, und das Verhältnis Werte zwischen  $0 \le r \le 1$  annimmt [175]. Zu erwähnen ist, dass die Ergebnisse aus Tabelle 3.3 auf jeweils fünf gemittelten Ausführungen basieren.

Aus den Ergebnissen geht hervor, dass bei Verwendung des D-Wave 2000Q Systems die Wahrscheinlichkeit den Grundzustand zu messen fast immer größer ist, als die, die sich aus der Ausführung des QAOA auf dem IBMQ QASM Simulator ergibt. Während bei dem Simulator die GZ-Wahrscheinlichkeit bereits ab Spielen der Größe 15 Variablen unter 1.0% liegt, ist dies bei der D-Wave Hardware erst ab Spielen mit 21 Variablen der Fall. Interessanterweise fällt die GZ-Wahrscheinlichkeit des D-Wave 2000Q ab Spielen mit 21 Variablen so sehr ab, dass sie sogar dem QASM Simulator unterliegt und der D-Wave 2000Q Quantum Annealer in den meisten Ausführungen das globale Optimum nicht mehr findet. Fairerweise muss man jedoch bei dem Vergleich mit QAOA berücksichtigen, dass es sich bei dem IBMQ QASM Simulator um simulierte Ergebnisse handelt und der Vergleich daher mit Vorsicht zu betrachten ist.

In Bezug auf das Näherungsverhältnis r ist zu sehen, dass auch hier die Verteilung der Lösungen des D-Wave 2000Q Systems im Durchschnitt näher am globalen Optimum liegt, als die des QASM Simulators. D-Wave 2000Q Quantum Annealer schafft in jeder der Spielinstanzen eine 0.97 bis 0.99 Annäherung an das Optimum. Der QASM Simulator liegt demgegenüber bei einem Näherungsverhältnis von 0.79 bis 0.96. Erstaunlicherweise bleibt das Näherungsverhältnis für sowohl D-Wave Quantum Annealer als auch IBMQ-QASM Simulator relativ konstant über die unterschiedlich großen Spielinstanzen. Diese Beobachtungen werden auch im nachfolgenden Abschnitt 3.4.3.1, bei der Visualisierung der Wahrscheinlichkeitsverteilung der Lösungen, bestätigt.

Abschließend sei zu den Ergebnissen aus Tabelle 3.3 gesagt, dass sowohl für QA als auch QAOA keine Hyperparameteroptimierungen durchgeführt wurden und die voreingestellten hardwareherstellerspezifischen Parameter für die jeweilige Methode verwendet wurden. Ziel dieser Untersuchung war es eher einen grundlegenden Vergleich in Bezug auf die Lösungsqualität des Q-Nash

Variablen (Label)	GZ-Wahrscheinlichkeit (%)				Verhältnis r			
	D-Wave		QAOA		D-Wave		QAOA	
	$3\mu s$	$20 \mu s$	p = 1	p = 6	$3\mu s$	$20\mu s$	p = 1	p = 6
9 (A)	22.36	19.69	0.55	2.18	0.97	0.98	0.89	0.86
9 (B)	21.71	14.52	7.16	1.92	0.97	0.97	0.86	0.9
9 (C)	30.53	19.86	0.29	10.81	0.98	0.98	0.88	0.88
12 (A)	7.42	6.22	0.42	3.03	0.98	0.98	0.82	0.88
12 (B)	5.47	13.54	5.83	2.77	0.98	0.98	0.96	0.87
12 (C)	13.48	11.26	0.36	1.82	0.98	0.98	0.9	0.9
15 (A)	7.98	4.95	0.0	0.36	0.98	0.98	0.81	0.84
15 (B)	3.32	5.05	0.07	0.52	0.98	0.98	0.89	0.84
15 (C)	2.12	6.64	0.03	0.59	0.97	0.98	0.81	0.86
18 (A)	0.29	7.94	0.03	0.1	0.98	0.99	0.83	0.79
18 (B)	1.2	1.14	0.0	0.36	0.98	0.98	0.83	0.87
18 (C)	2.9	3.74	0.33	0.13	0.98	0.99	0.89	0.81
21 (A)	0.03	0.68	0.03	0.65	0.98	0.99	0.95	0.79
21 (B)	0.03	0.1	0.2	0.36	0.98	0.99	0.89	0.8
21 (C)	0.07	0.2	0.0	0.26	0.98	0.99	0.81	0.8
24 (A)	0.0	0.0	0.0	0.2	0.98	0.97	0.79	0.79
24 (B)	0.0	0.0	0.0	0.42	0.97	0.98	0.81	0.84
24 (C)	0.0	0.03	0.0	0.1	0.97	0.98	0.87	0.83

Hamiltonians auf unterschiedlicher Quantum Computing Hardware zu liefern.

Tabelle 3.3: GZ-Wahrscheinlichkeiten und Näherungsverhältnisse von Q-Nash mit QAOA (QASM Simulator) und Quantum Annealing (D-Wave 2000Q Hardware) für unterschiedlich große Spielinstanzen. Für QAOA wurde die Anzahl an alternierende Operatoren auf p = 1 und p = 6 gesetzt. Entsprechend wurde die Annealing-Zeit von  $3\mu s$  und  $20\mu s$  gewählt.

Vergleich der Q-Nash Wahrscheinlichkeitsverteilungen Nachdem in Tabelle 3.3 die GZ-Wahrscheinlichkeit und das Näherungsverhältnis der beiden Ansätze (QA und QAOA) bewertet wurden, sind nun in der Abbildung 3.3 die entsprechenden Wahrscheinlichkeitsverteilung der Messungen des jeweiligen Hamiltonians beispielhaft für zwei Spielinstanzen dagestellt. Hierfür wurden zwei unterschiedlich große Spiele aus Tabelle 3.3 mit 9 und 21 Variablen ausgewählt. In der Abbildung sind die gemittelten Histogramme von fünf Ausführungen mit jeweils 1024 Messungen für Q-Nash mit dem D-Wave 2000Q (QA), IBMQ QASM Simulator (QAOA) und IBMQ-27 Ehningen (QAOA) dargestellt. Für Quantum Annealing wurde die Annealing-Zeit auf  $20\mu s$  und für QAOA die Anzahl der alternierenden Operatoren auf p = 6 gesetzt. Auf der Abszisse sind die Energien des jeweiligen Hamiltonian-Energiespektrums aufgetragen. Die niedrigste Energie (-93 und -356 für Spiel 9 (A) bzw. Spiel 21(A)) stellt den Eigenwert des Grundzustands und somit die Energie des reinen Nash-Gleichgewicht dar. Die zu den Histogrammen ermittelten Wahrscheinlichkeitsverteilungen wurden mittels der Kerndichteschätzung (engl. Kernel Density Estimation (KDE)) angenähert. Dies ist ein gängiges Verfahren, wenn die zugrundeliegende Verteilung der Daten- bzw. Messpunkte unbekannt ist. KDE ist formal als folgende Funktion definiert [172],

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right),$$
(3.6)

mit K(x) als Kernel-Funktion und h > 0 als Parameter für die Bandbreite. Die Kernel-Funktion wird meist durch eine glatte, symmetrische Funktion, wie beispielsweise die Gauß-Funktion, beschrieben und die Bandbreite steuert dabei den Grad der Funktionsglättung. Die KDE glättet jeden Messpunkt  $X_i$ in mehrere kleine Teildichteverteilungen und summiert dann alle diese auf, um die endgültige Dichteschätzung für die Messungen zu erhalten [172].

Aus der Abbildung 3.3a geht hervor, dass die Q-Nash Verteilungen des Simulators als auch des IBMQ-27 Ehningen Quantum Computers vergleichsweise sehr ähnlich sind und sich über das ganze Energiespektrum ausbreiten. Auffallend ist zudem, dass die Wahrscheinlichkeitsverteilung des IBMQ-27 Rechners in Abbildung 3.3b besser abschneidet, als die des Simulators. Während sich erstere in der energetisch niedrigeren Hälfte des Energiespektrums befindet, breitet sich letztere wiederum über das ganze Spektrum aus. Jedoch wurde auf dem IBMQ-QASM Simulator der Grundzustand mit 0.65% Wahrscheinlichkeit gemessen, während auf dem IBMQ-27 Rechner keine der Messungen zum Grundzustand führten. D-Wave 2000Q schneidet in beiden Fällen am besten ab. In Abbildung 3.3a erstreckt sich die Verteilung im energetisch niedrigsten Zehntel (ca. -356 bis 214) des jeweiligen Energiespektrums. In beiden Fällen konnte D-Wave 2000Q den Grundzustand mit einer Wahrscheinlichkeit von 19.69% beziehungsweise 0.68% finden. Wie bereits im vorherigen Abschnitt beschrieben, könnte eine jeweilige Hyperparameteroptimierung der Methoden (QAOA/QA) zu besseren Ergebnissen führen. In Bezug auf die voreingestellten Parameter ist jedoch zu schließen, dass der Q-Nash Hamiltonian zum aktuellen Zeitpunkt und Reifegrad der Hardware auf dem D-Wave 2000Q Quantum Annealer vorteilhafter zu lösen ist.



(b) 21 (A) Spiel

Abbildung 3.3: Gemittelte Histogramme und angenäherte Wahrscheinlichkeitsverteilungen von fünf Ausführungen mit jeweils 1024 Messungen von Q-Nash auf dem D-Wave 2000Q (QA), QASM Simulator (QAOA) und IBMQ-27 Ehningen (QAOA) für das 9 (A) und 21 (A) Spiel. Für Quantum Annealing wurde die Annealing-Zeit auf 20 $\mu s$  und für QAOA die Anzahl der alternierenden Operatoren auf p = 6 gesetzt.
Skalierung des Q-Nash Algorithmus Im nächsten Abschnitt wird die Skalierung von Q-Nash sowohl auf IBMQ-27 Ehningen (QAOA) als auch auf der D-Wave 2000Q (QA) Hardware bewertet. Für die Transpilierung der QAOA-Schaltkreise wurde die physikalisch implementierbare Gattermenge  $\{CX, ID, RZ, SX, X\}$  des IBMQ-27 Ehningen Rechners verwendet. In Abbildung 3.4 ist der transpilierte Schaltkreis für p = 1 und zwei Qubits dargestellt. Vergleicht man die in Abbildung 2.8 des Grundlagenkapitels exemplarisch dargestellte alternierende Trotter-Suzuki Sequenz des generalisierten QAOA, wurde nun jedes dort abgebildete Hadamard-Gatter durch die ersten drei physikalisch implementierbaren Gatter (gelb markiert) des Schaltkreises aus Abbildung 3.4 ersetzt. Diese Transpilierung ist dem Hadamard-Gatter äquivalent mit dem Unterschied einer anderen globalen Phase  $(\frac{7\pi}{4})$ , die aber keinen Einfluss auf die weitere Berechnung des QAOA hat <sup>4</sup>. Um nun im darauf folgenden Schritt die Ising-Modell Koeffizienten  $k_i$  des Q-Nash Problem-Hamiltonians  $H_P$  in den entsprechenden QAOA Problem-Operator  $U_0 = e^{-i\gamma_0 H_P}$  zu kodieren, wird für jeden Diagonaleintrag des Ising-Modells ein RZ-Gatter mit dem entsprechenden Koeffizienten  $k_i$  mal dem  $\gamma$ -Parameter als Rotationsparameter auf das Qubit angewendet (vgl. viertes, in rot markiertes, Gatter aus Abbildung 2.8). Die Off-Diagonaleinträge des Ising-Modells, die den Interaktionen zwischen zwei Qubits entsprechen, werden im QAOA-Schaltkreis wiederum mit jeweils einem parametrisierten RZ-Gatter und zwei umschließenden CX-Gatter auf die jeweiligen zwei Qubits angewendet, wie in Abbildung 3.4 durch die in blau markierten Gatter 5 bis 7 dargestellt. Die Operatoren des Transversalfeld-Hamiltonians  $V_0 = e^{-i\beta_0 H_D}$ , die auf jedes Qubit im Schaltkreis wirken, werden mittels der Transpilierung durch die grün gekennzeichneten Gatter 8 bis 12 aus Abbildung 2.8 realisiert.



Abbildung 3.4: Q-Nash QAOA-Schaltkreis für p = 1 und zwei Qubits unter Berücksichtigung der physikalisch implementierbaren Gattern des IBMQ-27 Ehningen Rechners.

In Tabelle 3.4 sind nun die QAOA-Quantenschaltkreiseigenschaften für p = 1und p = 6 der bereits bekannten Spiele aus Tabelle 3.3 aufgelistet. Zu er-

<sup>&</sup>lt;sup>4</sup>Wie bereits in Abschnitt 2.4.2 erwähnt, entspricht das physikalisch implementierbare  $\sqrt{X}$ -Gatter dem  $R_X(\frac{\pi}{2})$ -Gatter.

kennen ist, dass die Zwei-Qubit Gatter (CX-Gatter) linear mit der Anzahl der alternierenden Operatoren *p* skaliert und sich aus zwei mal der Anzahl der Off-Diagonaleinträge des Ising-Modells ergeben, da, wie bereits beschrieben, das jeweilige parametrisierte RZ-Gatter von zwei CX-Gattern umschlossen wird. Die Anzahl der RZ-Gatter setzt sich zusammen aus den gesamten Einträgen des Ising-Modells, jeweils zwei RZ-Gatter pro Transpilierung des Hardamard-Gatters und jeweils zwei RZ-Gatter pro Transpilierung der Operatoren des Transversalfeld-Hamiltonians. Die SX-Gatter ergeben sich aus der Transpilierung der Hardamard-Gatter und jeweils zwei SX-Gatter pro Transpilierung der Transversalfeld-Operatoren.

Die Tiefe der Schaltkeise skaliert linear mit der Anzahl der logischen Qubits. In Bezug auf die Ergebnisse ist zudem anzumerken, dass die evaluierten Schaltkreise noch nicht die Konnektivität des IBMQ-27 Ehningen Rechners berücksichtigen. Das bedeutet grundsätzlich, dass bei der Transpilierung dieser Schaltkreise auf die NISQ-Hardware, die nicht über vollständig verbundene Qubits verfügt, zusätzliche SWAP-Gatter verwendet werden müssen. Diese werden auf der IBM Hardware wiederum durch drei CX-Gatter abgebildet, was zu einer größeren Schaltkreistiefe beiträgt. Die zunehmende Tiefe des QAOA-Schaltkreises hat bei den Ausführungen des Q-Nash Hamiltonian auf dem IBMQ-27 Ehningen Rechners zum Abbruch für die Spielinstanzen ab 24 logischen Variablen und einer Trotterisierung von p = 6 in der IBMQ-Cloud geführt.

In Tabelle 3.5 ist die Skalierung des Hardwareembeddings des Q-Nash Hamiltonians für die unterschiedlich großen Spielinstanzen auf dem Chimeragraphen des D-Wave 2000Q Quantum Annealers dargelegt. Wie bereits in Abschnitt 2.3.4 des Grundlagenkapitels beschrieben, liefert D-Wave eine Heuristik, um ein möglichst effizientes Embedding für die vollvermaschten logischen Ising-Hamiltonians und dem Hardware Chimera-Graphen zu finden. Aufgrund dieser Vollvermaschung und dem nicht vollständigen Chimera-Graphen müssen sogenannte physikalische Qubit-Ketten gebildet werden, die dann jeweils eine logische Variable repräsentieren. Zu erkennen ist, dass die physikalischen Qubits und die maximale Qubit-Kettenlänge polynomiell mit den Variablen skalieren. Grundsätzlich ist es möglich einen vollständigen Graphen der Größe 64 direkt auf den D-Wave 2000Q Chimera-Graphen abzubilden. Jedoch ist aus Tabelle 3.3 ersichtlich, dass bei bereits 24 vollvermaschten logischen Variablen des Q-Nash Ising-Hamiltonians der D-Wave 2000Q Quantum Annealer nicht mehr in der Lage war, den Grundzustand zu finden. Obwohl die Ergebnisse in Bezug auf das Näherungsverhältnis r aus Tabelle 3.3 zwar vielversprechend sind, ist im Falle der hier betrachteten Problemstellung eine gute Annäherung ungenügend.

#### 3.4.3.2 Q-Nash Algorithmus - QBSolv

In diesem Abschnitt wird die Lösungsqualität und die Laufzeit von Q-Nash unter Verwendung der QBSolv Heuristik auf größeren Spielinstanzen untersucht.

		(			
Variablen (Logische Qubits)	р	CX	SX	RZ	Tiefe
9	1	72	27	90	54
9	6	432	117	450	219
12	1	132	36	138	72
12	6	792	156	708	282
15	1	210	45	195	90
15	6	1260	195	1020	345
18	1	306	54	261	108
18	6	1836	234	1386	408
21	1	420	63	336	126
21	6	2520	273	1806	471
24	1	552	72	420	144
24	6	3312	312	2280	534

Tabelle 3.4: Skalierungsverhalten des Q-Nash QAOA-Schaltkreises für p = 1und p = 6 unter Verwendung der physikalisch implementierbaren Gatter des IBMQ-27 Ehningen Rechners.

Die Spielinstanzen wurden, wie in Abschnitt 3.4.2.2 beschrieben, zufällig für verschiedene Topologien erstellt und belaufen sich auf Spiele von 62 bis zu 1476 Variablen bzw. beste Antwortstrategien pro Spiel. Zum Vergleich, in den vorherigen Q-Nash Experimenten wurden lediglich Spiele mit bis zu maximal 24 Variablen untersucht.

Abbildung 3.5 zeigt nun die Lösungsqualität der drei in Abschnitt 3.4.2.1 beschriebenen Methoden (Q-Nash mit QBSolv, BF und IBR) in Bezug auf das Finden reiner Nash-Gleichgewichte in unterschiedlich strukturierten graphischen Spielen. Für jede Graphtopologie (Baum, Kreis und Straße) wurden Spiele mit 6 bis 30 Spielern verwendet. Da die Straßentopologie eine gerade Anzahl von Spielern benötigt, wurden die Instanzen mit 15, 21 und 27 Spielern vernachlässigt. Sowohl Q-Nash als auch IBR wurde 20 Mal pro Spielinstanz ausgeführt. Um einen fairen Vergleich zu ermöglichen, wurde die klassische IBR genau so lange ausgeführt, wie die benötigte Lösungs- bzw. Laufzeit des Q-Nash Algorithmus. Der QBSolv Parameter *Num Repeats* wurde für alle Ausführungen auf 200 gesetzt.

Variablen (Logische Qubits)	Physikalische Qubits	Maximale Kettenlänge
9	27	4
12	48	5
15	72	6
18	111	8
21	139	8
24	186	9

Tabelle 3.5: Skalierung des Hardwareembeddings für unterschiedlich große Spielinstanzen auf den Chimera-Graphen des D-Wave 2000Q Quantum Annealers.



(c) Straßentopologie

(d) Num\_Repeats-Parameter

Abbildung 3.5: In 3.5b, 3.5a und 3.5c ist die Lösungsqualität von Q-Nash mit QBSolv, IBR und BF auf unterschiedlich großen graphischen Spielinstanzen dargestellt. Die Spiele basieren auf der Baum-, Kreis- und Straßentoplogie. In 3.5d ist der Einfluss des Num Repeats Parameters in Bezug auf das Finden von reinen NE beispielhaft an einem Spiel mit 24 Spielern und Straßentopologie visualisiert.

Die Ergebnisse zeigen, dass Q-Nash mit QBSolv in allen Ausführungen für die kleineren Spielinstanzen (6 bis 12 Spieler, mit Ausnahme der 12 Spieler Instanz der Baumtopologie) immer die gleiche Anzahl an reinen NE fand wie der exakte BF-Algorithmus, während die klassische IBR nur alle reinen NEs in der Kreistopologie für diese Instanzen fand. In Bezug auf die größeren Spielinstanzen ist zu beobachten, dass Q-Nash mit QBSolv nicht immer in der Lage war, die gleiche Anzahl an reinen NE pro Ausführung zu finden (vgl. 95% Konfidenzintervall). Wenn man die Ergebnisse dem exakten BF-Algorithmus gegenüberstellt, ist zudem zu beobachten, dass Q-Nash auch nicht immer alle reinen NE pro Spiel finden konnte. Im Vergleich zu IBR ist festzustellen, dass Q-Nash im Allgemeinen bei der Kreistopologie besser abschneidet als IBR, während IBR die Spiele der Baumtopologie besser absolviert als Q-Nash. Bei der Straßentopologie schneidet Q-Nash bei Instanzen mit 12 und 18 Spielern besser ab, während IBR Q-Nash bei Instanzen mit 24 und 30 Spielern übertraf. Wie bereits in Abschnitt 3.4.2.1 erwähnt, zerlegt QBSolv den Ising-Hamiltonian bzw. die QUBO-Formulierung in kleinere sogenannte subQUBOS mit einer vordefinierten Größe, die dann unabhängig voneinander auf dem D-Wave Quantum Annealer gelöst und mittels einer klassischen Tabu-Suche zusammengefügt werden. Dieser Prozess wird iterativ solange ausgeführt und durch den QBSolv-Parameter Num Repeats definiert, bis keine Verbesserung in der Lösungsqualität mehr erzielt wird. Der Parameter bestimmt, wie oft die Aufteilung der QUBO-Formulierung wiederholt werden soll, nachdem eine bessere Lösung mit niedrigerer Energie gefunden wurde. In Abbildung 3.5d ist der Einfluss dieses Parameters dargestellt. Zur Veranschaulichung wurde beispielhaft ein Spiel der Straßentopologie mit 24 Spielern verwendet und Q-Nash mit QBSolv 20 Mal pro Parametereinstellung ausgeführt. Wie erwartet ist zu sehen, dass mit zunehmender Anzahl der Wiederholungen (Num Repeats) der Interquartilsbereich und der Median in Bezug auf die Anzahl der gefundenen reinen NE zunehmen. Obwohl eine Ausführung des Annealing-Prozesses standardmäßig nur  $20\mu s$  dauert, summieren sich die Ausführungszeiten entsprechend mit dem Parameter Num Repeats auf und beschreiben den Trade-off zwischen Rechenzeit und Lösungsqualität.

Laufzeitbetrachtung von Q-Nash mit QBSolv Im Hinblick auf die Laufzeitergebnisse in Tabelle 3.6 werden zunächst die einzelnen Komponenten der *Gesamtrechenzeit (1)* von QBSolv vorgestellt. In Abbildung 3.6 ist ein Überblick über die einzelnen Zeitkomponenten gegeben.

Komponente (2) umfasst die Zeit, die die klassische Phase von Q-Nash benötigt, um die besten Antwortstrategien eines jeden Spielers zu identifizieren und zusätzlich das QUBO-Problem zu konstruieren. Die Komponente (3) gibt die Zeit an, die die klassische Embedding-Heuristik von D-Wave benötigt, um ein gültiges subQUBO-Hardwareembedding zu finden. Komponenten (4) und (5)umfassen zum einen die klassische Berechungs- und zum anderen die Quantum Annealing-Zeit. Dabei beinhaltet (4) nicht nur die Tabu-Suche, die iterativ al-



Abbildung 3.6: Überblick über die Laufzeitkomponenten von Q-Nash mit QB-Solv.

le subQUBOs verarbeitet, sondern auch gegebenenfalls Latenz- und Wartezeit für die D-Wave Quantum Annealer, die in der Cloud zur Verfügung stehen. Die Quantum Annealing-Zeit (5) besteht aus der sogenannten QPU-Zugriffszeit, die pro subQUBO aufsummiert wird.

Laufzeitergebnisse von Q-Nash mit QBSolv Die Laufzeitergebnisse werden beispielhaft anhand graphischer Spiele der Kreistopologie gezeigt und sind in Tabelle 3.6 aufgelistet. Die verwendeten Spielinstanzen unterscheiden sich in der Anzahl der Spieler (6-20 Spieler) und die Rechenzeiten des Q-Nash Algorithmus mit QBSolv sind in Sekunden angegeben.

Aufgrund des relativ großen Overheads von QBSolv (Hardwareembedding, klassische Tabu-Suche, Latenz, Wartezeit, etc.) ist die Gesamtzeit (1) selbst für die kleineren Spielinstanzen (6 und 12 Spieler) verhältnismäßig groß (85.1 und 125.8 Sekunden). Nichtsdestotrotz ist die eigentliche Q-Nash Rechenzeit ((2) plus (5)) vergleichsweise gering (1.2 und 1.6 Sekunden). Im Allgemeinen ist zu sehen, dass die Quantum Annealing-Zeit (5) für diese Spielinstanzen konstant ist und die klassische Q-Nash Phase (2) eine polynomielle Laufzeit besitzt, wie bereits in Abschnitt 3.4.1 angegeben. Da das Hardwareembedding (3) für eine feste vordefinierte Graphengröße konstant ist, ist die einzige unvorhersehbare Laufzeitkomponente von Q-Nash Komponente (4). Dies ist auf die Tabu-Suche (einschließlich des Parameters Num Repeats) zurückzuführen, die erst nach einer definierten Anzahl an Iterationen endet, in denen keine Verbesserung zu der zuvor gefundenen Lösung erzielt werden konnte.

Diese Beobachtungen der Laufzeitkomponenten lassen schließen, dass mit zunehmender Skalierung der Hardwareressourcen und unter Voraussetzung, dass dann Q-Nash ohne QBSolv für relevante Probleminstanzen ausgeführt werden kann, der Ansatz vermutlich eine konkurrenzfähige Alternative zu rein klassischen Methoden darstellt. Abschließend sei jedoch noch erwähnt, dass

	Graphisches Spiel (Kreistopologie)							
Q-Nash mit QBSolv Laufzeiten (sek)	6 Spieler	8 Spieler	10 Spieler	12 Spieler	14 Spieler	16 Spieler	18 Spieler	20 Spieler
(1) Gesamtzeit	85.1	125.6	457.7	424.5	273.1	228.5	276.4	248.3
(2) Bestimmung der besten Antwortstrategien (klassisch)	0.3	0.8	1.8	3.0	4.9	7.3	10.7	14.3
(3) Finden eines Hardwareembeddings (klassisch)	4.1	4.0	4.0	5.1	4.8	5.7	4.2	4.3
(4) QBSolv (klassisch)	79.8	120.0	448.8	412.8	260.2	212.8	258.3	226.7
(5) QA Zeit (quantum)	0.9	0.8	3.1	3.6	3.2	2.7	3.2	3.0

Tabelle 3.6: Laufzeiten des Q-Nash-Algorithmus mit QBSolv bei verschiedenen graphischen Spielen der Kreistopologie.

sowohl Quantum Annealing als auch QBSolv beides Heuristiken sind für die die Laufzeit im komplexitätstheoretischen Sinne unbekannt ist. Daher sind die Aussagen in diesem Abschnitt mit Vorsicht zu genießen und lediglich Beobachtungen, die auf den gemessenen Laufzeiten basieren.

#### 3.4.4 Zusammenfassung und Ausblick

In diesem Unterkapitel wurde Q-Nash, ein hybrider Hamiltonian-Ansatz, zur Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen vorgestellt. Der Ansatz ist sowohl auf gatter- als auch annealingbasierter Hardware mittels QAOA beziehungsweise QA ausführbar.

Zu Beginn wurde Q-Nash mit vergleichsweise kleinen Spielinstanzen (9 bis 24 Variablen) auf D-Wave 2000Q Quantum Annealer und IBMQ-27 Ehningen Rechners bzw. IBMQ QASM Simulator evaluiert. Die Ergebnisse zeigten, dass Q-Nash auf D-Wave Hardware zum aktuellen Zeitpunkt in Bezug auf die Lösungsqualität (GZ-Wahrscheinlichkeit und Näherungsverhältnis) der IBMQ Hardware und Simulator überlegen war. Das Skalierungsverhalten ist sowohl bei Q-Nash unter Verwendung des QAOA als auch unter Verwendung der D-Wave Quantum Annealing Heuristik ähnlich. Während die physikalischen Qubits von Q-Nash auf dem D-Wave Chimera-Graphen polynomiell mit der Anzahl an Variablen skalieren, würde die Schaltkreistiefe von Q-Nash mit dem QAOA bei einer vollvermaschten QPU linear in der Anzahl an alternierenden Operatoren skalieren. Zusätzliche SWAP-Gatter, die bei einer nicht vollvermaschten Hardware benötigt werden, wurden allerdings nicht berücksichtigt, was letztendlich vermutlich zu einer ebenfalls polynomiellen Skalierung der Schaltkreistiefe auf aktueller NISQ-Hardware führt.

Da in aktuellen Zeiten der NISQ-Ära die Hardware in Bezug auf die Anzahl der Qubits und deren Konnektivität noch immer sehr begrenzt ist, musste für die vergleichsweise großen und relevanten graphischen Spiele auf die quantenklassische hybride Software QBSolv zurückgegriffen werden. Der zusätzliche Overhead dieser Software macht es schwierig, einen fairen Vergleich zwischen Q-Nash und klassischen modernen Lösungsmethoden hinsichtlich der Laufzeit zu ziehen. Daher wurde dessen Gesamtzeit lediglich in ihre Laufzeitkomponenten aufgeschlüsselt, um deren jeweiligen Einfluss aufzuzeigen. Den experimentellen Ergebnissen zufolge ist für die verwendeten Spielinstanzen die einzige unvorhersehbare Zeitkomponente die klassische Tabu-Suche von QBSolv, zusammen mit den Latenz- und Wartezeiten, die in der D-Wave Cloud auftreten können. Hinsichtlich der Effektivität von Q-Nash mit QBsolv wurde gezeigt, dass der Algorithmus für die Spielinstanzen zwischen 6 und 12 Spielern immer in der Lage war, alle reinen NE in unterschiedlich strukturierten graphischen Spielen zu finden. Allerdings nahm, wie erwartet, mit steigender Spielerzahl auch die Varianz in Bezug auf die gefundenen reinen NE zu.

Im Hinblick auf zukünftige Arbeiten wäre es interessant zu sehen, welche Auswirkung eine umfangreiche Hyperparameteroptimierung der einzelnen Methoden (QAOA und QA) auf die Lösungsqualität der jeweiligen Hardware haben würde. Solch eine Optimierung war allerdings nicht das Hauptaugenmerk dieses Kapitels, weshalb in den vorliegen Experimenten lediglich die voreingestellten Parameter des jeweiligen Hardwareherstellers verwendet wurden. Zahlreiche verwandte Arbeiten zeigen jedoch auf, dass mit entsprechenden Optimierungen eine Verbesserung der Lösungsqualität erzielt werden kann [56, 84, 166, 180].

## 3.5 Grover-Ansatz zur Findung von reinen Nash-Gleichgewichten

In den vorherigen Abschnitten wurde der hybride Q-Nash Ansatz vorgestellt, der, obwohl er hinsichtlich der Lösungsqualität mit klassischen Verfahren mithalten kann, doch eine Metaheuristik ist, für die der Quantenvorteil in Bezug auf die Laufzeit unbekannt ist.

Daher wird nun in den nachfolgenden Abschnitten der Grover-Algorithmus, für den eine quadratische Beschleunigung bei der Suche in einer unsortierten Datenbank nachgewiesen wurde [79], untersucht und geklärt, ob dieser für das Finden von reinen Nash-Gleichgewichten in graphischen Spielen von Vorteil ist. Seit der Einführung im Jahr 1996 wurde der Grover-Algorithmus und die ihm zugrundeliegende Amplitudenverstärkung mehrfach untersucht und auf verschiedene Problemklassen angewendet [64, 110, 115]. Darunter befindet sich auch das NP-vollständige Erfüllbarkeitsproblem der Aussagenlogik (engl. Boolean Satisfiability Problem (SAT)), das eine boolesche Zuordnung zu Variablen sucht, die eine gegebene boolesche Formel erfüllt [31, 62]. Es ist zwar bekannt, dass der Grover-Algorithmus nicht in der Lage ist, klassische Methoden für k-SAT-Probleme zu übertreffen [82], bei denen jede Klausel auf höchstens drei Literale oder Variablen ( $k \leq 3$ ) beschränkt ist, aber eine Beschleunigung bei k > 3 könnte möglich sein [35].

Im Nachfolgenden wird ein Grover-basierter Quantenalgorithmus zum Finden von PNE in graphischen Spielen vorgestellt und im Weiteren als *Grover-Nash* 

bezeichnet. Die Komplexität des Ansatzes aus Anwendersicht liegt in der Konstruktion der Orakelfunktion. Dazu wird im ersten Schritt ein graphisches Spiel und dessen Eigenschaften in ein SAT-Problem übersetzt, aus dem anschließend der entsprechende Orakel-Quantenschaltkreis entworfen werden kann. Von besonderem Interesse ist die Abschätzung der Skalierbarkeit des Algorithmus für Spiele unterschiedlicher Größe und die damit verbundene Anwendbarkeit in realen Szenarien.

#### 3.5.1 Konzept

Das Orakel ist eine Funktion, die bestimmt, ob eine Eingabe eine Lösung für das entsprechende Problem ist. Das globale Strategieprofil eines graphischen Spiels mit  $n = C_{\mathcal{B}}$  gesamten besten Antwortstrategien kann durch einen *n*dimensionalen Vektor dargestellt werden, dessen Einträge entweder 0 oder 1 sind und somit insgesamt  $2^n$  mögliche Zustände im Konfigurationsraum zulässt.

Die Orakelfunktion für die Suche nach reinen Nash-Gleichgewichten muss 1 oder 0 zurückgeben, wenn der Eingabevektor einen PNE-Zustand darstellt, beziehungsweise nicht darstellt. Mit diesem Orakel wird das Finden von PNE zu einem unstrukturierten Suchproblem. Unter Verwendung des Grover-Algorithmus und dem Orakel als Eingabefunktion ist es möglich im Konfigurationsraum des Problems nach Zuständen zu suchen, die das Orakel erfüllen und somit PNE-Zuständen entsprechen.

In den nachfolgenden Unterabschnitten wird das Orakel für das Problem der Suche nach PNE in graphischen Spielen beschrieben und die Transformation in ein boolesches Erfüllbarkeitsproblem formuliert. Sobald das Orakel konstruiert ist, wird der Grover-Algorithmus mit dem Verfahren der Amplitudenverstärkung angewendet, um die PNE-Zustände zu finden. Der grundsätzliche Ablauf des Grover-Algorithmus wurde bereits im Grundlagenkapitel 2.4.2.2 beschrieben.

#### 3.5.1.1 Das Grover-Nash Orakel als Erfüllbarkeitsproblem

Das Erfüllbarkeitsproblem ist ein NP-vollständiges Entscheidungsproblem, das danach fragt, ob eine aussagenlogische boolesche Formel  $f(x_1, ..., x_n)$  mit Variablen  $x_1..., x_n \in \{0, 1\}$  erfüllbar ist. Das heißt, ob es eine Zuordnung zu den booleschen Variablen gibt, die dazu führt, dass die gesamte Formel auf das logische *True* reduziert werden kann. Eine gängige Art, eine solche boolesche Formel darzustellen, ist die konjunktive Normalform (engl. Conjunctive Normal Form (CNF)). Sie besteht aus einer Konjunktion ( $\wedge$ ) von Klauseln, wobei jede Klausel eine Disjunktion ( $\vee$ ) von Literalen ist. Dabei ist jedes Literal eine Variable oder ihre Negation ( $\neg$ ). In Gleichung (3.7) ist ein Beispiel für eine boolesche Formel  $f(x_1, x_2, x_3)$  in ihrer CNF gegeben, die aus zwei konjunkten Klauseln mit jeweils zwei disjunkten Literalen besteht.

$$f(x_1, x_2, x_3) = (x_1 \lor x_3) \land (x_3 \lor \neg x_2) \tag{3.7}$$

Die CNF kann nun zur Beschreibung der Orakelfunktion des Grover-Algorithmus verwendet werden. Die Bedingungen, die ein Zustand für ein PNE erfüllen muss, können in zwei Kategorien unterteilt werden: Die Inter-Bedingungen und Intra-Bedingungen. Das Orakel  $\mathcal{O}$  kann somit ebenfalls in zwei Teilorakel  $\mathcal{O}_{Inter}$  und  $\mathcal{O}_{Intra}$  aufgeteilt werden, die beide erfüllt sein müssen.

$$\mathcal{O} = \mathcal{O}_{Intra} \wedge \mathcal{O}_{Inter} \tag{3.8}$$

**Intra-Bedingungen** Die Intra-Bedingungen stellen sicher, dass jeder Spieler nur eine einzige Aktion aus seiner Menge der besten Antwortstrategien spielt. Dies wird dadurch erreicht, dass zunächst alle verschiedenen Kombinationen von zwei Aktionen in der Menge der besten Antwortstrategien eines bestimmten Spielers mit NOT-AND-Gattern (NAND-Gatter,  $\bar{\Lambda}$ ) und alle Paare mit AND-Gattern ( $\Lambda$ ) verknüpft werden. Der daraus resultierende boolesche Ausdruck liefert nur dann *True*, wenn eine der Eingaben *True* ist und alle anderen *False* sind, mit der einzigen Ausnahme, wenn alle Eingaben *False* sind. Diese Ausnahme kann jedoch ausgeschlossen werden, indem mit einem UND-Gatter eine Klausel angehängt wird, in der alle besten Antwortstrategien des Spielers mit OR-Gattern ( $\vee$ ) verknüpft sind. Dieser boolesche Ausdruck der Intra-Bedingungen garantiert, dass die Ausgabe nur dann *True* liefert, wenn der Spieler genau eine seiner besten Antwortstrategien spielt. Beispielsweise entsprechen die Intra-Bedingungen, für einen Spieler *A* mit einer Menge von drei besten Antwortstrategien { $x_1, x_2, x_3$ }, dem folgenden booleschen Ausdruck:

$$\mathcal{O}^A_{intra} = (x_1 \bar{\wedge} x_2) \wedge (x_1 \bar{\wedge} x_3) \wedge (x_2 \bar{\wedge} x_3) \wedge (x_1 \vee x_2 \vee x_3) \tag{3.9}$$

Das vollständige Orakel  $\mathcal{O}_{intra}$  für das graphische Spiel erhält man, indem man die einzelnen Orakel für die Intra-Bedingungen eines jeden Spielers mit AND-Gattern verknüpft:

$$\mathcal{O}_{intra} = \mathcal{O}^A_{intra} \wedge \mathcal{O}^B_{intra} \wedge \dots \wedge \mathcal{O}^N_{intra}$$

Inter-Bedingungen Im Gegensatz zu den Intra-Bedingungen, die nur von der Anzahl der besten Antwortstrategien jedes Spielers abhängen, werden bei den Inter-Bedingungen die besten Antwortstrategien jedes Spielers mit denen der Nachbarspieler verglichen. Daher muss hierfür die Topologie des graphischen Spiels berücksichtigt werden. Angenommen, ein globales Strategieprofil ist ein PNE, dann muss es ein Element in der Schnittmenge aller Mengen der besten Antwortstrategien aller Spieler sein. Andernfalls würde dies bedeuten, dass mindestens zwei Spieler kein Strategieprofil haben, das jeweils ihre Auszahlung maximiert und auch mit dem Strategieprofil der anderen Spieler kompatibel ist. Es gibt keine generische Form für den booleschen Ausdruck der Inter-Bedingungen, da man ihn aus der Topologie des jeweiligen Spiels konstruieren muss.

Das Orakel für die Inter-Bedingungen  $\mathcal{O}_{inter}$  wird durch einen Ausschlussprozess konstruiert. Für zwei im Graph verbundene Spieler muss man die besten Strategieprofile ausschließen, die nicht miteinander kompatibel sind. Dies geschieht durch die Verwendung von Klauseln, die aus der Negation der beiden Variablen bestehen, die den inkompatiblen Strategien entsprechen und mit einem OR-Gatter verbunden sind. Wenn z.B.  $x_1$  und  $y_2$  inkompatible beste Antworten von zwei verbundenen Spielern A und B sind, ist die Klausel, die diese Antworten aus der Menge der möglichen Lösungen ausschließt, durch den folgenden booleschen Ausdruck gegeben:  $(\neg x_1 \lor \neg y_2)$ . Diese Klausel ist nur dann *False*, wenn sowohl  $x_1$  als auch  $y_2$  *True* sind. Somit schließt diese Klausel aus, dass zwei inkompatible Strategien gleichzeitig gespielt werden können.

Schließlich erhält man die vollständige Grover-Nash Orakelfunktion als SAT-Problem, indem man die zuvor beschriebenen Teilorakel der Inter- und Intra-Bedingungen mit AND-Gattern, wie in Gleichung (3.8) angegeben ist, verknüpft.

#### 3.5.1.2 Das Grover-Nash Orakel in konjunktiver Normalform

Um mit Hilfe allgemeiner Quantenlogik-Synthesemethoden einen Quantenschaltkreis aus einer booleschen Funktion zu konstruieren, muss sie zunächst in ihre CNF umgewandelt werden.

Das Grover-Nash Orakel der Inter-Bedingungen liegt bereits in ihrer CNF vor, wie aus der Beschreibung in Abschnitt 3.5.1.1 hervorgeht. Für das Orakel der Intra-Bedingungen ist dies jedoch noch nicht der Fall, da die Literale innerhalb der Klauseln aktuell mit NAND-Gattern verknüpft sind, siehe Gleichung (3.9). Mit der nachfolgenden äquivalenten Formulierung wird das NAND-Gatter beseitigt:

$$(x_1 \overline{\land} x_2) \land (x_1 \lor x_2) = (x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2)$$

Diese Formulierung kann auf eine beliebige Anzahl an Variablen erweitert werden. Unter Verwendung dieser Äquivalenz ist der boolesche Ausdruck in CNF für das Orakel in Gleichung (3.9) gegeben durch:

$$\mathcal{O}^A_{intra} = (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_3) \land (\neg x_2 \lor \neg x_3)$$

Mit dieser Übersetzung kann das vollständige Grover-Nash Orakel nun in seiner konjunktiven Normalform beschrieben werden.

Alle obigen theoretischen Überlegungen lassen sich ohne Weiteres in ein Verfahren zur Konstruktion des Quantenschaltkreises des Grover-Nash Orakels für ein graphisches Spiel einbetten, das nachfolgend beschrieben wird.

## 3.5.2 Aufbau der Experimente

Wie bei dem Hamiltonian-Ansatz Q-Nash wird bei Grover-Nash das gleiche klassisches Pre-Processing durchgeführt, um die besten Antwortstrategien in polynomieller Zeit aus der Tabellen- bzw. Matrixdarstellung der graphischen Spiele zu berechnen. Nach der Bestimmung der besten Antwortstrategieprofile der einzelnen Spieler und der Konstruktion der Inter- und Intra-Bedingungen mit Hilfe der booleschen Ausdrücke in konjunktiver Normalform, wie in Abschnitt 3.5.1 beschrieben, erhält man die boolesche Funktion für das Grover-Nash Orakel in seiner CNF. Diese definiert dann das k-SAT-Problem, bei dem jede beste Antwortstrategie durch eine boolesche Variable dargestellt wird. Der nächste Schritt besteht darin, die boolesche Funktion in einen Quantenschaltkreis umzuwandeln, die dann mit Grovers Amplitudenverstärkung zur Lösung des booleschen SAT-Problems verwendet werden kann. Hierfür wurde die im IBM *Qiskit SDK* eingebauten Funktionen verwendet [5]. Man kann den booleschen Ausdruck des SAT-Problems in ein Quantenorakel umwandeln, indem man die Klasse *PhaseOracle* verwendet. Es verwendet standardmäßig eine Drittanbieterbibliothek namens Tweedledum zur Analyse, Synthese und Optimierung von Quantenschaltungen [155]. Tweedledum konvertiert eine Funktion in CNF mittels der Pseudo-Kronecker Reed Muller (PKRM) Synthese in einen Quantenschaltkreis, um den booleschen Ausdruck zu verarbeiten und das entsprechende Quantenorakel zu konstruieren [55]. Dabei ist zu beachten, dass die PKRM-Synthese jeder Variablen des k-SAT-Problems ein Qubit im Quantenschaltkreis zuordnet. Daher ist die Anzahl der Qubits in der Quantenschaltung für das Grover-Nash Orakel gleich der Anzahl der besten Antwortstrategien im Spiel.

Nachdem man den Quantenschaltkreis für das Orakel erhalten hat, kann man den in Qiskit integrierten Algorithmus zur Amplitudenverstärkung verwenden, um die Grover-Iterationen auszuführen.

## 3.5.3 Ergebnisse und Diskussion

Da die boolesche SAT-Formulierung des vorgeschlagenen Grover-Nash Orakels exakt ist, konzentrieren wir uns hier hauptsächlich auf die Skalierbarkeit des Quantenschaltkreises und damit auf seine Anwendbarkeit in Bezug auf das Finden von reinen Nash-Gleichgewichten in graphischen Spielen. Um die vorgestellte Orakelfunktion in Bezug auf Breite und Tiefe zu analysieren, wurden zufällige Spielinstanzen mit drei bis vier Spielern erzeugt, die nach dem Pre-Processing in 9 bis 18 beste Antwortstrategiemengen resultierten. Zudem wurde bei der Erstellung der Spielinstanzen sichergestellt, dass sie sich in der Anzahl der PNE unterscheiden, um deren Einfluss auf die Schaltkreistiefe der Orakelfunktion zu untersuchen.

Die Ergebnisse bezüglich der Skalierbarkeit von Grover-Nash ist in der Tabelle 3.7 dargestellt. Wie in Abschnitt 3.5.2 beschrieben, hängt die Breite bzw. die Anzahl der für den Algorithmus benötigten Qubits von der Anzahl der besten

Antwortstrategien ab. Die Ergebnisse zeigen, dass die Tiefe des Schaltkreises exponentiell mit der Anzahl der Qubits bzw. der Anzahl der besten Antwortstrategien und linear mit der Anzahl der PNE skaliert. Das ermittelte exponentielle Wachstum ist von entscheidender Bedeutung, da die Kohärenzzeit von NISQ-Hardware sehr begrenzt ist. Leymann et al. geben eine Faustregel für die Anwendbarkeit von Quantenalgorithmen an, die besagt, dass die Schaltkreistiefe multipliziert mit der Schaltkreisbreite sehr viel kleiner als  $\frac{1}{Fehlerrate}$ sein muss. Wobei die Fehlerrate die Dekohärenzzeiten der Qubits, die Präzision und die Fehlerhäufigkeit der Gatter umfasst [106]. Auch wenn man in der Regel die Schaltkreistiefe durch den Einsatz zusätzlicher Hilfs-Qubits, sogenannten Ancillae, verringern kann [38, 94, 113], geht dies immer auf Kosten des jeweils anderen. Dies beschreibt im Wesentlichen den Kompromiss zwischen Schaltkreistiefe und -breite in der Quantenschaltkreissynthese und ist selbst ein vielfach untersuchtes Forschungsgebiet [9, 66]. In Bezug auf die Er-

			(			
Spieler	PNE	$Qubit_S$	CX	Х	RZ	Tiefe
3	1	9	764	12	765	1.275
3	2	9	1.528	16	1.530	2.550
3	3	9	2.292	22	2295	3.825
4	1	12	6.140	16	6.141	10.236
4	2	12	12.280	24	12.282	20.471
4	3	12	18.420	28	18.423	30.705
3	1	15	49.148	24	32.766	81.916
3	2	15	98.296	30	98.298	163.830
3	3	15	147.444	32	147.447	245.745
4	1	18	393.212	28	393.213	655.356
4	2	18	786.424	36	786.426	1.310.710
4	3	18	1.179.636	44	1.179.639	1.966.064

Tabelle 3.7: Schaltkreisbreite und -tiefe der Grover-Nash Orakelfunktion für unterschiedlich große graphische Spielinstanzen.

gebnisse ist zusätzlich anzumerken, dass die evaluierten Schaltkreise zwar ausschließlich aus physikalisch implementierbaren IBM-Quantengattern (CX-, RZ- und X - Gatter) bestehen, jedoch noch nicht die Konnektivität realer Quantenhardware berücksichtigen. Das bedeutet, dass bei der Transpilierung dieser Schaltkreise auf einer realen NISQ-Hardware, die nicht über vollständig verbundene Qubits verfügt, zusätzliche SWAP-Gatter verwendet werden müssen. Diese müssen wiederum durch drei CX-Gatter abgebildet werden, was zum einen zu einer größeren Tiefe des Schaltkreises beiträgt und zum anderen die durchschnittliche Fehlerrate steigen lässt, die durch jedes zusätzliche Gatter zunimmt. Somit lässt sich zusammenfassen, dass der Grover-Nash Ansatz kaum anwendbar für relevante graphische Spiele auf aktueller NISQ-Hardware ist.

#### 3.5.4 Zusammenfassung und Ausblick

In diesem Abschnitt wurde ein weiterer hybrider Ansatz zum Finden reiner Nash-Gleichgewichte in graphischen Spielen vorgestellt. Der auf dem Grover-Algorithmus basierende Ansatz Grover-Nash ist in der Lage, alle reinen NE in einem graphischen Spiel zu berechnen. Der Algorithmus ist keine Heuristik wie der zuvor vorgestellte Q-Nash Ansatz und lässt sich zudem auf jede beliebige Graphtopologie anwenden. Trotz der Verwendung des Grover-Algorithmus, der eine quadratische Beschleunigung verspricht, hat Grover-Nash jedoch immer noch eine exponentielle Laufzeit und ist daher nur für graphische Spiele begrenzter Größe von Bedeutung. Zudem muss berücksichtigt werden, dass bei Spielinstanzen, bei denen die Menge der Lösungen bzw. PNE unbekannt ist, die Anzahl der Grover-Iterationen entweder zufällig ausgewählt oder mit dem Quantum Counting Algorithm geschätzt werden müssen [1, 24], was zusätzlichen Overhead bei der Lösungsfindung verursacht. Weitere Untersuchungen zur Skalierbarkeit durch Einbindung von Ancillae-Qubits, wodurch die Schaltkreistiefe verringert werden kann oder effizienteren Quantenlogik-Synthesemethoden müssen in Erwägung gezogen werden [178].

## 3.6 Kapitelzusammenfassung

Dieses Kapitel widmete sich der Entwicklung und Evaluation hybrider Algorithmen zur Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen unter Berücksichtigung verschiedener Quantum Computing Hardwarearchitekturen. Daher wurde zum einen die Fragestellung untersucht, welche Möglichkeiten zur Eingliederung von annealingbasierter als auch gatterbasierter Quantum Computing Hardware zur Lösung dieser Problemstellung bestehen und zum anderen, welchen Reifegrad die verschiedenen Ansätze zum aktuellen Zeitpunkt auf der NISQ-Hardware besitzen.

In Abschnitt 3.4 wurde der hybride Hamiltonian-Ansatz Q-Nash vorgestellt. Dieser berechnet in einer klassischen Preprocessing Phase die besten Antwortstrategien der Spieler, um im Anschluss, den Grundzustand bzw. die Grundzustände reiner NE, mittels eines Hamiltonians, zu identifizieren. Der hybride Ansatz ist sowohl mit dem QAOA als auch mit QA ausführbar. Die Ergebnisse haben gezeigt, dass Q-Nash auf der D-Wave Hardware mit klassischen Algorithmen konkurrieren kann und mit zunehmender Quantum Computing Hardwarereife eine Alternative zum Lösen dieser Problemstellung bietet.

Anschließend wurde in Abschnitt 3.5 die Konstruktion einer Orakelfunktion zur Findung von reinen NE in graphischen Spielen vorgestellt, die mittels dem Grover-Algorithmus und seiner Amplitudenverstärkung angewendet werden kann. Im Gegensatz zum Q-Nash Ansatz, der eine Heuristik ist, ist Grover-Nash eine exakte Methode. Trotz der quadratischen Beschleunigung des Grover-Algorithmus, besitzt Grover-Nash immer noch eine exponentielle Laufzeit. Das wiederum macht ihn nur für sehr große Spielinstanzen, bei den die klassischen Methoden an ihre Grenzen stoßen, relevant. Allerdings ist anhand der exponentiellen Skalierung des Schaltkreises ersichtlich, dass Grover-Nash wenn überhaupt erst in ferner Zukunft auf Quantum Computing Hardware anwendbar sein wird.

# 4 Penalty-Parameter Optimierung eingeschränkter Hamiltonians

In Kapitel 3 wurde anhand eines speziell ausgewählten Anwendungsfall ersichtlich, dass die Lösungsqualität und Anwendung entsprechender Algorithmen auf NISQ-Hardware noch immer sehr begrenzt ist, durch die aktuellen Hardwareressourcen, wie der Anzahl der Qubits, deren Rauschverhalten und deren Konnektivität auf der jeweiligen QPU. Zudem wurde festgestellt, dass zum aktuellen Zeitpunkt die Berechnung von reinen NE in graphischen Spielen und dessen zugrundeliegende diskrete Optimierungs- beziehungsweise Suchproblem in Form eines Hamiltonian-Ansatzes vorteilhafter ist, gegenüber einer entsprechenden Orakelfunktion, die mit der Amplitudenverstärkung gelöst werden kann. Nichtsdestotrotz besteht selbst bei dem vergleichsweise besser abschneidenden Hamiltonian-Ansatz Handlungsbedarf, um die Lösungsqualilät zu steigern.

Diese Erkenntnisse führen daher in Kapitel 4 zur Untersuchung mehrerer NPschwerer Hamiltonians und zur Möglichkeit, diese effizient ohne zusätzliche Quantum Computing Hardwareressourcen zu optimieren. Schließlich wird die Kreuzentropie-Methode vorgestellt, die auf die Penalty-Parameter, die die Nebenbedingungen eines eingeschränkten Hamiltonians gewichten, angewendet werden kann.

Neben der Motivation und Zielsetzung in Abschnitt 4.1 dieses Kapitels wird zunächst in Abschnitt 4.2 die Definition und Grundlagen der Kreuzentropie-Methode eingeführt, bevor in Abschnitt 4.3 verwandte Arbeiten in Bezug auf die Optimierung von Hamiltonians und deren damit verbundenen Quantenalgorithmen, wie die QA Heuristik und der QAOA, vorgestellt werden.

Abschnitt 4.4 beschreibt dann das Konzept der Kreuzentropie-Methode zur Optimierung der Penalty-Parameter eingeschränkter Hamiltonians in Kombination mit gatterbasierten als auch annealingbasierten Quantenalgorithmen.

In Abschnitt 4.5 wird die Anwendung der Optimierungsmethode unter Verwendung variationeller Algorithmen wie dem QAOA evaluiert. Es wird empirisch gezeigt, dass durch die Anwendung der Kreuzentropie der klassische Optimierer des QAOA im Vergleich zum konventionellen Ansatz bessere Gatterparameter findet, was sich wiederum in einer gesteigerten Lösungsqualität widerspiegelt. Abschnitt 4.6 beschreibt die Experimente und Anwendung der Kreuzentropie-Methode in Bezug auf Hamiltonians für das Quantum Annealing. Die Ergebnisse weisen ebenfalls eine signifikant verbesserte Lösungsqualität bei der Ausführung auf Quantum Annealing Hardware auf. Zudem zeigt die numerische Berechnung des Hamiltonian-Eigenspektrums, dass die Lösungsqualität mit einer größeren minimalen Spektrallücke korreliert, welche als Metrik für die Anwendbarkeit von Hamiltonians auf entsprechender Hardware angesehen werden kann.

Beide Abschnitte basieren inhaltlich in sequentieller Reihenfolge auf den Publikationen [141] und [140]. Abschnitt 4.7 fasst das Kapitel und dessen wichtigste Erkenntnisse kurz zusammen. Eine detailliertere Zusammenfassung der vorgestellten Ansätze sowie mögliche Ansatzpunkte für zukünftige Arbeiten befindet sich am Ende eines jeden Abschnitts.

## 4.1 Motivation und Zielsetzung

In den letzten Jahren haben immer mehr Hersteller von Quantum Computer ihre Geräte für Forschung und Industrieanwendungen zur Verfügung gestellt [48, 90]. Dies hat zu einer Vielzahl von Forschungsarbeiten über Quantenalgorithmen geführt, insbesondere über die Entwicklung rauschresistenter Algorithmen, die in naher Zukunft auf Quantencomputern ausgeführt werden können. Wenngleich bereits theoretische Algorithmen existieren, die einen bekannten Quantenvorteil gegenüber den klassischen Algorithmen bieten [79, 158], erfordern diese Algorithmen in der Praxis meist oft Tausende von Gattern [145], die ohne Fehlerkorrektur auf der aktuellen NISQ-Hardware unmöglich zu realisieren sind. Darunter fällt unter anderem auch der Grover-Algorithmus, der in Kapitel 3 für die Berechnung von reinen NE in graphischen Spielen untersucht wurde. Aus diesem Grund wurden in den letzten Jahren variationelle Quantenalgorithmen (VQA) entwickelt, die einen parametrisierbaren Quantenschaltkreis mit geringer Tiefe und einem klassischen Optimierer kombinieren. Diese Architektur macht es möglich, sie auf heutiger NISQ-Hardware anzuwenden. Einer der bekanntesten VQAs ist der bereits in Kapitel 3 vorgestellte QAOA, bei dem der Anfangszustand des Systems in eine gleichmäßige Uberlagerung von Basiszuständen gebracht wird und anschließend durch Anwendung der algorithmusspezifischen Gatter weiterentwickelt wird, mit dem Ziel den Grundzustand eines entsprechenden Problem-Hamiltonians zu finden. Die Leistungsfähigkeit solcher hybriden VQAs hängt einerseits von der klassischen Optimierungsmethode und andererseits von der Sequenz an parametrisierten Gattern ab. Dabei ist eine der größten Schwierigkeiten, effizient in der nichtkonvexen, hochdimensionalen Landschaft der Gatterparameter, die viele qualitativ schlechte und oft nicht entartete lokale Optima enthält, zu optimieren [184].

Es existieren zahlreiche Möglichkeiten und Forschungsarbeiten zu diversen Verbesserungen des QAOA. Meist sind diese Optimierungen auf Algorithmenebene anzutreffen. Beispielsweise kann der QAOA mit zuvor berechneten Gatterparametern initialisiert werden [77, 151] oder anstelle einer gleichverteilten Superposition aus einem klassisch berechneten lokalen Optimum starten [56, 166].

Ziel dieses Kapitels ist es jedoch eine algorithmusunabhängige Optimierungsmethode vorzustellen, die demgegenüber auf Problemebene wirkt und somit auch algorithmusübergreifend angewendet werden kann. Zudem sollte eine solche Optimierung ressourcenschonend sein und möglichst wenige bis gar keine zusätzlichen Qubits, Gatter, oder Couplers der jeweiligen Hardware benötigen. Daher wird in diesem Kapitel die Kreuzentropie-Methode vorgestellt, die auf die Penalty-Parameter eingeschränkter Hamiltonian des jeweiligen eingeschränkten Optimierungsproblems angewendet werden kann.

Die Optimierung auf Problemebene macht es möglich, diese auch für annealingbasierte Architekturen, wie den D-Wave Quantum Annealern, anzuwenden und zu evaluieren. Eine der größten Problematiken bei der D-Wave Quantum Annealing Heuristik, die an dem adiabatischen Theorem angelehnt ist (vgl. Kapitel 2.2.2), ist der Einfluss thermischer Fluktuationen beziehungsweise anderer äußerer Störeinflüsse, die zu einem nicht-adiabatischen Annealing führen. Somit erhöht sich die Wahrscheinlichkeit, dass das System aus dem präparierten Grundzustand des initialen Hamiltonians in einen angeregten Zustand übergeht. Obwohl man bei D-Wave Hardware eine längere Annealing-Zeit definieren kann, um somit theoretisch entsprechend des adiabatischen Theorems ausreichend langsam den Annealing-Prozess durchführen könnte, ist dies in der Praxis aufgrund von sehr kurzen Kohärenzzeiten aktueller Quantensysteme kaum anwendbar. Daher wird auch in Bezug auf die annealingbasierte Hardware die Anwendung der Kreuzentropie-Methode zur Optimierung auf Problemeebene untersucht, um einen möglichen Mehrwert in der heutigen NISQ-Ära zu erzielen.

## 4.2 Definition der Kreuzentropie-Methode

Die Kreuzentropie-Methode (KE-Methode) ist eine vielseitige Heuristik zur Lösung schwieriger Schätz- und Optimierungsprobleme, die auf der Kullback-Leibler- oder Kreuzentropie-Minimierung basiert. Sie wurde von Rubinstein als adaptives Verfahren zur Stichprobenentnahme für die Schätzung von Wahrscheinlichkeiten seltener Ereignisse vorgestellt [150]. Rubinstein macht die Annahme, dass das Finden einer optimalen oder nahezu optimalen Lösung für ein Optimierungsproblem mittels einer Zufallssuche der Wahrscheinlichkeit für ein solches seltenes Ereignis entspricht [149]. Daher ist das Kernprinzip seiner KE-Methode, die Stichprobenverteilung einer Zufallssuche schrittweise so zu verändern, dass das Auftreten seltener Ereignisse wahrscheinlicher wird. Zu diesem Zweck schätzt die Methode, unter Verwendung der Kreuzentropie-Distanz, eine Folge von Stichprobenverteilungen, die dann zu einer Verteilung konvergiert, die in einer Region mit nahezu optimalen Lösungen konzentriert ist. Dieses Verfahren funktioniert bekanntermaßen auch bei Optimierungsproblemen mit unscharfen beziehungsweise verrauschten Zielfunktionen vergleichsweise gut [149]. 150].

Der in dieser Arbeit vorgestellte Ansatz ist an einer gängigen KE-Methode aus [173] angelehnt und in Algorithmus 1 dargestellt. Der iterative Prozess besteht darin, eine Menge von Datenpunkten  $a_1...a_n$  aus der Verteilung  $\mathfrak{p}$  zu entnehmen, basierend auf ihrer aktuellen Parametrisierung  $\Phi_{g-1}$  (Zeile 3). Die Zielfunktion f des Optimierungsproblems ordnet jedem Datenpunkt  $a_1...a_n$  Fitnesswerte  $v_1...v_n$  zu (Zeile 4). Danach wählt ein Selektionsmechanismus einen Teil  $\rho$  aus den Datenpunkten als Elitestichproben aus (Zeile 5 und 6) und die Berechnung der neuen Parametrisierung von  $\mathfrak{p}$ ,  $\Phi_g$  wird anhand dieser Elitestichproben durchgeführt (Zeile 6). Diese bildet somit die Grundlage für die nächste Iteration.

Algorithmus 1 Kreuzentropie-Methode								
1: function OPTIMIERE( $\mathfrak{p}, \Phi_0, f, \rho, n, G$ )								
2: for $g = 1 \rightarrow G$ do								
3: $a_1a_n \sim \mathfrak{p}(\cdot   \Phi_{g-1}), \mathbf{a} \leftarrow a_1a_n$								
4: $v_1v_n \sim f(a_1)f(a_n), \mathbf{v} \leftarrow v_1v_n$								
5: sortiere $\mathbf{a}$ entsprechend nach $\mathbf{v}$								
6: $\Phi_g \leftarrow \operatorname{argmax}_{\Phi} \prod_{i=1}^{\lfloor n\rho \rfloor} \mathfrak{p}(a_i   \Phi)$								
7: end for								
8: return $a_1$								
9: end function								

Ein zentraler Bestandteil des Algorithmus ist die Verteilung  $\mathfrak{p}$ , die die Wahl der neuen Stichprobenpunkte *a* für jede Generation bestimmt. Je näher die initiale Verteilung  $\mathfrak{p}(\cdot|\Phi_0)$  den optimalen Stichproben kommt, desto schneller konvergiert der Algorithmus. Wenn jedoch keine Informationen über die optimale Verteilung verfügbar sind oder das Ziehen von Stichproben ineffizient ist, ist es im Allgemeinen eine gute Idee eine Verteilung zu wählen, die den gesamten Stichprobenraum abdeckt. Dadurch erhöht sich die Wahrscheinlichkeit, dass sich der Algorithmus bereits in frühen Generationen zu einer guten Lösung konvergiert. Die Parametrisierung  $\Phi_g$  der Verteilung wird entsprechend einer Maximum-Likelihood-Schätzung des gewählten Eliteanteils in der aktuellen Generation aktualisiert. Für eine multivariate Normalverteilung lautet die Gleichung für die Aktualisierung der Parameter [173]:

$$\mu_g = \frac{\sum_{i=1}^{\lceil n\rho \rceil} X_i}{\lceil n\rho \rceil}$$
$$\sigma_g^2 = \frac{\sum_{i=1}^{\lceil n\rho \rceil} (X_i - \mu_g)^T (X_i - \mu_g)}{\lceil n\rho \rceil}$$
$$\Phi_g = \langle \mu_g, \sigma_g^2 \rangle,$$

wobe<br/>i $X_i$ mehrdimensional ist und  $\mu_g$  und<br/>  $\sigma_g^2$  den mehrdimensionalen Mittelwertvektor bzw. die Kovarianz über die Elitestich<br/>proben der aktuellen Ge-

neration darstellen. Neben der Wahl der Stichprobenverteilung, insbesondere des gewählten Anteils  $\rho$  der Elitestichproben, müssen die Populationsgröße nund die Anzahl der Generationen G für ein gegebenes Problem entsprechend angepasst werden, um die Wahrscheinlichkeit, eine gute Lösung zu finden, zu maximieren. In Abschnitt 4.4 wird erläutert, wie diese KE-Methode für die eingeschränkten Hamiltonians unter Verwendung des QAOA, beziehungsweise der QA Heuristik angepasst wird.

## 4.3 Verwandte Arbeiten

Wie bereits in Abschnitt 4.1 erwähnt, existieren viele verschiedene Arbeiten, um Quantenalgorithmen, wie den QAOA oder die Quantum Annealing Heuristik, die standardmäßig zur Lösung von Optimierungsproblemen eingesetzt werden, zu verbessern.

2014 wurde der ursprüngliche QAOA publiziert und es konnte gezeigt werden, dass der hybride Algorithmus im Vergleich zu dem klassischen naiven Algorithmus ein besseres Näherungsverhältnis für das Maximum Cut Problem auf 3regulären Graphen und einer QAOA-Schaltkreistiefe von p = 1 erzielt [60]. Jedoch wurde dies anschließend mit einem verbesserten klassischen Algorithmus widerlegt [11]. Nichtsdestotrotz wurde seitdem der QAOA und dessen Potenziale und Optimierungsmöglichkeiten sowohl experimentell als auch theoretisch umfassend untersucht [41, 184]. Meist sind diese Optimierungen auf Algorithmenebene anzutreffen. Beispielsweise kann der QAOA mit zuvor berechneten Gatterparametern initialisiert werden [77, 151] oder anstelle einer gleichverteilten Superposition aus einem klassisch berechneten lokalen Optimum starten. Dieser Ansatz ist als Warm-Starting QAOA (WS-QAOA) bekannt [56]. Selbst hierfür wurden bereits neue Optimierungsstrategien identifiziert. Beispielsweise verwenden Truger et al. einen klassischen Algorithmus, um die Hyperparameter des WS-QAOA effizient vorzuberechnen, und demonstrieren experimentell dessen Einfluss auf die allgemeine Lösungsqualität [166].

Auch die Quantum Annealing Heuristik wurde neben dem gatterbasierten QAOA über die Jahre stetig weiterentwickelt und optimiert. Pelofkse et al. untersuchen die Möglichkeiten Einfluss auf den D-Wave Annealing-Prozess zu nehmen, um die allgemeine Lösungsqualität zu steigern [130]. Insbesondere analysieren sie *Reverse Annealing*, ein Feature der D-Wave Systemen, um eine bereits gefundene lokale Lösung zu verbessern. Hierfür wird dem System durch den Tansversalfeldkoeffizienten zunehmend kinetische Energie hinzugefügt, um somit im Annealing-Prozess von dem lokalen Optimum (zuvor gefundenen Lösung) rückwärts zu höher gelegene Lösungen zu gelangen. Anschließend wird mit dem konventionellen Annealing und der Abnahme von kinetischer Energie, gesteuert durch das Transversalfeld, eine neue energetisch niedrige und im besten Fall bessere Lösung gefunden. Eine weitere Methode wurde von Yarkoni et al. vorgestellt. Sie verwenden einen evolutionären Algorithmus, um die D-Wave Annealing Offsets zu optimieren und Einfluss auf den Annealing-Prozess zu nehmen [180].

Die in diesem Kapitel vorgestellte Optimierungsmethode unterscheidet sich jedoch von den bisher beschrieben Methoden in dem Sinne, dass sie auf Problemebene, sprich dem Hamiltonian des jeweiligen Optimierungsproblem, ansetzt und somit algorithmusunabhängig ist. In Bezug auf die Hamiltonians existieren vor allem Arbeiten, die sich entweder mit ressourcenschonenden Formulierungen für die Hamiltonians (Anzahl der Variablen und deren Abhängigkeiten) [124, 168] oder mit effizienten Hardware-Embeddings für verschiedene Hardwarearchitekturen, beschäftigen [183].

2019 publizierte Choi eine Arbeit, in der sie theoretisch anhand des *Maximum-Weighted Independent Set Problem* (MIS) zeigte, dass durch die Anpassung der Penalty-Parameter des entsprechenden Hamiltonians die minimale Spektrallücke beeinflusst wird und das Energiespektrum von einem mit Anticrossing zu einem ohne Anticrossing oder umgekehrt genauso geändert werden kann [37].

Ausgehend von diesen Erkenntnissen wird in diesem Kapitel eine modifizierte Kreuzentropie-Methode vorgestellt, mit der die Penalty-Parameter der eingeschränkten Hamiltonians automatisiert optimiert werden, um das jeweilige Energiespektrum und dessen minimale Spektrallücke so anzupassen, dass die Wahrscheinlichkeit den Grundzustand zu messen, erhöht wird und somit eine allgemein verbesserte Lösungsqualität auf NISQ-Hardware erzielt werden kann.

# 4.4 Konzept

Wie in Abschnitt 2.3.3 angegeben, ist die Wahl der Penalty-Parameter für eingeschränkte Hamiltonians durch ihre jeweiligen Nebenbedingungen definiert. Um diese Bedingungen zu erfüllen, wird ein modifiziertes KE-Optimierungsverfahren (siehe Algorithmus 2) verwendet, bei dem die Penalty-Parameter A, B aus einer abgeschnittenen Normalverteilungen  $\mathfrak{p}$  gezogen werden. Da die zulässigen Werte für den Penalty-Parameter A von der Wahl des Penalty-Parameters B abhängen, wird zunächst ein Wert für B (Zeile 3) aus einem passend definierten Intervall  $\Gamma_A$  gezogen. Anschließend wird der Wert für A über ein Intervall  $\Gamma_A(B)$  gezogen, sodass die jeweilige problemspezifische Bedingung erfüllt ist (Zeile 5). Dies wird für n Stichproben durchgeführt. Für jede Probe wird der entsprechende Hamiltonian konstruiert, wie in Abschnitt 2.3.3 beschrieben und der eigentliche Quantenalgorithmus (QAOA/QA) ausgeführt, um ihr einen Fitnesswert  $v_1...v_n$  zuzuweisen, der dem Näherungsverhältnis an die BKS für das jeweilige  $A_i, B_i$ -Paar entspricht (Zeile 7). Dies wird iterativ für die angegebene Anzahl an Generationen G durchgeführt. In Abbildung 4.1 ist das Suchverfahren nach optimalen Penalty-Parametern für einen KP Problem-Hamiltonian graphisch dargestellt.

Grundsätzlich könnte zur Optimierung der Penalty-Parameter der Hamiltonians auch eine andere Optimierungsmethode, wie beispielsweise evolutionäre Algorithmus 2 Kreuzentropie Penalty-Parameter Optimierung in Kombination mit QAOA bzw. QA

1: function OPTIMIERE( $\mathfrak{p}, \Phi_0, f, \rho, n, G$ ) for  $g = 1 \rightarrow G$  do 2:  $B_1...B_n \sim \mathfrak{p}(\cdot | \Phi_{q-1}, \Gamma_B)$ 3:  $\mathbf{B} \leftarrow B_1 \dots B_n$ 4:  $A_1...A_n \sim \mathfrak{p}(\cdot | \Phi_{g-1}, \Gamma_A(\mathbf{B}))$ 5:  $\mathbf{A} \leftarrow A_1 \dots A_n$ 6:  $v_1...v_n \sim QAOA/QA(A_1, B_1)...QAOA/QA(A_n, B_n)$ 7:8:  $\mathbf{v} \leftarrow v_1 \dots v_n$ sortiere  $\mathbf{A}$ ,  $\mathbf{B}$  entsprechend nach  $\mathbf{v}$ 9:  $\Phi_g \leftarrow \operatorname{argmax}_{\Phi} \prod_{i=1}^{\lceil \hat{n} \rho \rceil} \mathfrak{p}(A_i, B_i | \Phi)$ 10: end for 11: 12: return  $A_1, B_1$ 13: end function



Abbildung 4.1: Beispielhafte Darstellung der KE-Methode für zehn Generationen (G = 10). Die graue Schattierung repräsentiert die ungültigen Bereiche der abgeschnittenen Normalverteilung. Die Ellipsen stellen den Erwartungswert  $\mu_g$  und die Kovarianz  $\sigma_g^2$ der Generation g dar. Die gefüllten Kreise entsprechen dem besten  $\rho$ -Eliteanteil der Individuen. Die besten Werte, die die KE-Methode für die Probleminstanz  $\mathcal{A}$  des KP (vgl. Tabelle 4.1) gefunden hat, waren A = 2.7 und B = 1.1. Die Fitness der jeweiligen Individuen wurde in dieser Darstellung mit dem QAOA auf dem IBMQ QASM Simulator berechnet. oder gradientenbasierte Optimierung verwendet werden. Die Wahl für die Kreuzentropie-Optimierung basiert auf verschiedenen positiven Eigenschaften der Methode. Zum einen ist die KE-Methode eine globale Optimierungsmethode und vermeidet daher mit einer größeren Wahrscheinlichkeit die Konvergenz zu einem lokalen Optimum. Zum anderen benötigen lokale Suchverfahren oftmals Domänenwissen und geeignete Preprocessing-Methoden, um die Suche nahe des globalen Optimums initialisieren zu können. Eine weitere Eigenschaft der vorgestellten KE-Methode ist deren lineare Laufzeit mit der Anzahl der Stichproben n und die Möglichkeit, die Methode grundsätzlich innerhalb einer Generation parallelisieren zu können, was bedeutet, dass die Optimierung selbst vergleichsweise weniger rechenintensiv ist [173].

# 4.5 Optimierung der Penalty-Parameter eingeschränkter Hamiltonians für gatterbasierte Architekturen

Hybride quanten-klassische Algorithmen gelten als eine der vielversprechendsten Ansätze, um die Vorteile von gatterbasierten Quantencomputern in der NISQ-Ära für praktische Anwendungen zu nutzen. Solche Algorithmen werden in der Regel in Form eines variationellen Quantenschaltkreis implementiert, der einen klassischen Optimierer mit dem Quantencomputer kombiniert. Die Lösungsqualität hängt dabei im Wesentlichen von den Parametern ab, die der klassische Optimierer bei jeder Iteration für die Gatter des Schaltkreises wählt. Die Lösungslandschaft dieser Gatterparameter ist jedoch meist hochdimensional und enthält viele lokale Optima.

Im nachfolgenden Abschnitt wird die Kreuzentropie-Methode in Kombination mit QAOA anhand des KP evaluiert. Es wird untersucht, welchen Einfluss die Optimierung der Penalty-Parameter des eingeschränkten KP-Hamiltonian auf die Lösungslandschaft der Gatterparameter des QAOA hat.

## 4.5.1 Aufbau der Experimente

Für die Experimente wurde der IBMQ QASM Simulator [5] verwendet, um die Simulationen des QAOA-Schaltkreis durchzuführen. Das IBM Qiskit SDK enthält verschiedene Optimierungsalgorithmen aus der *NLopt-Bibliothek* [96], die in Kombination mit dem QAOA verwendet werden können. Die Experimente zeigten eine ähnliche Lösungsqualität für unterschiedliche Optimierer für die verwendeten Probleminstanzen, sodass letztendlich die Wahl auf einen modifizierten Evolutionären Algorithmus für globale Optimierung (ESCH) fiel [159].

Um den QAOA-KE Ansatz zu testen, wurden fünf Probleminstanzen des KP verwendet, dessen Eigenschaften in Tabelle 4.1 dargestellt sind. Da die Lösungsqualität des QAOA auf IBM Hardware als auch auf Simulatoren für große

Probleminstanz	Gegenstand	Gewichte	Wertigkeiten	W	BKS
$\mathcal{A}$	2	1,1	2,1	1	101
B	2	$1,\!1$	$1,\!2$	1	011
С	2	$1,\!1$	$2,\!1$	2	1101
$\mathcal{D}$	2	$2,\!3$	2,1	2	1001
E	2	1,2	$2,\!1$	2	1010

Tabelle 4.1: Verwendete KP-Probleminstanzen. Die Problemstellung und Eigenschaften des KPs sind aus der entsprechenden Definition in Kapitel 2.3.3.5 zu entnehmen. Die *Best Known Solution* (BKS) entspricht hier dem klassischen Binärvektor  $x \in \{0, 1\}^n$  nach der Messung des Quantensystems.

Probleminstanzen zum aktuellen Zeitpunkt relativ gering ist (vgl. Abschnitt 3.4.3), wurden eher kleine Instanzen verwendet, um die Wirkung beziehungsweise den Einfluss der KE-Methode zu demonstrieren. Der KE-Ansatz ist theoretisch aber auch auf größere Instanzen anwendbar.

In Tabelle 4.2 sind die Parametereinstellungen der KE-Methode aufgeführt. Da die hier vorgestellte angepasste Methode eine abgeschnittene Normalverteilung zur Stichprobenziehung verwendet, müssen zusätzliche Clipping-Parameter für die beiden Penalty-Parameter A und B angegeben werden. Das Intervall von A wird anhand von  $\Gamma_A(B) = [B \cdot max(c_{\alpha}) + 0.1, B \cdot max(c_{\alpha}) + 10.0]$  definiert.

G	n	ρ	$\gamma$	min $\sigma^2$	$\sigma_0^2$	$\mu_0$	Intervall von B
10	100	0.1	0.5	0.1	1.0	0.0	[0.1, 10.0]

Tabelle 4.2: Verwendete Parametereinstellungen der Kreuzentropie-Methode. Die Lernrate  $\gamma$  gibt die Änderungsrate der Parameter von  $\Phi_{g-1}$ zu  $\Phi_g$  an.

#### 4.5.2 Ergebnisse und Diskussion

Die Experimente zeigen, dass das Ändern der Penalty-Parameter des jeweiligen Problem-Hamiltonians dessen Energie- beziehungsweise Lösungslandschaft des entsprechenden Optimierungsproblems beeinflusst und damit auch den Pfad des klassischen Optimierers. Ein Beispiel ist in Abbildung 4.2 zu sehen. In Abbildung 4.2a sieht man zufällig gewählte Penalty-Parameter, während in Abbildung 4.2b die mittels Kreuzentropie optimierten Parameter verwendet wurden. Die Histogramme in Abbildung 4.2c und 4.2d zeigen die entsprechenden Lösungsqualitäten in Bezug auf die *Best Known Solution* (BKS). Als Metrik für die Lösungsqualität wurde das Näherungsverhältnis verwendet, das wie folgt definiert ist:

$$N\ddot{a}herungsverh\ddot{a}ltnis = \frac{\#BKS}{\#Messungen}$$
(4.1)

wobei #BKS die Anzahl der Messungen bei denen die BKS gemessen wurde und #Messungen die Gesamtzahl der Messungen beschreibt. Hierfür wurden die voreingestellten 1024 Messungen verwendet. In Zusammenhang mit den hier verwendeten Probleminstanzen ist die BKS auch der Grundzustand des jeweiligen Hamiltonians.

In Abbildung 4.2b ist der klassische Optimierer (grüne Kreise) in der Lage, in der Nähe des globalen Minimums (rotes Kreuz) zu suchen, was sich in einer verbesserten Lösungsqualität in der entsprechenden Abbildung 4.2d widerspiegelt. Die Optimierung der Penalty-Parameter des Problem-Hamiltonians begünstigt den klassischen Optimierer bessere Gatterparameter für den Quantenschaltkreis des QAOA zu finden und folglich eine bessere Lösungsqualität nach erfolgter Messung zu erreichen.

Um diese Beobachtung zu festigen, wurden in Abbildung 4.3 die Ergebnisse des QAOA mit Kreuzentropie und dem konventionellen QAOA miteinander verglichen. Beide Ansätze wurden an verschiedenen Probleminstanzen, wie in Tabelle 4.1 dargestellt, getestet. Abbildung 4.3a-4.3c unterscheiden sich in der QAOA-Schaltkreistiefe p = 1, 2 und 3. Die Kreuzentropie-Methode wurde mit der Parametereinstellung von Tabelle 4.2 initialisiert und der klassische QAOA-Optimierer ESCH wurde auf 200 Iterationen eingestellt. Für den konventionellen QAOA-Ansatz wurden fünf Penalty-Parameterpaare aus dem gleichen Intervall, wie in Tabelle 4.2 angegeben, gezogen. Jedes Penalty-Parameterpaar wurde 10-mal auf dem IBMQ QASM Simulator ausgeführt und die entsprechenden Lösungsqualitäten (Näherungsverhältnis) sind in den jeweiligen mit *zufällig* gekennzeichneten Boxplots dargestellt. Für die QAOA-KE Boxplots, die als *optimiert* gekennzeichnet sind, wurde der  $\rho$ -Anteil der Penalty-Parameterpopulation der letzten Kreuzentropie-Generation verwendet und deren Lösungsqualität in Form des Näherungsverhältnis berechnet.

Die Ergebnisse zeigen, dass für den QAOA-Schaltkreis mit einer Tiefe von p = 1 der Mittelwert der Lösungsqualität durch die Verwendung der optimierten Penalty-Parameter in jeder Probleminstanz um mehr als 100% gesteigert

4.5 Optimierung der Penalty-Parameter eingeschränkter Hamiltonians für QAOA



(c) Lösungsqualität (A=3.2, B=0.2)

(d) Lösungsqualität (A=2.7,B=1.1)

Abbildung 4.2: Lösungslandschaften und Lösungsqualitäten der QAOA-Zielfunktion  $\min \langle \psi(\beta, \gamma) | H_C | \psi(\beta, \gamma) \rangle$  für die Probleminstanz  $\mathcal{A}$  mit  $H_C$  als deren Kosten- bzw. Problem-Hamiltonian. Die Probleminstanz  $\mathcal{A}$  besteht aus zwei Gegenständen  $x_1$  und  $x_2$ mit Gewichten  $w_1 = 1$  bzw.  $w_2 = 1$ . Die Rucksackkapazität  $y_1$ beträgt W = 1. Die Wertigkeit der Gegenstände wurden auf  $c_1 = 2$  und  $c_2 = 1$  gesetzt. Die BKS für die Variablen  $|x_1, x_2, y_1\rangle$ ist demnach 101 und in dunkelblau markiert. Die grünen Kreise entsprechen den Parametern, die durch den klassischen Optimierer abgetastet wurden. Das rote Kreuz markiert das globale Minimum der Lösungs- bzw. Energielandschaft, das für die Abbildungen 4.2a und 4.2b bei -2.85 bzw. -3.2 liegt. In Abbildung 4.2a sieht man vergleichsweise schlechte und zufällig gewählte Penalty-Parameter, während in Abbildung 4.2b, die mittels der KE-Methode optimierten Parameter verwendet wurden. Die Histogramme in Abbildung 4.2c und 4.2d zeigen die entsprechenden Lösungsqualitäten in Form des Näherungsverhältnisses.



Abbildung 4.3: Darstellung der Lösungsqualitäten für verschiedene QAOA-Schaltkreistiefen p = 1, 2, 3 (4.3a-4.3c). Für jede Schaltkreistiefe wurden fünf KP-Instanzen  $\mathcal{A}$ - $\mathcal{E}$  verwendet. Die mit zufällig gekennzeichneten Boxplots repräsentieren das Näherungsverhältnis von fünf zufällig gezogenen Penalty-Parameterpaaren (mit jeweils zehn Ausführungen), während die mit optimiert gekennzeichneten Boxplots das Näherungsverhältnis des  $\rho$ -Anteils der Penalty-Parameterpopulation der 10. Generation des KE-Verfahrens darstellen. werden konnte (siehe Abbildung 4.3a). Auch für p = 2 konnten die Lösungsqualitäten um etwa 100% gesteigert werden (vgl. Abbildung 4.3b). In Abbildung 4.3c ist für p = 3 ebenfalls ein deutlicher Zugewinn an Lösungsqualität zu erkennen. Da die Suchraumgröße der Gatterparameter polynomiell mit der Schaltkreistiefe p zunimmt, ist es grundsätzlich für den klassischen Optimierer schwieriger die optimalen Parameter zu finden. Das spiegelt sich auch hier in der Lösungsqualität wider, die im Vergleich zu den geringeren Schaltkreistiefen weniger signifikant ist.

Ein weiterer Aspekt, der in Abbildung 4.3 über alle Probleminstanzen hinweg zu sehen ist, ist die vergleichsweise geringe Varianz des Näherungsverhältnisses der optimierten Penalty-Parameter. Der Grund hierfür ist, dass diese Werte aus dem besten  $\rho$ -Anteil der Individuen der letzten KE-Generation ausgewählt wurden, während die fünf zufällig ausgewählten Penalty-Parameter unvorteilhafte Paare enthalten können. Dies ändert jedoch nichts an der Tatsache, dass die KE-Methode verhältnismäßig stabil gegenüber dem stochastischen klassischen Optimierer ist.

Zusätzlich ist zu erwähnen, dass bei der Verwendung der zufällig gezogenen Penalty-Parameter auch die Möglichkeit besteht, die optimierten Penalty-Parameter zu übertreffen (siehe z.B. das maximale Näherungsverhältnis in Abbildung 4.3c für die Probleminstanz  $\mathcal{B}$  oder  $\mathcal{E}$ ). Dies ist vermutlich auf das stochastische Quantensystem und den klassischen Optimierer zurückzuführen, der gegebenenfalls auch mal bei einem Ausreißer gute Gatterparameter finden und somit eine vergleichsweise gute oder sogar bessere Lösungsqualität erzielen kann.

Da sich die mit der KE-Methode gefundenen Penalty-Parameter in den verwendeten Probleminstanzen unterscheiden, ist es schwer, eine allgemeingültige Aussage über inhärente Korrelationen zwischen den Parametern unterschiedlicher Probleminstanzen zu treffen. Daher müssen die Penalty-Parameter für jede Probleminstanz individuell optimiert werden. Infolgedessen liegt die Stärke dieses Ansatzes in der Verbesserung der Lösungsqualität und weniger in der Minimierung der Laufzeit.

#### 4.5.3 Zusammenfassung und Ausblick

In diesem Abschnitt wurde eine Kreuzentropie-Methode zur Optimierung der Penalty-Parameter des KP Ising-Hamiltonian vorgestellt. Die Optimierung ermöglicht es dem klassischen QAOA Optimierer, bessere Gatterparameter für den Quantenschaltkreis zu finden, sodass eine signifikante Steigerung der Lösungsqualität im Vergleich zum konventionellen Ansatz, bei dem die Penalty-Parameter zufällig durch den Entwickler gewählt werden, erreicht werden kann. Es wurde gezeigt, dass für die QAOA-Schaltkreistiefen p = 1 und p = 2 eine Verbesserung der Lösungsqualität von mehr als 100% erreicht werden kann. Außerdem führt die Wahl der optimierten Penalty-Parameter der letzten KE-Generation zu einer geringeren Varianz in der Lösungsqualität, was wiederum bedeutet, dass weniger Messungen gemittelt werden müssen, um gute Lösungen zu finden.

Der Nachteil der Kreuzentropie-Methode liegt allerdings in dem Trade-off zwischen Laufzeit und Lösungsqualität. Die Experimente wiesen keine Regelmäßigkeiten beziehungsweise Korrelationen in den optimierten Penalty-Parametern auf, sodass für jede Probleminstanz die Kreuzentropie-Methode angewendet werden musste. Jedoch soll angemerkt werden, dass die Absicht bei der Verwendung dieser Optimierung genau diese verbesserte Lösungsqualität war. Auch Farhi et al. zielten mit ihrem ursprünglichen konventionellen QAOA eher darauf ab, eine gute Lösungsqualität für ein gewähltes Optimierungsproblem zu erreichen, da der klassische Optimierer ohnehin den Engpass in Bezug auf die QAOA Laufzeit darstellt [60].

In Bezug auf zukünftige Arbeiten wäre es interessant zu sehen, ob bestimmte Mustererkennungsalgorithmen in der Lage sind, mögliche Korrelationen zwischen den optimierten Penalty-Parametern zu finden. Außerdem sollte der KE-Ansatz in Zukunft auch für größere Probleminstanzen untersucht werden, wenn die Quantum Computing Hardware skaliert und in der Lage ist, eine ausreichende Lösungsqualität mit verbesserten Rauschverhältnis zu gewährleisten.

# 4.6 Optimierung der Penalty-Parameter eingeschränkter Hamiltonians für annealingbasierte Architekturen

Das Konzept der Anwendung der Kreuzentropie-Methode auf die Penalty-Parameter der Problem-Hamiltonians kann auch analog mit annealingbasierter Hardware, wie beispielsweise den D-Wave Quantum Annealern, kombiniert werden. Wie bereits in Abschnitt 4.5 gezeigt wurde, hat die Wahl der Penalty-Parameter einen Einfluss auf die zugrundeliegende Energielandschaft des Hamiltonians. In Bezug auf Quantum Annealing und dem adiabatischen Quantum Computing ist die Größe der minimalen Spektrallücke des Hamiltonian-Eigenspektrums ein ausschlaggebendes Maß für dessen Anwendbarkeit bzw. Lösbarkeit auf entsprechender Hardware.

Im nachfolgenden Abschnitt wird die Kreuzentropie-Methode in Kombination mit der D-Wave Quantum Annealing Hardware anhand des KP, MECP und SPP evaluiert. Es wird untersucht, welchen Einfluss die Optimierung der Penalty-Parameter des eingeschränkten Problem-Hamiltonians auf dessen Eigenspektrum hat.

## 4.6.1 Aufbau der Experimente

Für die Bewertung der Kreuzentropie-Methode zur Optimierung der Penalty-Parameter der Hamiltonian in Kombination mit Quantum Annealing wurde die



Abbildung 4.4: Verschiedene D-Wave Hardware-Emebeddings und ihre Lösungsqualität in Form des Näherungsverältnisses zu der BKS für die MECP-Instanz  $\mathcal{A}$ .

D-Wave 2000Q Hardware verwendet. Es wurde die voreingestellte Annealing-Zeit von  $20\mu s$  und die standardmäßigen Energiefunktionen A(t) und B(t), wie in [45] angegeben, verwendet, um die Eigenwerte bzw. -energien eines Hamiltonians zu einem bestimmten Zeitpunkt im Annealing-Prozess zu berechnen. Für das Embbeding der Probleminstanzen auf die D-Wave Hardware wurde wiederum die D-Wave Heuristik *Minorminer* verwendet [30]. Da aufgrund der spärlichen Konnektivität der Hardware-Qubits nicht jeder logische Problemgraph direkt auf die Architektur abbildbar ist, müssen physikalische Ancillae verwendet werden, die dann ein logisches Qubit repräsentieren. Zudem ist bekannt, dass nicht jedes Qubit der D-Wave QPU die gleiche physikalische Qualität hat und ein unterschiedliches Rauschverhalten aufweisen kann. Das bedeutet, dass dieselbe Graphstruktur, die auf verschiedenen physikalischen Qubits der QPUs abgebildet wird, zu unterschiedlichen Lösungsqualitäten führen kann [43]. In Abbildung 4.4 ist der logische Problemgraph der MECP-Instanz  $\mathcal{A}$  zehnmal auf unterschiedlichen Qubits der D-Wave 2000Q QPU abgebildet worden. Für jedes Embedding wurden die gleichen 100 Penalty-Parameterpaare verwendet und deren Fitness über drei Durchläufe pro Paar gemittelt. Somit repräsentiert jeder Boxplot 300 Messwerte. Es ist zu sehen, dass bei dem besten Embedding (Nr. 3) die mittlere Lösungsqualität in Bezug auf die BKS etwa 71% beträgt, während bei dem schlechtesten Embedding (Nr. 2) die mittlere Lösungsqualität etwa 50% beträgt. Folglich ist zu erkennen, dass die Wahl des Hardware-Embeddings die absolute Lösungsqualität beeinflussen kann. Um eine bessere Vergleichbarkeit zu gewährleisten, wurde für alle Probleminstanzen die gleiche Qubit-Auswahl der QPU verwendet.

In Tabelle 4.3 sind die Parametereinstellungen der KE-Methode in Kombination mit der D-Wave Quantum Annealing Heuristik aufgeführt. Da auch hier wiederum für die Stichprobenziehung eine abgeschnittene Normalverteilung verwendet werden muss, wurden zusätzliche Clipping-Parameter für die beiden Penalty-Parameter A und B angegeben. Das Intervall von A wird gemäß den Nebenbedingungen des jeweiligen Problems berechnet.

Für die drei untersuchten Problemklassen wurden jeweils vier Probleminstan-

G	n	ρ	$\gamma^*$	min $\sigma^2$	$\sigma_0^2$	$\mu_0$	Intervall von B
10	100	0.1	0.5	0.1	1.0	0.0	[0.1, 10.0]

Tabelle 4.3: Verwendete Parametereinstellungen der Kreuzentropie-Methode in Kombination mit Quantum Annealing

zen generiert. Diese wurden entsprechend mit den Bezeichnungen  $\mathcal{A}, \mathcal{B}, \mathcal{C}$  und  $\mathcal{D}$  betitelt und umfassen 4 bis 7 logische Qubits. Da D-Wave Quantum Annealing Hardware zum aktuellen Zeitpunkt noch immer sehr begrenzt in der Anzahl der Qubits und deren Konnektivität ist, mussten für diese logischen Probleme bereits physikalische Ancillae eingeführt werden, um ein gültiges Hardware-Embedding zu ermöglichen. Diese zusätzlichen Qubits sind durch den D-Wave Parameter *Chain Strength* gekoppelt, um sicherzustellen, dass sie bei der Messung auf denselben Basiszustand kollabieren. Dieser zusätzliche Parameter beeinflusst jedoch auch die Gesamtlösungsqualität und macht es für die Kreuzentropie-Methode schwieriger, große Problemfälle zu optimieren. Deshalb wurden vergleichsweise eher kleine Instanzen verwendet, um die Wirkung der Optimierungsmethode zu demonstrieren. Der Ansatz ist jedoch theoretisch auch auf größere Probleme anwendbar.

#### 4.6.2 Ergebnisse und Diskussion

Die Experimente zeigen, dass auch hier durch die Variation der Penalty-Parameter die Energielandschaft des entsprechenden eingeschränkten Hamiltonians erheblich verändert werden kann, sodass auch das Tunneln durch Energiebarrieren für die Quantum Annealing Heuristik beeinflusst wird. Ein Beispiel ist in Abbildung 4.5 zu sehen. Hier ist das Eigenspektrum eines MECP-Hamiltonian mit zufällig gewählten Penalty-Parametern über die Annealing-Zeit dargestellt (vgl. Abbildung 4.5a). Das gleiche Setup, aber mittels Kreuzentropie optimierten Parametern, wird in Abbildung 4.5b gezeigt. Die Histogramme in Abbildung 4.5c und 4.5d repräsentieren die entsprechenden Lösungsqualitäten in Form des bereits definierten Näherungsverhältnisses.

In Abbildung 4.5a beträgt die minimale spektrale Lücke  $g_{min} \approx 1 \ h \times GHz$ und befindet sich in der Mitte des Annealing-Prozesses, während in Abbildung 4.5b die Lücke  $g_{min} \approx 3 \ h \times GHz$  beträgt und an den Anfang des Annealing-Prozesses verschoben wurde. Die Auswirkung der Skalierung und Verschiebung von  $g_{min}$  spiegelt sich auch in einer verbesserten Lösungsqualität der Quantum Annealing Heuristik wider (vgl. Abbildung 4.5d). Durch Optimierung der Penalty-Parameter des Problem-Hamiltonians, in solch einer Weise, dass  $g_{min}$ sich vergrößert, erhöht die Wahrscheinlichkeit öfter im Grundzustand zu verbleiben, was letztendlich beim Quantum Annealing zu einer besseren Lösungs-



qualität führt.

(c) Lösungsqualität (A=0.536, B=0.107) (d) Lösungsqualität (A=10.38, B=2.25)

Abbildung 4.5: Eigenspektren und Lösungsqualitäten für die MECP-Probleminstanz  $\mathcal{A}$ . Das Problem  $\mathcal{A}$  besteht aus vier Teilmengen mit Zahlen von 1 bis 4, [(1, 2), (1, 3), (1, 2, 4), (3)]. Die BKS (dunkelblau markiert) ist 0011, d.h., die Mengen mit Index 2 und 3 sind in der Lösung enthalten.  $g_{min} = \delta$  markiert die minimale Spektrallücke, die für zufällige Penalty-Parameter A und B etwa 1  $h \times GHz$  und für optimierte Werte etwa 3  $h \times GHz$  beträgt, siehe Abbildungen 4.5a bzw. 4.5b. Die Abbildungen 4.5c und 4.5d zeigen die Wahrscheinlichkeiten, einen entsprechenden Zustand zu messen.

In Abbildung 4.6 ist der lineare Zusammenhang zwischen dem Näherungsverhältnis zur BKS und der minimalen Spektrallücke beschrieben. Die rote Zahl stellt den Pearson-Korrelationskoeffizienten dar, der Werte im Bereich von -1bis +1 annehmen kann. Eine starke negative oder positive Korrelation spiegelt sich in den Extrema (±1) des Koeffizienten wieder, während ein Wert von 0 ein unkorreliertes Verhalten zwischen zwei Variablen beschreibt. Die erste Reihe repräsentiert die KP-Instanzen  $\mathcal{A} - \mathcal{D}$ , während die zweite und dritte Reihe die MECP- bzw. SPP-Instanzen darstellen. Die blauen Kreise repräsentieren die Individuen (Penalty-Parameterpaare) über zehn Generationen (insgesamt 1000 Individuen) des KE-Verfahrens. Die Lösungsqualität ist in Form des Näherungsverhältnisses zur BKS angegeben. Man beachte, dass das Näherungs-



Abbildung 4.6: Korrelation des Näherungsverhältnisses zur BKS und der minimalen spektralen Lücke. Die blauen Kreise repräsentieren die Individuen (Penalty-Parameterpaare) über zehn Generationen der KE-Methode (insgesamt 1000 Individuen pro Abbildung). Die erste Reihe repräsentiert die KP-Instanzen  $\mathcal{A} - \mathcal{D}$ , während die zweite und dritte Reihe die MECP- bzw. SPP-Instanzen darstellen. Die rote Zahl gibt den jeweiligen Pearson-Korrelationskoeffizienten an.

verhältnis für jedes Individuum der Population über drei Durchläufe auf dem D-Wave 2000Q Quantum Annealer gemittelt wurde, um die Stochastizität des Quantensystems zu kompensieren.

Die Ergebnisse zeigen, dass für jede Probleminstanz (mit Ausnahme von MECP C) eine Korrelation zwischen dem mittels der D-Wave 2000Q Hardware berechneten Näherungsverhältnis und der minimalen Spektrallücke besteht. Es ist zu erkennen, dass das Näherungsverhältnis in den Instanzen, bei dem der Wertebereich der minimalen Spektrallücke vergleichsweise groß ist, ebenfalls größer ist und in einigen Instanzen nahe bei 1.0 liegt (siehe MECP-Instanz  $\mathcal{A}$  und  $\mathcal{B}$ ), während in den Instanzen, in denen der Wertebereich der minimalen Spektrallücke klein ist, auch das Näherungsverhältnis vergleichsweise kleiner ist und eine größere Varianz aufweist (siehe MECP-Instanz  $\mathcal{C}$ ).

In Abbildung 4.7 werden die Ergebnisse von Quantum Annealing in Kombination mit der Kreuzentropie-Methode mit dem konventionellen Ansatz verglichen. Beide Methoden wurden an den Probleminstanzen, die in Abschnitt 4.6.1 beschrieben wurden, getestet. Die Ergebnisse sind in den Abbildungen 4.7a-4.7c für das KP, MECP bzw. SPP dargestellt.

Die KE-Methode wurde mit der Parametereinstellung aus Tabelle 4.3 initialisiert und die D-Wave 2000Q Hardware wurde, wie in Abschnitt 4.6.1 erläutert, verwendet. Beim konventionellen QA-Ansatz wurden zufällig fünf Penalty-Parameterpaare aus demselben Stichprobenintervall, wie in Tabelle 4.3 angegeben, gezogen. Jedes Penalty-Parameterpaar wurde zehnmal auf der Hardware ausgeführt und die entsprechenden Lösungsqualitäten (Näherungsverhältnis zur BKS) sind in dem jeweiligen Boxplot, der als *zufällig* gekennzeichnet ist, dargestellt. Bei den mit *optimiert* gekennzeichneten Boxplots wurde der  $\rho$ -Anteil der Parameterpopulation der letzten KE-Generation verwendet und deren Fitness, d.h. Lösungsqualität, auf der Hardware berechnet.

Die Ergebnisse zeigen, dass für das KP in Abbildung 4.7a die Lösungsqualität, bezogen auf den Mittelwert, durch die Verwendung der optimierten Penalty-Parameter im besten Fall (siehe Problem  $\mathcal{B}$ ) um etwa 500% und im schlechtesten Fall (siehe Problem  $\mathcal{D}$ ) um etwa 85% gesteigert werden konnte.

Auch für das MECP konnten die Lösungsqualitäten im besten Fall um etwa 170% (Problem C) und im schlechtesten Fall um 30% (Problem A) gesteigert werden (vgl. Abbildung 4.7b). Weiterhin ist zu beobachten, dass bei den kleinen Probleminstanzen A und B mit den optimierten Penalty-Parametern ein Näherungsverhältnis von nahezu 1.0 erreicht werden konnte.

In Abbildung 4.7c ist auch für das SPP-Problem eine signifikante Steigerung in der Lösungsqualität zu sehen. Im besten Fall (Problem  $\mathcal{B}$  und  $\mathcal{D}$ ) konnte eine Steigerung von rund 600% erreicht werden, während im schlechtesten Fall (Problem  $\mathcal{A}$ ) die optimierten Penalty-Parameter noch zu einer Steigerung von 80% in der Lösungsqualität führten.

Im Allgemeinen ist zu beobachten, dass die Gesamtlösungsqualität mit der Größe der Probleminstanzen abnimmt. Dies ist jedoch offensichtlich, da die Anzahl der möglichen Lösungen und damit der gesamte Lösungsraum zunimmt.



(c) SPP

Abbildung 4.7: Die Lösungsqualitäten für die vier Probleminstanzen  $\mathcal{A}-\mathcal{D}$  des KP, MECP und SPP sind jeweils in 4.7a-4.7c dargestellt. Die als *zufällig* gekennzeichneten Boxplots repräsentieren das Näherungsverhältnis von fünf zufällig ausgewählten Penalty-Parametern (mit jeweils zehn Ausführungen), während die mit *optimiert* gekennzeichneten Boxplots das Näherungsverhältnis des  $\rho$ -Anteils der Penalty-Parameterpopulation der 10. Generation des KE-Verfahrens darstellen.
Ein weiteres Merkmal ist die vergleichsweise geringe Varianz des Näherungsverhältnisses der optimierten Penalty-Parameter über alle Probleminstanzen hinweg (vgl. Abbildung 4.7). Das liegt daran, dass die Werte aus dem besten  $\rho$ -Anteil der Individuen der letzten KE-Generation ausgewählt wurden, während die zufällig ausgewählten Parameter auch ungünstige enthalten können. Dies ändert jedoch nichts an der Tatsache, dass der KE-Ansatz in Bezug auf die stochastische D-Wave Quantum Annealing Hardware vergleichsweise stabil ist.

#### 4.6.3 Zusammenfassung und Ausblick

In diesem Abschnitt wurde die Kreuzentropie-Methode zur Optimierung der Penalty-Parameter eingeschränkter Hamiltonians unter Anwendung der D-Wave Quantum Annealing Heuristik evaluiert. Anhand der numerischen Berechnung des Eigenspektrums wurde gezeigt, dass die Optimierung zu einer Skalierung der minimalen Spektrallücke führt und es somit der D-Wave Quantum Annealing Heuristik erleichtert, während des Annealing Prozesses, im Grundzustand zu bleiben bzw. zu finden. Dies führt zu einer verbesserten Lösungsqualität (Näherungsverhältnis). Die Experimente wurden anhand der eingeschränkten Hamiltonians des KP, MECP und SPP durchgeführt. Für alle drei Problemklassen konnte eine deutlich verbesserte Lösungsqualität im Vergleich zum konventionellen Ansatz, bei dem die Parameter zufällig gewählt werden, erzielt werden. Darüber hinaus führt die Verwendung der optimierten Penalty-Parameter der letzten Generation der KE-Methode zu einer geringeren Varianz der Lösungsqualität, was bedeutet, dass weniger Messungen gemittelt werden müssen, um gute Lösungen zu finden.

Da sich, wie auch hier, die optimierten Penalty-Parameter in den verwendeten Probleminstanzen unterscheiden, wäre es interessant zu wissen, ob Methoden des maschinellen Lernens Korrelationen und Muster zwischen den optimierten Penalty-Parameter finden können und diese somit für andere Probleminstanzen wiederverwendbar zu machen.

# 4.7 Kapitelzusammenfassung

In diesem Kapitel wurde eine algorithmusunabhängige Methode zur Optimierung eingeschränkter Hamiltonians vorgestellt. Dieser Ansatz optimiert die Penalty-Parameter der Hamiltonians mittels der Kreuzentropie und wird daher auf Problemebene angewendet. Die Experimente zeigten, dass sowohl auf gatter-, als auch annealingbasierten Architekturen eine signifikante Steigerung in der Lösungsqualität im Vergleich zum konventionellen Ansatz, bei dem die Penalty-Parameter zufällig aus einem vorgegebenen Intervall gezogen werden, erzielt werden konnte. Zudem wurde numerisch festgestellt, dass die Optimierung der Penalty-Parameter das Eigenspektrum des Hamiltonians verändert und zu einer Skalierung der minimalen Spektrallücke führt, die als Maß für die Anwendbarkeit auf adiabatischen Quantencomputern und die davon inspirierte D-Wave Quantum Annealing Hardware gilt.

Obwohl der Ansatz in den durchgeführten Experimenten nur an Hamiltonians mit jeweils zwei Penalty-Parametern evaluiert wurde, ist er grundsätzlich nach dem gleichen Konzept aus Abschnitt 4.4 auf mehrere Parameter erweiterbar. Jedoch müsste mit jeder zusätzlichen Dimension die Anzahl an Generationen und die Populationsgröße angepasst werden, um die Wahrscheinlichkeit zur Parameterkonvergenz zu erhöhen. Das beschreibt gleichzeitig den größten Nachteil des vorgestellten Ansatzes, der in dem Trade-off zwischen Laufzeit und Lösungsqualität liegt. Die Experimente zeigten keine Regelmäßigkeiten in den optimierten Penalty-Parametern auf, sodass für jede Probleminstanz die zeitaufwendige Kreuzentropie-Methode angewendet werden muss.

# 5 Einfluss der Penalty-Parameter eingeschränkter Hamiltonians auf das Eigenspektrum

In diesem Kapitel wird der Einfluss der Penalty-Parameter eingeschränkter Hamiltonians auf das jeweilige Eigenspektrum mittels Methoden des *maschinellen Lernens* (engl. Machine Learning (ML)) untersucht, um Regelmäßigkeiten und Muster für die Parameterwahl zu identifizieren. Motiviert ist das Vorhaben aus den Erkenntnissen aus Kapitel 4, die keine Zusammenhänge zwischen den optimierten Penalty-Parametern zwischen den verschiedenen Probleminstanzen aber auch problemklassenübergreifend aufzeigten. Aufgrund dessen musste die Kreuzentropie-Methode für jede Instanz eigens angewendet werden.

Neben der Motivation und Zielsetzung in Abschnitt 5.1 dieses Kapitels wird zunächst in Abschnitt 5.2 die Definitionen und Grundlagen der verwendeten ML-Methoden eingeführt, bevor in Abschnitt 5.3 verwandte Arbeiten in Bezug auf die Optimierung von Penalty-Parameter eingeschränkter Hamiltonians und der generelle Einsatz von ML-Methoden für Quantenalgorithmen vorgestellt werden. Anschließend wird in Abschnitt 5.4 der Aufbau der Experimente und das zugrundeliegende Konzept zur Verwendung der ML-Methoden zur Erkennung von Korrelationen zwischen Penalty-Parametern und der minimalen Spektrallücke des jeweiligen Eigenspektrums beschrieben.

In Abschnitt 5.5 wird dann die Anwendung der Methoden auf einen eigens für die Arbeit generierten Datensatz, der verschiedene Instanzen eingeschränkter Hamiltonians und Penalty-Parametereinstellungen mit der entsprechenden minimalen Spektrallücke beinhaltet, evaluiert. Es wird empirisch gezeigt, dass durch die Anwendung einer Clustering-Methode verschiedene Muster, beziehungsweise Korrelationen zwischen Penalty-Parametereinstellungen und der minimalen Spektrallücke innerhalb von Problemklassen bestehen. Zudem wird anhand eines Regressionsmodells demonstriert, dass es möglich ist anhand des generierten Datensatzes optimale Penalty-Parametereinstellungen vorherzusagen.

Die Inhalte dieses Kapitels basieren auf [144]. Eine Zusammenfassung des vorgestellten Ansatzes, sowie mögliche Ansatzpunkte für zukünftige Arbeiten befinden sich am Ende des Kapitels 5.6.

# 5.1 Motivation und Zielsetzung

In den letzten Jahren hat sich durch die zunehmende Skalierung und Kommerzialisierung der Quantum Computing Hardware die Möglichkeit entwickelt, komplexere Problemstellungen auf eine völlig andere Weise als mit klassischen Lösungsmethoden anzugehen. Insbesondere die D-Wave Systeme, die die Quantum Annealing Heuristik in Hardware implementieren, sind aus industrieller und anwendungsbezogener Sicht, aufgrund ihrer vergleichsweise großen Qubitanzahl und größeren Konnektivität, interessant [42, 53, 61, 119, 167, 179].

Die Lösungsqualität beim Einsatz von entsprechender Hardware hängt sowohl von der Reife der Hardware, als auch von der Problemformulierung, beziehungsweise dem auszuführenden Quantenalgorithmus selbst, ab. Bei letzterem wurde bereits einiges an Forschungsaufwand in die Entwicklung von Ising-Hamiltonians für verschiedene Problemklassen betrieben [28, 75, 108], die bekanntermaßen als Eingabetyp für die D-Wave Systeme dienen. Um bereits auf NISQ-Computern eine potenziell bessere Lösungsqualität zu erreichen, können viele hardware-, aber auch softwarebezogene Hyperparameter angepasst werden [83, 114, 141, 180]. Jedoch befindet sich die angewandte Quantum Computing Forschung noch immer im Anfangsstadium und es ist wenig über die Hyperparameter der Ising-Hamiltonians bekannt und wie sie optimiert werden können, um die Problemformulierungen effektiv auf der Hardware auszuführen.

Mögliche Hyperparameter sind die bereits vorgestellten Penalty-Parameter eingeschränkter Hamiltonians, um deren Nebenbedingungen zu gewichten. Lucas stellt einige Ising-Hamiltonians für verschiedene eingeschränkte Optimierungsprobleme vor [108]. Bisher gibt es jedoch keine Faustregel, wie man deren Penalty-Parameter effizient, d.h., ohne Versuch und Irrtum oder Erraten, festlegen kann. Im vorherigen Kapitel wurde gezeigt, dass die Optimierung dieser Parameter mit Kreuzentropie zu einer Skalierung der minimalen spektralen Lücke führen kann, so dass die Lösungsqualität von den untersuchten Problemklassen bei Verwendung der D-Wave Quantum Annealing Heuristik zunahm [140]. Da jedoch keine Korrelation zwischen den optimierten Penalty-Parametern für die Probleminstanzen ersichtlich waren, musste man die zeitaufwändige Kreuzentropie-Methode für jede Instanz einzeln anwenden.

Dies motiviert die Untersuchung des Einflusses der Penalty-Parameter auf die minimale spektrale Lücke von sechs ausgewählten eingeschränkten Hamiltonians mit Hilfe von Methoden des maschinellen Lernens. Ziel ist es, die Notwendigkeit intensiver Parametertests durch die Verwendung eines auf einem neuronalen Netzwerk basierenden Regressionsmodells zur Vorhersage nützlicher Penalty-Parameter zu vermeiden und Korrelationen zwischen der minimalen Spektrallücke und den Penalty-Parametern zu identifizieren, um Richtlinien für die Wahl dieser Parameter für die untersuchten Problemklassen zu liefern. Die Ergebnisse zeigen, dass die Vorhersage der optimalen Penalty-Parameter in dem definierten Versuchsaufbau tatsächlich möglich ist und dass Korrelationen zur minimalen Spektrallücke zwischen und innerhalb der entsprechenden Problemklassen existieren.

## 5.2 Maschinelles Lernen

In diesem Abschnitt wird ein kurzer Überblick über notwendige Grundlagen der verschiedenen Methoden des maschinellen Lernens gegeben, die für dieses Kapitel relevant sind. Verfahren des MLs lassen sich grob in zwei Kategorien aufteilen: *Überwachtes* (engl. Supervised) und *unüberwachtes Lernen* (engl. Unsupervised Learning). Weitere Kategorien sind beispielsweise hybride Verfahren aus den beiden genannten Kategorien oder das sogenannte *Bestärkende Lernen* (engl. Reinforcement Learning), die aber für dieses Kapitel irrelevant sind und deshalb nur der Vollständigkeit halber erwähnt sein sollen.

1. Überwachtes Lernen

Das überwachte Lernen umfasst maschinelle Lernalgorithmen, bei denen *Labels* für die Daten vorliegen und für das Training verwendet werden können. Ziel ist es, ein Modell zu erlernen, das die korrekten Labels für ungesehene Daten vorhersagen kann. Dabei werden die Labels während das Trainings eingesetzt, um den Fehler in den Vorhersagen des Modells zu messen und diesen wiederum zur Verbesserung des Modells zu nutzen. Die bekanntesten Vertreter des überwachten Lernens sind die *Klassifikation* und die *Regression*. Bei ersteren werden Labels in Form von Kategorien angegeben (z.B. Auto und Fahrrad), bei letzteren als reelle Zahlen oder Vektoren (z.B. Alter, Temperatur, Körpergröße).

2. Unüberwachtes Lernen

Im Gegensatz zum überwachten Lernen gibt es beim unüberwachten Lernen keine Labels. Ein Großteil der unüberwachten Lernalgorithmen kann dem *Clustering* zugeordnet werden. Beim Clustering werden inhärente Muster und Informationen in den Daten identifiziert und diese anschließend einer Gruppe (engl. Cluster) zugeteilt. D.h., die Labels bzw. Cluster werden automatisch gelernt, ohne dass ein vorheriger Labeling-Prozess stattfinden musste. Welche Datenpunkte gruppiert werden, hängt weitestgehend von der Wahl der Distanzfunktion oder des Ähnlichkeitsmaßes ab. Die Ähnlichkeit zwischen hochdimensionalen Daten ist jedoch nicht trivial definierbar. Eine ungünstige Wahl des Ähnlichkeitsmaßes kann dazu führen, dass beispielsweise Bilddaten eher nach Farben als nach ihrer Semantik gruppiert werden. Viele Algorithmen des maschinellen Lernens können aufgrund der Distanzkonzentration, eine Besonderheit, die besagt, dass alle paarweisen Distanzen mit zunehmender Dimensionalität der Daten gegen den gleichen Wert konvergieren [2], nur sehr beschränkt auf hochdimensionalen Daten arbeiten. Ein weiteres Verfahren für das unüberwachte Lernen ist die *Dimensionalitätsreduktion* (engl. Dimensionality Reduction), die versucht dieser Problematik entgegenzuwirken. Ziel ist es, kompaktere, niedrigdimensionalere Darstellungen von Daten zu finden, die dann anstelle der eigentlichen Daten als Eingabe für Methoden des maschinellen Lernens verwendet werden können. Infolge gehen Informationen zwangsläufig verloren, aber der Speicherplatzbedarf und die Trainingszeit wird durch die Dimensionalitätsreduktion reduziert.

Im Rahmen dieser Arbeit wird die Clusteringanalyse aus der Kategorie des unüberwachten Lernens verwendet, um gewisse inhärente Trends und Datenmuster, welche die minimale Spektrallücke und das Verhältnis der Penalty-Parameter verschiedener eingeschränkter Hamiltonians beinhalten, zu erkennen. Des weiteren wird die Regressionsanalyse aus dem Bereich des überwachten Lernens eingesetzt, um optimale Penalty-Parametereinstellungen für ungesehene Probleminstanzen der Hamiltonians vorherzusagen. Im Nachfolgenden werden nun die zwei in dieser Arbeit verwendeten ML-Methoden beschrieben.

## 5.2.1 Clusteringanalyse

Clustering-Methoden identifizieren Teilmengen eines Datensatzes, welche hinsichtlich bestimmter Merkmale ähnlich oder gleich sind. Grundsätzlich gibt es viele verschiedene Ausprägungen von Clustering-Algorithmen. Die wichtigsten Ansätze sind das *schwerpunktbasierte Clustering*, *verteilungsbasierte Clustering*, *hierarchische Clustering* und das *dichtebasierte Clustering*. Eine umfassende Übersicht ist in [176] gegeben. Für die Aufgabenstellung in diesem Kapitel ist das dichtebasierte Clustering nützlich, dessen Eigenschaften am Ende dieses Abschnitts erläutert werden. Bei diesem Clustering werden Bereiche mit hoher Dichte zu Clustern verbunden. Auf diese Weise sind Cluster mit beliebiger Form und Ausprägung möglich, solange deren Dichte-Bereiche miteinander verbunden werden können. Der bekannteste Vertreter dieser Verfahren ist der *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) [58], der im Nachfolgenden angelehnt an [156] skizziert wird.

**DBSCAN** gruppiert typischerweise dichte Bereiche von Punkten im Datenraum, die durch Bereiche mit geringer Dichte getrennt sind. Als zentrale Ausgangslage steht die Abbildung  $\rho: X \to \mathbb{R}$ , die jedem Datenpunkt aus X einen Dichtewert zuweist, der sich aus der Anzahl an Punkten und einen benutzerdefinierten Radius  $\varepsilon$  zusammensetzt:

$$\rho_{\varepsilon}(x) = \sum_{s \in X \setminus \{x\}} \begin{cases} 1, \text{ falls } d(x, s) \leq \varepsilon \\ 0, \text{ falls sonst,} \end{cases}$$

wobei  $d(\cdot, \cdot)$  eine Distanzmetrik auf dem Merkmalsraum X ist. Anhand seiner Dichte wird nun jeder Datenpunkt einer Kategorie {Kernpunkt, Randpunkt, Rauschpunkt} zugeordnet:

 $c_{\varepsilon,minPkt}(x) = \begin{cases} \text{Kernpunkt, falls } \rho_{\varepsilon}(x) \ge minPkt \\ \text{Randpunkt, falls } \exists s \in X : c_{\varepsilon,minPkt}(s) = \text{Kernpunkt} \land \\ d(x,s) \le \varepsilon \\ \text{Rauschpunkt, sonst,} \end{cases}$ 

wobei minPkt die benutzerdefinierte minimale Punkteanzahl pro Cluster angibt. Anschließend wird ein Graph G mit allen Kernpunkten als Knoten aufgestellt und mit Kanten zwischen den Knoten ergänzt, wenn deren Abstand kleiner gleich  $\varepsilon$  ist. Die daraus resultierenden verbundenen Strukturen repräsentieren die gefundenen Cluster. Schlussendlich werden dann alle Randpunkte den Clustern ihrer nächstgelegenen Kernpunkte zugeordnet und die Rauschpunkte bleiben als Ausreißer getrennt von den Clustern bestehen.

DBSCAN ist vor allem in Szenarien von Vorteil, bei denen die Anzahl der Cluster unbekannt ist. Zudem ist der Algorithmus aufgrund seines dichtebasierten Verfahrens in der Lage, Cluster beliebiger Form und Gestalt zu finden und ist darüber hinaus robust gegenüber Ausreißern. Unterscheidet sich jedoch die Dichte des Datensatzes lokal stark, kann der von DBSCAN einzeln verwendete Dichteschwellwert, der mittels  $\varepsilon$  und minPkt durch den Benutzer definiert wird, hingegen problematisch sein.

In Abschnitt 5.4.2.1 wir die Umsetzung und die verwendeten Parameter des DBSCAN Algorithmus zur Clusteringanalyse in dieser Arbeit beschrieben.

#### 5.2.2 Regressionsanalyse

Als Regressionsanalyse wird eine Reihe von statistischen Verfahren zur Schätzung der Korrelationen zwischen einer oder mehrerer abhängigen Variablen und einer oder mehrerer unabhängigen Variablen bezeichnet. Die Problemstellung in dieser Arbeit umfasst das Lernen und die Vorhersage der Zusammenhänge zwischen den einzelnen Probleminstanzen der eingeschränkten Optimierungsprobleme und deren Penalty-Parametereinstellungen, um eine bestmögliche Skalierung der minimalen Spektrallücke zu erreichen. Es gibt verschiedene Ansätze, um solche Regressionsanalyseverfahren umzusetzen [21]. In dieser Arbeit werden künstliche neuronale Netze (KNNs) verwendet, die einige vorteilhafte Eigenschaften mit sich bringen und im Folgenden in Verbindung mit der Regression erläutert werden.

**KNNs** sind im ursprünglichen Sinn eine abstrakte, vereinfachte Umsetzung der Verknüpfung biologischer Neuronen, wie sie im Gehirn komplexer Lebensformen vorkommen, zur Anwendung im maschinellen Lernen. Die grundlegende Basis der KNNs bilden sogenannte Perzeptrone [146]. In Abbildung 5.1 (links) ist ein einzelnes Perzeptron dargestellt, das aus einer Eingabeschicht mit n Eingaben, einem Neuron, einer Aktivierungsfunktion und einer Ausgabe besteht.



Abbildung 5.1: Links: Konzept eines einfachen Perzeptrons. Rechts: Mehrschichtiges Perzeptron mit einer Eingabeschicht (rot), einer Ausgabeschicht (grün), mehreren verdeckten Schichten (orange). Jedes Perzeptron ist mit jedem anderen der benachbarten Schicht verbunden.

Mehrere Perzeptrone können in Schichten kombiniert werden und formen dann mehrschichtige Perzeptrone (MSPs), vgl. Abbildung 5.1 (rechts). Dabei ist jedes Perzeptron im Netz mit jedem Perzeptron in den benachbarten Schichten verbunden. Zwischen der Eingabe- und Ausgabeschicht können beliebig viele weitere Schichten hinzugefügt werden, die dann als verdeckte Schichten bezeichnet werden. Der Prozess, der die Daten durch ein oder mehrere verbundene Perzeptrone propagiert wird als Forward Pass bezeichnet. Dabei wird für jede Eingabe der Eingabewert  $x \in \mathbb{R}^n$  mit dem Gewicht  $w \in \mathbb{R}^n$  multipliziert und aufsummiert, vgl. Gleichung (5.1), wobei b ein Bias ist, der auf die Summe der multiplizierten Werte addiert wird. Dieser Bias ist in den meisten Fällen notwendig, um die gesamte Aktivierungsfunktion nach links oder rechts zu verschieben, um die gewünschten Ausgabewerte zu erzeugen. Die Gewichte w beschreiben zudem, wie groß der Einfluss der jeweiligen Eingabe auf die Ausgabe des Neurons sein soll.

$$z = \sum x \cdot w + b, \tag{5.1}$$

Anschließend wird der erhaltene Wert z in eine Aktivierungsfunktion  $\sigma(z)$  weitergegeben. Mit deren Anwendung erhält man dann den vom Modell vorhergesagten Wert,

$$\hat{y} = \sigma(z). \tag{5.2}$$

Als Aktivierungsfunktionen kommen meist nicht-lineare, differenzierbare Funktionen zum Einsatz, wie z.B. die Softmax-Funktion, der hyperbolische Tangens oder der sogenannte Rectifier [81]. Mit diesen lassen sich die Ausgaben in einen gewünschten Wertebereich überführen. Beispielsweise kann man mit der Softmax-Funktion Wahrscheinlichkeiten modellieren, sodass  $\hat{y} \in [0, 1]$  annimmt. Nach Anwendung der Aktivierungsfunktion wird dann anhand einer Fehlerfunktion E die Vorhersagen des Modells mit den wahren Werten verglichen. Als Fehlerfunktion können je nach Anwendungsfall verschiedene Funktionen gewählt werden [170]. In Bezug auf die Regressionsanalyse ist die *mittlere quadratische Abweichung* (engl. Mean Squared Error (MSE)) eine bewährte Wahl, die die Differenz zwischen dem wahren Wert  $(y_i)$  und dem durch das Modell vorhergesagten Wert  $(\hat{y}_i)$  quadriert und über den ganzen Datensatz mittelt,

$$E = MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$
 (5.3)

Um nun den berechneten Fehler zu minimieren und die Vorhersagen des Modells kontinuierlich zu verbessern, müssen die Gewichte des Perzeptrons entsprechend iterativ angepasst werden. Daher wird im nächsten Schritt die Fehlerrückführung (engl. Backpropagation of Errors) und anschließende Optimierung der Gewichte des Perzeptrons ausgeführt [52, 71]. Die Fehlerrückführung wird zur Berechnung des Gradienten der Fehlerfunktion in Bezug auf die Gewichte verwendet. Dabei nutzt sie die Kettenregel, um die partielle Ableitung jedes Parameters zu berechnen. Im Anschluss erhält man den Gradienten der Fehlerfunktion, der die Richtung des steilsten Anstiegs der Funktion angibt. Um nun die Fehlerfunktion anhand des Gradienten zu minimieren, wird ein Gradientenabstiegsverfahren verwendet [71]. Hier bewegt man sich unter Berücksichtigung einer Lernrate  $\alpha$  im Parameterraum (Gewichte und Bias) in die entgegengesetzte Richtung des Gradienten, um im besten Fall das globale Minimum der Parameter zu finden, das die Fehlerfunktion minimiert. Die Lernrate beschreibt dabei einen Hyperparameter, der die Schrittgröße bei jeder Iteration des Gradientenabstiegs bestimmt. Sie definiert indirekt die Geschwindigkeit mit der man sich durch die Parameterlandschaft bewegt. Anschließend werden die Parameter wie in Gleichung (5.4) aktualisiert und der Prozess wird solange wiederholt, bis die Fehlerrückführung und das Gradientenabstiegsverfahren konvergieren und keine Verbesserung in den Vorhersagen des Modells mehr möglich sind.

$$w_{i} = w_{i} - \left(\alpha \cdot \frac{\partial E}{\partial w_{i}}\right)$$
  
$$b = b - \left(\alpha \cdot \frac{\partial E}{\partial b}\right)$$
  
(5.4)

Das vorgestellte Verfahren ist wie bereits erwähnt auch auf mehrschichtige Perzeptrone anwendbar. Ein Vorteil bei der Verwendung von mehrschichtigen Perzeptronen und geeigneten Aktivierungsfunktionen für die Regression, aber auch für andere Problemstellungen ist, dass im Vergleich zur einfachen *linearen Regression*, nicht-lineare Zusammenhänge abgebildet werden können, und sie somit auch für komplexere Problemstellungen verwendet werden können.

Dieser Vorteil wird unter anderem im sogenannten Universal Approximation Theorem subsumiert, das beschreibt, dass jede beliebige Eingabe-AusgabeFunktion zwischen zwei Euklidischen Räumen angenähert bzw. dargestellt werden kann [87, 88]. In Abschnitt 5.4.2.2 wird das implementierte KNN zur Regressionsanalyse für diese Arbeit beschrieben.

# 5.3 Verwandte Arbeiten

Die Untersuchung der minimalen Spektrallücke und des Eigenspektrums von Hamiltonians spielt eine wesentliche Rolle, um entsprechende Problemformulierung effektiv auf Quantum Computing Hardware auszuführen. Coffey transformierte in seiner Fallstudie das Knapsack Problem in den entsprechenden Ising-Hamiltonian und untersuchte dessen Lösbarkeit innerhalb eines klassisch simulierten AQC-Framework. Aufgrund der geringen Lösungqualität kommt er zu dem Schluss, dass sowohl theoretische, als auch numerische Analysen in Bezug auf die minimale Spektrallücke in der adiabatischen Evolution von großer Bedeutung sind, um das AQC zu verbessern [39].

Farhi et al. argumentieren anhand des 3-SAT Problems, dass man durch zufällige Änderung des initialen Hamiltonians (mit großer Wahrscheinlichkeit) eine adiabatische Evolution erhält, die es dem AQC-Prozess erleichtert im Grundzustand zu verweilen. Sie zeigen für 3-SAT Probleminstanzen, dass die Variation des initialen Hamiltonians einen Einfluss auf die minimale Spektrallücke haben kann und empfehlen daher den adiabatischen Algorithmus für jede Instanz mit verschiedenen zufälligen initialen Hamiltonians auszuführen, um die Wahrscheinlichkeit zu erhöhen, im Grundzustand des Problem-Hamiltonians zu erreichen [59].

Wie bereits in Kapitel 4.3 erwähnt, zeigte Choi theoretisch anhand des MIS, dass man ebenso durch Anpassung der Penalty-Parameter des entsprechenden Problem-Hamiltonians die minimale Spektrallücke beeinflusst kann [37].

In Anlehnung an diese Erkenntnisse wurde in Kapitel 4 und der zugrundeliegenden Vorveröffentlichung [140] die Kreuzentropie-Optimierungsmethode zur automatischen Anpassung der Penalty-Parameter von drei eingeschränkten Hamiltonians (KP, MECP, SPP) vorgestellt, um deren minimale Spektrallücken zu skalieren. Auf diese Weise konnte eine verbesserte Lösungsqualität in Bezug auf das Finden des globalen Optimums auf D-Wave Hardware erzielt werden, d.h., die Wahrscheinlichkeit das globale Optimum zu messen, stieg. Allerdings waren keine Korrelationen zwischen den optimierten Penalty-Parametern der einzelnen Probleminstanzen ersichtlich, sodass man die rechenintensive KE-Methode für jede Instanz einzeln anwenden musste. Dies motiviert die Untersuchung des Energiespektrums bzw. der minimalen Spektrallücke verschiedener Hamiltonians mit Methoden des maschinellen Lernens, um Muster und Richtlinien zu finden, wie die entsprechenden Penalty-Parameter zu setzen sind.

Es gibt einige verwandte Arbeiten, die klassische (ML) Methoden zur Optimierung aber auch Generierung von Quantenalgorithmen einsetzen. Dieses Forschungsfeld ist in der Literatur als *Quantum Control Optimization* bekannt und befasst sich mit Methoden und Möglichkeiten, quantenmechanische Systeme effizient so zu manipulieren, um ein gewünschtes Verhalten zu erzeugen. Beispielsweise wurden Methoden des bestärkenden Lernens oder genetische Algorithmen zur automatischen quantenmechanischen Zustandspräparierung vorgeschlagen [27, 69, 123, 132]. Zudem werden klassische *Empfehlungsdiens*te (engl. Recommender Systems (RS)) eingesetzt, um den Anwendern und Entwicklern Feedback und Empfehlungen zur Parameterwahl von Quantenalgorithmen zu liefern [13]. Aber auch auf Hardware- und Algorithmenebene gibt es bereits Empfehlungsdienste. Salm et al. präsentieren ein Konzept für die automatische Analyse und Auswahl von QC-Implementierungen und geeigneter -Hardware, die eine ausgewählte Implementierung mit bestimmten Eingabedaten ausführen können [152]. In Bezug auf die Inhalte dieses Kapitels ist zum aktuellen Zeitpunkt keine Arbeit bekannt, die ML-Methoden zur Analyse und zum Lernen der Penalty-Parameter eingeschränkter Hamiltonians und deren Einfluss auf die minimale Spektrallücke einsetzt.

# 5.4 Aufbau der Experimente

#### 5.4.1 Datengenerierung

In diesem Abschnitt wird die Erstellung der Daten beschrieben, die zur Berechnung der Informationen - d.h., das Penalty-Parameterverhältnis, die minimale Spektrallücke und ihre zeitliche Lage in Bezug auf den Annealing-Prozess - als Input für die ML-Analysemethoden notwendig sind. Wie in Abschnitt 2.3.3 erwähnt, hat jeder hier untersuchte eingeschränkte Hamiltonian ein bestimmtes, gültiges, halboffenes Intervall für den Penalty-Parameter A. Während der Penalty-Parameter B auf eins festlegt wurde, wurde das Stichprobenintervall des Penalty-Parameters A entsprechend der problemspezifischen Nebenbedingung berechnet, siehe Gleichungen (2.14), (2.16), ..., (2.24).

Am Beispiel des MECP wurde das Stichprobenintervall für Aauf  $[B \cdot n + 0.1, B \cdot n + 5.0]$  gesetzt, wobei n die Anzahl der Mengen der Probleminstanz beschreibt. Das MECP-Penalty-Parameterverhältnis wurde anhand von  $\frac{B \cdot n}{A}$  berechnet. Zu erwähnen ist, dass das im Allgemeinen halboffene Intervall von A auf  $[B \cdot n + 0.1, B \cdot n + 5.0]$  beschränkt wurde. Obwohl man theoretisch ein größeres Stichprobenintervall verwenden könnte, muss man bedenken, dass die Autoskalierungsfunktion von D-Wave jeden Koeffizienten des Ising-Hamiltonians auf einen bestimmten, von der Hardware abhängigen, Bereich von  $h_i$  und  $J_{ij}$  skaliert [46]. Das bedeutet, dass der größte Koeffizient des Ising-Modells (vermutlich durch den Penalty-Parameter A beeinflusst) auf die obere Grenze des gültigen Bereichs skaliert wird und alle anderen Werte entsprechend proportional kleiner abgebildet werden. Um gute und sinnvolle Lösungen zu erhalten, sollte man also die Ising-Modell-Koeffizienten der Nebenbedingungen groß genug wählen, um gültige Lösungen zu gewährleisten, aber klein genug, um die Bedeutung der einzelnen Ising-Hamiltonian-Terme

zu erhalten.

Für jede Probleminstanz wurden 50 gleichmäßig verteilte Penalty-Parameter für A aus dem entsprechenden Intervall gezogen, um die Änderungsrate der minimalen Spektrallücke und ihre Position im Annealing-Prozess zu ermitteln. Die minimale Spektrallücke wurde gemäß der Gleichung (2.25) aus Abschnitt 2.3.4 und den Annealing-Funktionen  $\mathcal{A}(t)$  und  $\mathcal{B}(t)$  des D-Wave Advantage 4.1 Systems für jeden eingeschränkten Hamiltonian mit den gezogenen Penalty-Parametern berechnet [45]. Zudem wurde vor der Berechnung der minimalen Spektrallücke die D-Wave Autoskalierungsfunktion auf die Daten angewendet, sodass die Experimente das Verhalten der D-Wave Hardware möglichst genau abbilden [46].

Um die folgenden Experimente in Abschnitt 5.5 zu verstehen, sind in Abbildung 5.2 die entsprechenden Rohdaten von 25 zufälligen Instanzen für jede Problemklasse visualisiert. Die Abbildungen stellen die Korrelationen zwischen der minimalen Spektrallücke und ihrer Position im Annealing-Prozess und dem Penalty-Parameterverhältnis dar. Für jede Probleminstanz (dargestellt durch eine Farbe) wurden 50 gleichmäßig verteilte Penalty-Parameterpaare (dargestellt durch die Punkte) gezogen. Zu erwähnen ist, dass sich innerhalb der Abbildungen einige Trajektorien überdecken können. Die x-Achse zeigt das Penalty-Parameterverhältnis, welches, wie gerade beschrieben, problemspezifisch berechnet wurde. Der Annealing-Prozess und die Lage der minimalen Spektrallücke wurden zur Veranschaulichung diskretisiert (0 bis 100), wobei 0 die Startzeit des Annealing-Prozesses angibt, vergleiche y-Achse. Die Größe der minimalen Spektrallücke ist auf der z-Achse angegeben. Wie in Abschnitt 2.2.1 erläutert, ist in der Theorie, die Wahrscheinlichkeit während des Annealings im Grundzustand zu verbleiben umso höher, je größer die minimale Spektrallücke ist.

Damit jedoch die Trends der minimalen Spektrallücke, deren Lage im Annealing-Prozess und das Verhältnis der Penalty-Parameter mit Methoden des maschinellen Lernens analysiert werden können, wurden pro Problemklasse 1000 indivduelle Probleminstanzen erzeugt. Außerdem gilt, dass die Größe der Instanzen auf acht Variablen bzw. logische Qubits beschränkt wurde, da die numerische Berechnung des Eigenspektrums rechenintensiv ist und der Berechnungsaufwand mit der Anzahl der Variablen bzw. Matrixgröße steigt.

Um einen solch großen Datensatz von Ising-Hamiltonian-Instanzen und deren minimale Spektrallücken für das Training von ML-Methoden zu generieren, wurden die Eigenspektren der vergleichsweise kleineren logischen Hamiltonians der Probleminstanzen berechnet, die sich in gewissem Maße von denen, die letztendlich in der D-Wave Hardware eingebettet werden, unterscheiden. Dieses Thema wird in Detail in Abschnitt 5.5.6 erörtert.



Abbildung 5.2: Visualisierung der Korrelationen zwischen der minimalen Spektrallücke und ihrer Position im Annealing-Prozess und dem Penalty-Parameterverhältnis für 25 zufällige Instanzen pro Problemklasse. Für jede Probleminstanz (dargestellt durch eine Farbe) wurden 50 gleichmäßig verteilte Penalty-Parameterpaare (dargestellt durch die Punkte) gezogen.

# 5.4.2 Methoden zur Evaluierung

#### 5.4.2.1 Clusteringmethode

Um die Muster und Ähnlichkeiten zwischen und innerhalb der Problemklassen in den erzeugten Trajektorien der minimalen Spektrallücke zu bestimmen, wurde DBSCAN verwendet, ein dichtebasiertes Clustering-Verfahren, das beliebig große und geformte Cluster in hochdimensionalen Daten identifiziert kann [58]. Die beiden wichtigsten Hyperparameter von DBSCAN sind  $\varepsilon$  und minPkt. In einem Preprocessing-Schritt wurden die Daten auf den Bereich von 0 und 1 normalisiert und die paarweise Distanzmatrix der dreidimensionalen Trajektorien (Größe der minimalen Spektrallücke, Lokalisierung und das Penalty-Parameterverhältnis) als Eingabe für DBSCAN gegeben. Als Clustering-Metrik wurde der Pearson-Korrelationskoeffizient (engl. Pearson Correlation Coefficient (PCC)) zwischen der paarweisen Distanzmatrix und den gefundenen Labels des DBSCAN Algorithmus verwendet.

#### 5.4.2.2 Regressionsmethode

Für die Vorhersage der Penalty-Parameter, die mit der größten minimalen Spektrallücke für eine bestimmte Probleminstanz korrelieren, wurden KNNs mit vollvermaschten Schichten (engl. Dense Laver) verwendet. Die Architektur wurde grundsätzlich einfach gehalten, ohne versteckte Schichten und mit Rectifier als Aktivierungsfunktion zwischen der Eingabe- und der Ausgabeschicht. Für die Eingabeschicht wurden die Probleminstanzen auf einen Bereich von [-1; +1] normalisiert und in einen eindimensionalen Vektor überführt. Hierfür werden einfach die mehrdimensionalen Merkmale der Instanzen hintereinander gehängt. Dieses Verfahren wird als *Flattening* bezeichnet. Da die Probleminstanzen zum Teil unterschiedlich groß sind, führt das beim Flattening zu unterschiedlich großen Vektoren für die Eingabeschicht des KNNs. Eine konventionelle Möglichkeit, dieses Problem zu umgehen, ist das sogenannte Zero-Padding, das dafür sorgt, dass alle Probleminstanzen entsprechend der größten erzeugten Probleminstanz mit Nullen aufgefüllt werden. Die Anzahl der Ausgangsneuronen des KNNs wurde gemäß der Penalty-Parameter A und B auf zwei gesetzt.

Für die Optimierung der KNN Parameter wurde der klassische Optimierer Adam, ein stochastisches Gradientenabstiegsverfahren, gewählt [101]. Die besten Ergebnisse wurden nach Hyperparameteroptimierng mit einer Lernrate von 0,001 für MECP, MCP, MVCP, BILP, KP und 0,0005 für SPP erzielt. Für das Training wurde für alle Problemklassen außer SPP eine Batchgröße von 50 beziehungsweise 32 verwendet. Der Datensatz wurde in 90% zu 10% in Trainings- und Testdaten aufgeteilt. Zur Bewertung des Regressionsmodells wurde die *Wurzel der mittleren quadratischen Abweichung* (engl. Root Mean Squared Error (RMSE)) und der Determinationskoeffizient ( $R^2$ ) als Metriken verwendet, um festzustellen, wie gut das Modell die besten Penalty-Parameter in absoluten Zahlen vorhersagen kann, beziehungsweise wie gut die abhängige

Variable (Penalty-Parameterverhältnis) anhand der betrachteten unabhängigen Variablen (Merkmale der Probleminstanzen) vorhergesagt werden kann.

#### 5.4.2.3 Quantum Annealing Hardware

Für die Bewertung der Lösungsqualität der vorhergesagten Penalty-Parameter auf realer Hardware wurde das D-Wave Advantage 4.1 System verwendet. Die Lösungsqualität ist mit dem bereits bekannten Näherungsverhältnis gleich gesetzt, das wie folgt berechnet wird: Näherungsverhältnis =  $\frac{\#BKS}{\#Messungen}$ , wobei #BKS die Anzahl der Messungen der besten bekannten Lösung und #Messungen die Gesamtzahl der Messungen ist (Standardwert ist 100). Für die Experimente wurde eine vollständige Hardware-Graphstruktur der Größe 8 verwendet, um die logischen Ising-Hamiltonians auf diese abzubilden. Da alle zufällig generierten Probleminstanzen die gleiche Größe haben (8 Variablen). lässt sich jede in diese vollständige Graphstruktur der Hardware abbilden. Der einzige Unterschied könnte darin bestehen, dass bestimmte Coupler der Hardware nicht verwendet werden, wenn der logische Ising-Hamiltonian nicht vollständig vermascht ist. In diesem Fall wird der Coupler auf null gesetzt. Da andere Arbeiten [140] bereits gezeigt haben, dass das Hardware-Embedding auch die Lösungsqualität beeinflusst, wurde immer dieselbe Graphstruktur der Hardware (mit denselben Qubits und Couplern) für jede Problemklasse wiederverwendet, um sie hinsichtlich des Näherungsverhältnisses, unabhängig von der Qualität bzw. dem Rauschen der Hardware-Qubits, vergleichbar zu machen. Darüber hinaus wurde den D-Wave Richtlinien folgend, der Parameter Chain Strength auf eins höher als der Betrag des maximalen Ising-Modell-Koeffizienten gesetzt, um gleich geschaltete Qubit-Ketten zu gewährleisten [44].

# 5.5 Ergebnisse und Diskussion

In diesem Abschnitt wird die Optimierung der Penalty-Parameter von den vorgestellten Ising-Hamiltonians und deren Zusammenhang mit der minimalen Spektrallücke empirisch untersucht. Zudem werden die gewonnenen Erkenntnisse praktisch auf dem D-Wave Advantage System angewendet, um deren Effekt auf das Näherungsverhältnis zur jeweiligen BKS zu bestimmen.

### 5.5.1 Identifizieren von Trends im Eigenspektrum von eingeschränkten Hamiltonians

Da es grundsätzlich schwierig ist, Trends und Muster in den 1000 generierten Probleminstanzen pro Problemklasse visuell zu erkennen, wurde, wie in Abschnitt 5.4 beschrieben, DBSCAN zum Clustern der generierten Trajektorien verwendet. In Abbildung 5.3 sind die DBSCAN-Clustering Ergebnisse der minimalen Spektrallücke und ihre Lage im Annealing-Prozess im Zusammenhang mit den Penalty-Parameterverhältnissen aufgetragen. Die Achsenbeschriftung ist identisch wie in Abbildung 5.2 gewählt. In jeder Teilabbildung ist die gemittelte Trajektorie jedes gefundenen Clusters und das 95%-Konfidenzintervall als Fehlerbalken eingezeichnet. Zusätzlich ist das Cluster-Label in Textform annotiert. Tabelle 5.1 zeigt die zugehörigen besten gefundenen Werte der DBSCAN-Hyperparameter und den erzielten Pearson-Korrelationskoeffizienten für jede Problemklasse, der als Metrik für das Clustering verwendet wurde.

Problemklasse	ε	minPkt	PCC
MCP	1.17	6	-0.962
MVCP	0.34	6	-0.929
KP	0.55	48	-0.994
SPP	0.37	24	-0.889
BILP	0.46	12	-0.919
MECP	0.22	24	0.911

Tabelle 5.1: Die besten gefundenen DBSCAN-Hyperparameter für jede Problemklasse. Der Pearson-Korrelationskoeffizient zwischen der paarweisen Distanzmatrix der Trajektorien und den Cluster-Labels wurde als Metrik für das Clustering verwendet.

Im Allgemeinen ähneln sich die gefundenen Cluster von MVCP, MCP und SPP insofern, als dass sie alle mehrere tief liegende Cluster in Bezug auf die Größe der minimalen Spektrallücke (Trajektorien im unteren Bereich in jeder Teilabbildung) und einer MSG-Position von etwa 60 - 90 im diskretisierten Annealing-Prozess aufweisen (vgl. y-Achse). Darüber hinaus besitzen sie alle ein Cluster, das mit zunehmendem Penalty-Parameterverhältnis und einer vergleichsweise frühen Position von 10 - 20 in Bezug auf die Größe der minimalen Spektrallücke ansteigt (vgl. Abbildungen 5.3b, 5.3c und 5.3e). Das bedeutet, dass die Probleminstanzen dieser Cluster theoretisch mit einem bestimmten Penalty-Parameterverhältnis in Bezug auf die Größe der minimalen Spektrallücke optimiert werden können. Außerdem ist anzunehmen, dass die frühe minimale Spektrallücke im Annealing-Prozess dieser Instanzen günstig sein könnte, da die Kohärenzzeiten von NISQ-Rechnern, wie den D-Wave Systemen, noch begrenzt ist. Mit einer frühen minimalen Spektrallücke könnte die Wahrscheinlichkeit, aufgrund von natürlich auftretender Dekohärenz vom Grundzustand in einen angeregten Zustand zu springen, verringert werden. Zudem ist in Bezug auf das MCP zu erkennen, dass es ein Cluster (Label -1) mit einer vergleichsweise großen minimalen Spektrallücke gibt. Unter-



Abbildung 5.3: Clustering Ergebnisse der minimalen Spektrallücke und ihrer Lage in Bezug auf die Penalty-Parameterverhältnisse. Um die 1000 Probleminstanzen in jeder Teilabbildung zu visualisieren, ist für jedes Cluster die gemittelte Trajektorie und ihr 95%-Konfidenzintervall als Fehlerbalken dargestellt. Jedes Cluster ist zudem eindeutig gefärbt und sein Label in Textform annotiert.

suchungen haben gezeigt, dass die MCP-Instanzen in diesem Cluster vollständige Graphen beziehungsweise Cliquen entsprechen, was wiederum ein triviales MCP darstellt und in der Ising-Hamiltonian-Formulierung zu einer Aufhebung der Nebenbedingung bzw. dem Penalty-Parameter A führt, siehe Gleichung (2.19). Diese Instanzen können daher nicht mit einem bestimmten Penalty-Parameterverhältnis optimiert werden und ist dementsprechend in Abbildung 5.3c als horizontale Trajektorie visualisiert.

MECP, KP und BILP besitzen alle unterschiedliche Cluster. Wie bereits in den Rohdaten zu sehen war, gibt es bei MECP Instanzen mit unterschiedlichen Trends im Clustering (vgl. Abbildungen 5.2a und 5.3a). Mit zunehmendem Penalty-Parameterverhältnis hat MECP ein Cluster (Label -1) ohne Steigung und drei Cluster mit Aufwärts- (Label 0 und 1) bzw. Abwärtstrends (Label 2). KP besitzt zwei Cluster, die beide einen vergleichsweise geringen Aufwärtstrend mit zunehmendem Penalty-Parameterverhältnis aufweisen (vgl. Abbildung 5.3d). Innerhalb des BILP-Clusterings bleiben alle gefundenen Cluster trotz unterschiedlichem Penalty-Parameterverhältnis bei der gleichen minimalen Spektrallückengröße. Drei Cluster (Label -1, 7 und 8) besitzen eine relativ kleine minimale Spektrallücke und eine vergleichsweise späte Position (80 - 100) im Annealing-Prozess, während die übrigen Cluster größere minimale Spektrallücken in einem frühen Stadium (20 - 40) im Annealing-Prozes aufweisen (vgl. Abbildung 5.3f). Bei genauerer Betrachtung wurde festgestellt, dass sich die BILP-Instanzen nicht in den On- und Off-Diagonalwerten der Ising-Hamiltonian-Formulierung, bzw. ihren  $h_i$  und  $J_{ij}$  des Ising-Modells unterscheiden. Daher kann das Verhältnis der Penalty-Parameter nicht angepasst, beziehungsweise geändert werden, siehe Gleichung (2.23).

Aufgrund dieser Beobachtungen wäre interessant zu wissen, ob vor allem für die drei Problemklassen MVCP, MCP und SPP, die jeweils Instanzen bzw. Cluster mit vergleichsweise großer Steigung in ihren MSG-Trajektorien besitzen, vorab entscheidbar wäre, wie die Penalty-Parameter für die jeweilige Instanz gesetzt werden müssen, um eine große minimale Spektrallücke zu erhalten. Deshalb wird nun im nächsten Schritt untersucht, ob ein neuronales Netz anhand der Probleminstanz das Penalty-Parameterverhältnis, das mit der größten minimalen Spektrallücke korreliert, vorhersagen kann.

### 5.5.2 Lernen der Penalty-Parameter für eingeschränkte Hamiltonians

Abbildung 5.4 zeigt die resultierenden RMSE- und  $R^2$ -Koeffizienten beim Training der KNN Modelle, einschließlich des 95%-Konfidenzintervalls über 10 Durchläufe. Trivialerweise erreicht der  $R^2$ -Koeffizient am Ende des Trainings etwa 0.981  $\pm$  0.002 bis 1.0  $\pm$  0.0 für KP, MVCP, MCP und SPP, da für diese Probleme das beste Parameterverhältnis darin besteht, einfach das maximale Verhältnis zu wählen, d.h., bei dem der Parameter A auf den niedrigsten möglichen Wert in dem definierten Stichprobenintervall gesetzt wird. Auch der RMSE konvergiert gegen 0.0, was bedeutet, dass der Fehler zwischen den von den Regressionsmodellen vorhergesagten Penalty-Parametern und den tatsächlich besten Penalty-Parametern sehr klein oder sogar null ist. Überraschenderweise ist das Training eines guten Modells für MECP und BILP schwieriger. Der beste erreichte  $R^2$ -Koeffizient war  $0,918 \pm 0,048$  bzw.  $0,959 \pm 0,012$ . Dies spiegelt sich auch in einem vergleichsweise schlechteren RMSE von  $0,091 \pm 0,048$  und  $0,072 \pm 0,001$  wider. Ein möglicher Grund dafür ist in dem entsprechenden Clustering zu sehen, vgl. Abbildung 5.3. Während KP, MVCP, MCP und SPP alle keinen oder einen Aufwärtstrend mit zunehmendem Penalty-Parameterverhältnis aufweisen, enthält MECP entsprechend Clusters mit Auf- und Abwärtstrends, was dazu führt, dass unterschiedliche Penalty-Parameter am besten sind. Vermutlich führt dies zu einer etwas schlechteren Vorhersage der Parameter. Bei BILP hat kein Cluster eine Steigung. Dies resultiert in gleich gute Penalty-Parameterverhältnisse in Bezug auf die Größe der minimalen Spektrallücke. Vermutlich führt dies beim Trainieren des Modells zu einem ähnlichen Verhalten wie beim MECP.



Abbildung 5.4:  $R^2$ - (a) und RMSE-Koeffizienten (b) während des Trainings inklusive des 95%-Konfidenzintervalls über 10 Durchläufe für die sechs untersuchten Problemklassen. Ein großer  $R^2$  und ein kleiner RMSE-Koeffizient sind günstig.

### 5.5.3 Anwendung vorhergesagter und zufälliger Penalty-Parameter auf Quantum Annealing Hardware

In diesem Abschnitt wird getestet, ob die Wahl des besten Penalty-Parameterverhältnis aus dem vordefinierten spezifischen Stichprobenintervall das BKS-Näherungsverhältnis auf dem D-Wave Advantage 4.1 System beeinflusst und ob die Verwendung eines KNNs für die Vorhersage der Penalty-Parameter im Vergleich zur zufälligen Wahl aus dem gültigen spezifizierten Bereich von Vorteil ist.

Für jede der Problemklassen wird ein Satz von 100 zufälligen Probleminstanzen generiert und ein Penalty-Parameterpaar aus dem KNN-Modell abgeleitet. Anschließend wird der Ising-Hamiltonian unter Verwendung des vorhergesagten Penalty-Parameterpaar für diese 100 Probleminstanzen formuliert und auf der D-Wave Hardware ausgeführt.



Abbildung 5.5: Darstellung des durchschnittlichen BKS-Näherungsverhältnis des Modells (rot) in Bezug auf die vorhergesagten Penalty-Parameter gegenüber dem Näherungsverhältnis der zufällig ausgewählten Penalty-Parameter (grün), einschließlich des 95%-Konfidenzintervalls über 50 Iterationen und jeweils 100 Probleminstanzen. Zu beachten ist, dass das durchschnittliche Näherungsverhältnis des Modells als Basiswert verwendet und somit auf 1.0 gesetzt wurde, während das Näherungsverhältnis der zufällig ausgewählten Penalty-Parameter relativ zu dem des Modells aufgetragen ist. Bezüglich der zufällig gewählten Parameter wurde das aktuell bisher beste erzielte Näherungsverhältnis über die Iterationen beibehalten.

Abbildung 5.5resultierende durchschnittliche BKS-In ist das Modells 95%-Näherungsverhältnis des als rote Linie mit dem Konfidenzintervall über die 100 Probleminstanzen pro Klasse dargestellt. Im nächsten Schritt startet folgendes Experiment: Für 50 Iterationen wird ein zufälliges Penalty-Parameterpaar aus dem vordefinierten Bereich entnommen, der Ising-Hamiltonian formuliert und auf dem D-Wave Quantum Annealer ausgeführt. Dabei wird das zum aktuellen Zeitpunkt beste Näherungsverhältnis beibehalten (grüne Linie) und das 95%-Konfidenzintervall berechnet. Obwohl das Näherungsverhältnis wie in Abschnitt 5.4.2.3 angegeben, berechnet wurde, wurde das durchschnittliche Näherungsverhältnis des Modells als Basiswert verwendet und daher auf 1.0 gesetzt, während das Näherungsverhältnis der zufällig gezogenen Penalty-Parameterpaare relativ zu dem des Modells abgebildet wurde, um explizit zu veranschaulichen, an welchem Punkt der Zufallsstichprobenprozess die Basislinie erreicht oder überschreitet.

Für MVCP, MCP und SPP schneidet das KNN-Modell vergleichsweise gut ab. Im Falle von MVCP dauert es im Durchschnitt 43 Iterationen, bis der Zufallsprozess das Modell erreicht. Im Falle von SPP dauert es ungefähr 50 Iterationen, bis das Modell eingeholt wird, während in Bezug auf das MCP der Zufallsprozess das Modell in den 50 Stichproben, die generiert wurden, nie erreicht. Erwartungsgemäß sind diese drei Probleme auch diejenigen, bei denen die Inferenz einfach ist (das Modell muss lediglich das höchste Penalty-Parameterverhältnis vorherzusagen). Bei den anderen Problemklassen (MECP. KP, BILP) benötigt der Zufallsprozess im Durchschnitt nur 5 Iterationen, um das Modell zu übertreffen. Auffallend ist, dass sich das Konfidenzintervall des Modells in diesen Fällen vollständig mit dem Näherungsverhältnis des Zufallsprozesses überschneidet. Mögliche Gründe dafür sind in den Clustering-Ergebnissen zu finden. Obwohl die Trends der Trajektorien der minimalen Spektrallücke im KP-Clustering ähnlich wie bei MVCP, MCP und SPP sind, sind die vorhergesagten optimalen Penalty-Parameter auf realer Hardware vergleichsweise schlechter (vgl. Abbildung 5.3 und 5.5). Möglicherweise ist dies auf das relativ kleine Intervall der minimalen Spektrallückengröße bei den KP-Probleminstanzen zurückzuführen, das im Bereich von 0.0 bis 0.05 liegt (vgl. Abbildung 5.2d). Es ist davon auszugehen, dass diese kleinen Unterschiede in der Größe der minimalen Spektrallücke keinen Einfluss auf die Ausführung auf der D-Wave Hardware haben. In Bezug auf das MECP ist ein ähnliches Problem zu beobachten. Obwohl das Intervall der MSG um den Faktor 10 größer ist als das des KPs (vgl. Abbildung 5.2a), ist der Unterschied der minimalen Spektrallückengröße zwischen dem schlechtesten und dem besten Penalty-Parameterverhältnis einer Probleminstanz sehr klein (gekennzeichnet durch eine geringe Steigung), was zu der gleichen Annahme wie bei dem KP führt. Da beim BILP-Clustering keine Steigung vorhanden ist, kann hier sowohl theoretisch als auch praktisch keine Optimierung auf realer Hardware erreicht werden.

Da in diesem Experiment das aktuell beste Näherungsverhältnis für den Zufallsprozess beibehalten wird, soll nun die durchschnittliche Leistung des KNN-Modells mit dem Durchschnitt aller 50 zufällig gezogenen Stichproben aus dem iterativen Prozess der 100 Probleminstanzen pro Problemklasse verglichen werden. Als Metrik wird wieder das Näherungsverhältnis zur BKS verwendet. Tabelle 5.2 zeigt die entsprechenden Ergebnisse. Es wird deutlich, dass das Modell im Durchschnitt mindestens 12,9% ± 7,7% und bis zu 167,1% ± 29,8% besser abschneidet als der Zufallsprozess. Unter diesem

Problemklasse	Überlegenheit der KNN-Modelle im Vergleich zum zufälligen Stichprobenverfahren
MCP	$167.1\% \pm 29.8\%$
MVCP	$97.4\% \pm 26.2\%$
KP	$82.8\% \pm 55.7\%$
SPP	$87.3\% \pm 32.9\%$
BILP	$12.9\% \pm 7.7\%$
MECP	$18.4\% \pm 11.2\%$

Blickpunkt betrachtet, sollte das Modell also immer für die hier analysierten Problemklassen verwendet werden.

Tabelle 5.2: Durchschnittliche Überlegenheit der KNN-Modelle in Bezug auf das Näherungsverhältnis gegenüber dem Durchschnitt aller 50 zufälligen Stichproben aus dem iterativen Verfahren über die 100 Probleminstanzen pro Problemklasse.

# 5.5.4 Korrelation der Trajektorien-Cluster mit der Anzahl der globalen Optima und der Dichte der Probleminstanzen

Neben den bereits vorgestellten Experimenten konnte eine weitere interessante Beobachtung in Bezug auf die Anzahl der optimalen Lösungen gemacht werden. Wenn man das zuvor gefundene DBSCAN-Clustering verwendet und die Cluster ohne Steigung und einer minimalen Spektrallückengröße von 0.0 gruppiert (vgl. Abbildung 5.3) und sie mit den Probleminstanzen, die entweder genau eine optimale Lösung oder mehrere haben, korreliert, weist der Pearson-Korrelationskoeffizient im Allgemeinen eine sehr hohe Korrelation auf. Wie in Tabelle 5.3 dargestellt, verfügen MCP, MVCP, SPP und BILP alle eine starke (entweder positive oder negative) Korrelation, wobei MECP mit -0.799 immer noch eine vergleichsweise hohe negative Korrelation aufweist. Nur KP bildet eine Ausnahme durch eine relativ geringere Korrelation mit dem Clustering. Interessant ist, dass genau eine optimale Lösung in gut optimierbare MSG-Trajektorien resultiert, während zwei oder mehrere optimale Lösungen zu flachen MSG-Trajektorien mit MSG-Größen um die 0.0 führen. Dies ist bei allen Problemen mit hohen Korrelationen der Fall. Zudem ist anzumerken, dass die Korrelation des Clusterings mit der genauen Anzahl der optimalen Lösungen (z. B. mit genau 3 optimalen Lösungen) den Pearson-Korrelationskoeffizienten für alle Probleme verringert. Was wiederum zeigt, dass die genaue Anzahl der optimalen Lösungen vergleichsweise eher unbedeutend für die Korrelation ist. Zusätzlich wurde die Dichte der Probleminstanzen untersucht, um zu prüfen, ob die Trajektorien-Cluster mit ihren unterschiedlichen Trends auch mit der Dichte der Probleme korrelieren. Eine geringe Dichte einer Probleminstanz (in diesem Falle der Ising-Matrix) kann folgendermaßen,

$$Dichte = \frac{\#Nicht - Null - Elemente}{\#Elemente}$$
(5.5)

berechnet werden. Obwohl gering besetzte Probleme (also Matrizen mit wenigen Elementen) leichter auf die geringe Konnektivität der D-Wave Hardware abgebildet werden können und daher leichter zu lösen sein sollten, da kleinere (eingebettete) Probleme erzeugt werden, korrelierten die Trajektorien der minimalen Spektrallücke und ihre Trends nicht mit der Dichte der Probleme. Der Pearson-Korrelationskoeffizient lag für jede Problemklasse bei etwa 0.0. Im Allgemeinen enthielt jedes Cluster (mit Aufwärts-, Abwärts- und auch flachen Trends) Probleminstanzen mit unterschiedlicher Dichte.

Problemklasse	PCC	Vereinigte Cluster
MCP	1.0	$\{1, -1\}, \{0\}$
MVCP	0.990	$\{1\}, \{0, 1, 2\}$
KP	0.419	$\{0\}, \{-1\}$
SPP	0.972	$\{0\}, \{1, -1\}$
BILP	-0.958	$\{0, 1, 2, 4, 5, 6\}, \{-1, 3, 8, 7\}$
MECP	-0.799	$\{0, 1, 2\}, \{-1\}$

Tabelle 5.3: Korrelationsergebnisse zwischen den vereinigten DBSCAN-Trajektorien-Clustern, die keine Steigung aufweisen und eine ungefähre minimale Spektrallückengröße von 0.0 haben, vgl. Abbildung 5.3 und den Probleminstanzen, die entweder genau eine oder mehrere globale Lösungen haben. Dargestellt ist der Pearson-Korrelationskoeffizient. Die Zahlen in den geschweiften Klammern der letzten Tabellenspalte stehen für die entsprechenden annotierten Cluster-Labels in Abbildung 5.3.

### 5.5.5 Korrelation skalierter Probleminstanzen mit der minimalen Spektrallücke

Im folgenden werden die Trajektorien der minimalen Spektrallücke untersucht, wenn die Probleminstanzen in Bezug auf die Anzahl der Variablen nach oben und unten skaliert werden. Da bei allen bisherigen Experimenten die Probleminstanzen auf 8 Variablen beschränkt waren (wegen der zeitaufwendigen Berechnung der Eigenspektren), werden nun Instanzen mit 6, 8 und 10 Variablen für die vergleichsweise gut funktionierenden Problemklassen MVCP, MCP und SPP betrachtet.



Abbildung 5.6: DBSCAN-Clustering der minimalen Spektrallücke und ihrer Lage in Bezug auf die Penalty-Parameterverhältnisse. Die Probleminstanzen jeder Problemklasse unterscheiden sich in der Größe von 6 (blau), 8 (rot) und 10 (gelb) Variablen. Um die 2500 Probleminstanzen in jeder Teilabbildung zu visualisieren, wurde für jedes Cluster die Mittelwert-Trajektorie und das 95%-Konfidenzintervall als Fehlerbalken berechnet. Zu erwähnen ist, dass für Probleminstanzen der Größe 6 und 8 1000 Instanzen pro Problemklasse und für die Größe 10 nur 500 Instanzen pro Problemklasse generiert wurden. Abbildung 5.6 zeigt das kombinierte DBSCAN-Clustering unterschiedlich großer Probleminstanzen. Die Cluster gleicher Problemgröße sind identisch eingefärbt. Der erzielte Pearson-Korrelationskoeffizient (mit optimierten DBSCAN-Hyperparametern) beträgt -0.900, -0.992 und -0.901 für MVCP, MCP und SPP. Es ist zu erkennen, dass unabhängig von der Problemgröße die gleichen Trajektorientrends auftreten. Daher ist davon auszugehen, dass die beobachteten Trends größenunabhängig sind und auch für andere Problemgrößen gelten werden.

### 5.5.6 Vergleich der minimalen Spektrallücken von logischen und eingebetteten Instanzen

Bei der Generierung des Datensatzes wurden die minimalen Spektrallücken der logischen bzw. nicht eingebetteten Ising-Hamiltonians berechnet, da eingebettete Probleme meist zusätzliche Variablen bzw. physikalische Qubits beinhalten und eine Berechnung der Eigenspektren erschweren. Daher wird in diesem Abschnitt für eine kleine Teilmenge von Probleminstanzen überprüft, ob sich die Eigenspektren bzw. die minimalen Spektrallücken voneinander unterscheiden. Für die Analyse wurden logische Probleminstanzen mit 6 Variablen des SPP, MCP und MVCP (die Problemklassen, die in den Experimenten vorteilhafte Ergebnisse lieferten) verwendet.

Wie bereits in Abschnitt 5.4.1 erwähnt, wurde die D-Wave Autoskalierungsfunktion verwendet, um die logischen Ising-Koeffizienten auf den Präzisionsbereich der Hardware-Qubits zu skalieren. Daneben kamen die D-Wave Anneal-Funktionen der entsprechenden Hardware zum Einsatz. Der einzige Unterschied zwischen den logischen Ising-Probleminstanzen und den eingebetteten Problemen sind demnach die physikalischen Qubit-Ketten, die auftreten, wenn ein logisches Problem nicht direkt in die Hardware eingebettet werden kann. Um den Unterschied zwischen den logischen und den eingebetteten Instanzen zu analysieren, wurde das tatsächliche Embedding des D-Wave Hardwaregraphen ausgelesen. Die 6 logischen Variablen der Ising-Probleminstanzen führten zu 8 physikalischen Qubits auf der Hardware. Anschließend wurden die minimalen Spektrallücken (für verschiedene Penalty-Parameter) der Ising-Probleme mit nun 8 Variablen berechnet. Hierfür wurde die gleiche Methode wie für die logischen Ising-Probleme verwendet. In Abbildung 5.7 sind die entsprechenden MSG-Trajektorien für 25 Instanzen pro Problemklasse aufgetragen. Die Ergebnisse zeigen, dass es einen Unterschied in den absoluten Werten der minimalen Spektrallücke gibt, jedoch scheinen die Trends der Trajektorien die gleichen zu sein, wie bei den ursprünglichen (nicht eingebetteten) Ising-Probleminstanzen in Abbildung 5.2. Daher ist davon auszugehen, dass die Qubit-Ketten (zumindest bei solch kleinen Problemgrößen) die Verteilung der Eigenwerte des entsprechenden Eigenspektrums nicht verändern.



Abbildung 5.7: Darstellung der Korrelationen zwischen den minimalen Spektrallücken und ihrer Lage im Annealing-Prozess über das Penalty-Parameterverhältnis von 25 zufälligen Instanzen pro Problemklasse. Für jede simulierte eingebettete Probleminstanz (dargestellt durch eine Farbe) wurden 50 gleichmäßig verteilte Penalty-Parameterpaare (dargestellt durch die Punkte) gezogen. Es ist zu beachten, dass sich innerhalb der Teilabbildungen einige Trajektorien überschneiden können.

# 5.6 Kapitelzusammenfassung und Ausblick

In diesem Kapitel wurde die Wahl der Penalty-Parameter verschiedener eingeschränkter Hamiltonians und dessen Einfluss auf die minimale Spektrallücke analysiert. Es wurde gezeigt, dass bestimmte Penalty-Parameterverhältnisse die minimale spektrale Lücke vergrößern können, was sich wiederum in einem verbesserten Näherungsverhältnis auf realer Quantum Computing Hardware niederschlägt. Zudem konnten Regressionsmodelle trainiert werden, die in der Lage waren, die besten geeigneten Penalty-Parameter in dem definierten Versuchsaufbau vorherzusagen. Die Modelle schnitten durchschnittlich um mindestens 12,  $9\% \pm 7, 7\%$  und bis zu 167,  $1\% \pm 29, 8\%$  besser ab, als die des definierten Zufallsprozesses. Dies lässt den Schluss zu, dass derartige trainierte Modelle für die untersuchten Problemklassen verwendet werden sollten, um eine bessere Lösungsqualität für die entsprechenden Hamiltonians auf D-Wave Quantum Annealing Hardware zu erreichen.

Eine weitere Erkenntnis dieser Untersuchung war die hohe Pearson-Korrelation der Anzahl der optimalen Lösungen der Probleminstanzen mit den Clustern, die mit DBSCAN gefunden wurden. Die Ergebnisse zeigten, dass Probleminstanzen mit genau einem globalen Optimum durch ein passendes Penalty-Parameterverhältnis gut optimiert werden können, jedoch solche mit mehreren gleichwertigen optimalen Lösungen hingegen nicht. Außerdem war ersichtlich, dass die Instanzen, die gut optimiert werden können, ihre minimale Spektrallücke früh im Annealing-Prozess haben, verglichen mit denjenigen, die nicht optimiert werden können.

Da in diesem Kapitel ausschließlich eingeschränkte Hamiltonians mit nur einer Nebenbedingung behandelt wurden, sind für zukünftige Arbeiten vor allem Optimierungsprobleme mit mehreren Nebenbedingungen (wie sie in der realen Welt häufiger vorkommen) von Interesse. Darüber hinaus müssen für den Schritt der Datengenerierung effiziente Methoden zur Vorauswahl vielversprechender Penalty-Parameterverhältnisse untersucht werden, da die möglichen Kombinationen im ungünstigsten Fall exponentiell mit der Anzahl der Parameter ansteigen. Ein weiterer interessanter Aspekt für zukünftige Arbeiten wäre auch eine hardwarenahe empirische Untersuchung der Qubit-Präzisionsfähigkeiten der D-Wave Systeme. Da die Experimente gezeigt haben, dass für einige Problemklassen (MVCP, MCP, SPP) das beste Penalty-Parameterverhältnis das Größte war, d.h., wo der Parameter A auf den minimal möglichen Wert in dem definierten Stichprobenintervall gesetzt wurde. Daher wäre es interessant zu sehen, inwiefern sich die Lösungsqualität ändert, bevor Fehler von Quantisierungseffekten der digitalen Analogwandler [45] auftreten und keine Verbesserung in der Lösungsqualität mehr möglich ist.

# 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit stand die Anwendung und Optimierung verschiedener eingeschränkter Hamiltonians im Mittelpunkt. Hamiltonians bilden die grundlegenden Bausteine für annealingbasierte, aber auch viele gatterbasierte Quantenalgorithmen [25]. In dieser Arbeit wurde besonderer Wert auf den praktischen Nutzen bei deren Anwendung auf NISQ-Hardware gelegt. Da aktuelle NISQ-Hardware jedoch noch immer sehr begrenzt in der Anzahl der Qubits, deren Kohärenzeigenschaften und deren Konnektivität auf der QPU ist, ist eine der größten Herausforderungen, relevante Anwendungen und Problemformulierungen mit großem Potenzial für aktuelle Hardware zu identifizieren und entsprechende Algorithmen zu entwickeln, um schon frühzeitig einen möglichen Quantenvorteil zum klassischen Computing zu erzielen. Die in dieser Arbeit untersuchten Problemstellungen und deren Hamiltonians umfassen sowohl akademische Problemstellungen, die die Basis für viele Realwelt-Anwendungsfälle bilden, als auch praxisbezogene hybride Algorithmen für die Domäne der Spieltheorie. Neben der Entwicklung neuer hybrider Algorithmen und Hamiltonians für das Finden von reinen Nash-Gleichgewichten in graphischen Spielen, lag der Schwerpunkt dieser Arbeit auf der Ausarbeitung unterstützender Methoden, um Problemstellungen in Form von eingeschränkten Hamiltonians für aktuelle NISQ-Hardware zu optimieren. Die Arbeit umfasst zahlreiche Experimente, neu entwickelte Algorithmen und empirische Vergleiche für unterschiedliche eingeschränkte Hamiltonians. Dadurch konnte die besondere Bedeutung der Optimierung von Hamiltonians für die Anwendbarkeit auf NISQ-Hardware unterstrichen werden. Im Folgenden werden nochmals die wichtigsten Erkenntnisse und Ergebnisse der einzelnen Kapitel zusammengefasst.

# 6.1 Zusammenfassung

In Kapitel 3 der Arbeit wurden zwei hybride Quantum Computing Ansätze vorgestellt, die zur Berechnung von reinen Nash-Gleichgewichten in graphischen Spielen verwendet werden können. Die Berechnung von reinen Nash-Gleichgewichten in graphbasierten Spielen, bei denen die Knoten und Kanten des Graphen die Spieler und die Abhängigkeiten zwischen den Spielern darstellen, ist NP-vollständig und findet sich in vielen Realwelt-Anwendungsfällen der Wirtschaft wieder. Um einen möglichen Vorteil zu identifizieren, wurden die Ansätze zum einen auf ihre Anwendbarkeit auf aktueller NISQ-Hardware und zum anderen gegen klassische Methoden evaluiert.

Der erste Ansatz basierte auf einer eingeschränkten Hamiltonian-Formulierung, die sowohl auf Annealing- als auch Gatter-Hardwarearchitektur ausgeführt werden kann. Der zweite Ansatz war ein Quantenschaltkreis, der auf dem bekannten Grover-Algorithmus und der Amplitudenverstärkung basiert und für universelle Quantencomputer entwickelt wurde. Es wurde gezeigt, dass der eingeschränkte Hamiltonian Ansatz auf annealingbasierter Hardware zu *State of the Art* Algorithmen konkurrenzfähig ist, während der Grover-Ansatz aufgrund der benötigten Hardwareressourcen aktuell für diese spezielle Domäne nicht NISQ fähig ist.

Die Erkenntnisse aus Kapitel 3 führten im darauffolgenden Kapitel 4 zur Untersuchung von verschiedenen eingeschränkten Ising-Hamiltonians und die Möglichkeit, diese effizient ohne zusätzliche Hardwareressourcen zu optimieren. Es wurde eine algorithmusunabhängige Methode vorgestellt, die mittels der Kreuzentropie die Penalty-Parameter eingeschränkter Hamiltonians optimiert. Die Ergebnisse zeigten eine signifikant verbesserte Lösungsqualität bei der Ausführung auf Quantum Annealing Hardware. Die numerische Berechnung des Eigenspektrums zeigte, dass die Lösungsqualität mit einer größeren minimalen Spektrallücke korreliert, welche als Metrik für die Anwendbarkeit von Hamiltonians auf Quantum Annealing Hardware angesehen werden kann. Das Optimieren der Penalty-Parameter von eingeschränkten Hamiltonians, die ebenfalls mittels variationellen Ansätzen, wie dem QAOA, auf gatterbasierter Hardware ausgeführt werden können, zeigte eine ähnliche Verbesserung in der Lösungsqualität. Es wurde empirisch gezeigt, dass durch Anwendung der Kreuzentropie-Methode, der klassische Optimierer des QAOA bessere Gatterparameter findet, was sich wiederum in einer gesteigerten Lösungsqualität widerspiegelt. Der Nachteil der unterstützenden Kreuzentropie-Methode liegt allerdings in dem Trade-off zwischen Laufzeit und Lösungsqualität. Die Experimente zeigten keine Regelmäßigkeiten in den optimierten Penalty-Parametern auf, sodass für jede Probleminstanz die zeitaufwendige Kreuzentropie-Methode angewendet werden musste.

Im letzten inhaltlichen Kapitel 5 der Arbeit wurde aufgrund der zuvor gewonnen Erkenntnisse, eine umfangreiche Analyse durchgeführt, um geeignete Richtlinien und Muster bezüglich der Wahl der Penalty-Parameter zwischen den Probleminstanzen einer Problemklasse aber auch klassenübergreifend zu erarbeiten bzw. zu identifizieren. Für eine Auswahl an verschiedenen eingeschränkter Hamiltonians wurde ein Datensatz generiert, der die Größe der minimalen Spektrallücke zum Penalty-Parameterverhältnis der Hamiltonians setzt. Mittels Methoden des maschinellen Lernens wurden Trends und wiederkehrende Muster identifiziert, die eine effiziente Gewichtung der Nebenbedingungen ermöglichen. Zudem wurde experimentell gezeigt, dass Regressionsmodelle in Form von künstlichen neuronalen Netzen in der Lage sind, optimale Penalty-Parameter vorherzusagen und somit eine gesteigerte Lösungsqualität der Hamiltonians auf NISQ-Hardware erzielt werden kann. Die vorliegende Arbeit liefert damit verschiedene Methoden und Erkenntnisse in Bezug auf die Optimierung eingeschränkter Hamiltonians und stellt somit einen wichtigen Beitrag dar, entsprechende Problemstellung effizient auf aktueller NISQ-Hardware anzuwenden.

# 6.2 Ausblick

Am Ende eines jeden Kapitels wurden bereits Verbesserungsvorschläge und Ansatzpunkte für zukünftige Arbeiten dargelegt. An dieser Stelle soll deshalb auf offene Fragestellungen und mögliche Erweiterungen eingegangen werden, die auf einen möglichst großen Teil, der in dieser Arbeit vorgestellten Inhalte, zutreffen.

Im Wesentlichen wurde sich in dieser Arbeit auf die Anwendung und Optimierung diskreter kombinatorischer Optimierungsprobleme mit einer Nebenbedingung und deren quadratische Ising-Hamiltonians beschränkt. Eine Erweiterung der in Kapitel 4 vorgestellten Kreuzentropie-Methode oder auch die Untersuchung der Eigenspektren eingeschränkter Hamiltonians in Kapitel 5 mit Problemstellungen, die mehrere Nebenbedingungen besitzen, wäre denkbar und würde zudem eine größere Menge an Realwelt-Anwendungsfällen abbilden. Zu beachten ist, dass durch die Anzahl der Nebenbedingungen und deren Penalty-Parameter, die Berechnung der möglichen Parameterkombinationen im ungünstigsten Fall exponentiell steigt. Daher müssten zudem, vor allem in Bezug auf die Datengenerierung des Kapitels 5, effiziente Methoden für eine Vorselektierung vorteilhafter Kombinationen gefunden werden.

Interessant wäre zudem, kontinuierliche und höhergradige Optimierungsprobleme zu betrachten. Der QAOA kann beispielsweise Hamiltonians in Form von Polynomial Unconstrained Boolean Optimization (PUBO) ausführen. Auch wenn diese meist weniger Qubits für höhergradige Problemstellungen benötigen, als die entsprechende QUBO-Variante, ist mit einem größeren Schaltkreis bei der PUBO-Formulierung zu rechnen [109]. Nichtsdestotrotz wäre eine Untersuchung der Penalty-Parameter höhergradiger eingeschränkter Hamiltonians und deren Auswirkung auf das jeweilige Eigenspektrum interessant. D-Wave Quantum Annealing Hardware akzeptiert zum aktuellen Zeitpunkt nur quadratische Hamiltonians in Form des Ising-Modells oder der QUBO-Formulierung als Eingabe. Hier müsste man vorab höhergradige in quadratische Problemstellungenen transformieren, wie sie bereits in dieser Arbeit untersucht wurden. Diese Transformation ist jedoch mit zusätzlichen Variablen bzw. Qubits verbunden [54, 74]. Grundsätzlich wäre, mit ausreichenden Rechenkapazitäten, die Untersuchung der Penalty-Parameter größerer eingeschränkter Probleminstanzen und deren Einfluss auf das jeweilige Eigenspektrum bedeutsam.

Unabhängig von der Optimierung der Penalty-Parameter eingeschränkter Ha-

miltonians, gibt es weitere Optimierungsmöglichkeiten, Hamiltonians ressourcenschonend zu präparieren. Aufgrund der geringen Anzahl an Qubits und deren Konnektivität auf aktueller NISQ-Hardware zielen viele Forschungsarbeiten darauf ab, zum einen eine Variablenersparnis bei der Ising-Hamiltonian-Formulierung zu erzielen, oder zum anderen deren Abhängigkeiten untereinander zu reduzieren [89, 124]. Mittels dieser Optimierungen würden sich größere und somit relevantere Probleminstanzen auf aktuell spärlich vernetzten Hardwarearchitekturen abbilden und lösen lassen.

# Abbildungsverzeichnis

2.1	Die Bloch-Einheitskugel.	8
2.2	Vereinfachte Darstellung eines beliebigen Eigenspektrums mit vier Zuständen, dessen Grundzustand durch die unterste Linie beschrieben ist und die angeregten Zustände darüber. Der rote Kreis markiert die minimale Spektrallücke zwischen dem Grund- zustand und dem ersten angeregten Zustand über den gesamten Evolutionspfad $t: 0 \rightarrow t_f$ . Die Abbildung basiert auf [47]	16
2.3	Darstellung einer nicht-konvexen Lösungslandschaft eines belie- bigen Optimierungsproblems mit einem globalen Minimum und mehreren lokalen Minima.	18
2.4	Visualisierung eines $3 \times 3$ Chimera-Graphen $C_3$ , in dessen 9 Chimera-Zellen jeweils 8 Qubits als bipartiter Graph angeordnet sind. Abbildung basiert auf [32]	26
2.5	D-Wave 2000Q Systems Annealing Funktionen $\mathcal{A}(t)$ und $\mathcal{B}(t)$ . Der Annealing Prozess beginnt zum Zeitpunkt $t = 0$ mit $\mathcal{A}(t) \gg \mathcal{B}(t)$ und endet zum Zeitpunkt $t = t_f = 1$ mit $\mathcal{A}(t) \ll \mathcal{B}(t)$ . Die Abbildung basiert auf [129].	27
2.6	Quantenschaltkreis mit drei Qubits und drei Gattern. Die hori- zontalen Linien repräsentieren die Qubits und die Rechtecke die Quantengatter, die auf die entsprechenden Qubits wirken. Die Abbildung basiert auf [86].	31
2.7	Darstellung von Operationen und der entsprechenden Gatter in einem Quantenschaltkreis. In Bezug auf das $R_Z(\theta)$ -Gatter beschreibt Z die Rotationsachse und $\theta$ den Rotationsparameter. Analog kann die X- oder Y-Achse als Rotationsachse verwendet werden. Das CNOT- und Toffoli-Gatter besitzen beide ein Ziel- Qubit ( $\oplus$ ) und ein bzw. zwei Kontroll-Qubits ( $\bullet$ ). Das $\sqrt{X}$ - Gatter entspricht dem $R_X(\frac{\pi}{2})$ -Gatter mit einer anderen globalen Phase und ist zudem ein physikalisch implementierbares Gatter	
	der IBM Quantum Hardware	31

2.8 Übersicht des hybriden quanten-klassischen QAOA: Die Quantenevolution startet mit einer gleichverteilten Superposition der Basiszustände  $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$ , die durch die Anwendung von Hadamard-Gattern realisiert wird. Anschließend werden die Operatoren der Problem- und Transversalfeld-Hamiltonians  $H_P$  und  $H_D$  abwechselnd mit den entprechenden Rotationswinkeln  $\beta$  und  $\gamma$  p Runden auf den Quantenzustand  $|\psi\rangle$  angewendet, um den neuen Zustand  $|\psi(\beta,\gamma)\rangle$ zu präparieren. Abschließend werden mehrere Messungen  $\langle \psi(\beta,\gamma)| H_P |\psi(\beta,\gamma) \rangle$  genommen, deren Erwartungswert dann als Input für den klassischen Optimierer dient, um die Winkelparameter  $\beta, \gamma$  anzupassen. Dieses Verfahren wird über mehrere Iterationen ausgeführt, bis der Optimierer konvergiert oder eine Lösung mit ausreichender Qualität erzielt wurde. . . . . . . .

33

Darstellung der Amplitudenverstärung als Balkendiagramm: 2.9Links sind die Amplituden des gleichverteilten Superpositionszustands  $|s\rangle$  als Balken abgebildet. Der blaue Balken repräsentiert die Amplitude des gesuchten Zustands  $|w\rangle$ . Nach Anwendung der Orakelfunktion  $\mathcal{O}$  wird die Amplitude des gesuchten Zustands negiert, was zudem dazu führt, dass die durchschnittliche Amplitude (gekennzeichnet durch die gestrichelte Linie) verringert wird, siehe Abbildung Mitte. Abschließend wird der Diffusionsoperator angewendet, der die Spiegelung der Amplituden um deren Mittelwert verursacht, siehe Abbildung rechts. Da die durchschnittliche Amplitude durch die erste Spiegelung gesenkt wurde, erhöht die zweite Transformation bedingt durch den Diffusionsoperator die negative Amplitude von  $|w\rangle$  auf etwa das Dreifache ihres ursprünglichen Wertes, während sie die anderen Amplituden verringert. Durch mehrmaliges Ausführen dieser beiden Transformationen wird die Wahrscheinlichkeit, den gesuchten Zustand  $|w\rangle$  zu messen, schrittweise erhöht. . . . . . 35

# 2.10 Topologiegraph eines 27 Qubit Quantum Computers von IBM. Die Knoten repräsentieren die Qubits, und die Kanten die physikalischen Verbindungen zwischen den Qubits. Abbildung aus [91]. [91].

- 3.1 Auf der linken Seite ist ein Ausschnitt der Obermenge B gegeben. Auf der rechten Seite sind die Mengen vereinigt, wobei jede Vereinigung (1) und (2) eine globale Strategiekombination darstellt, aber nur (1) ein reines Nash-Gleichgewicht beinhaltet. 51
  3.2 Verschiedene graphische Spieltopologien, die die Abhängigkeiten gwischen den Spielern eines Spiele anzeigen (a) Baumtene.

3.3	Gemittelte Histogramme und angenäherte Wahrscheinlichkeits- verteilungen von fünf Ausführungen mit jeweils 1024 Messungen von Q-Nash auf dem D-Wave 2000Q (QA), QASM Simulator (QAOA) und IBMQ-27 Ehningen (QAOA) für das 9 (A) und 21 (A) Spiel. Für Quantum Annealing wurde die Annealing-Zeit auf $20\mu s$ und für QAOA die Anzahl der alternierenden Opera- toren auf $p = 6$ gesetzt	58
3.4	Q-Nash QAOA-Schaltkreis für $p = 1$ und zwei Qubits unter Berücksichtigung der physikalisch implementierbaren Gattern des IBMQ-27 Ehningen Rechners.	59
3.5	In 3.5b, 3.5a und 3.5c ist die Lösungsqualität von Q-Nash mit QBSolv, IBR und BF auf unterschiedlich großen graphischen Spielinstanzen dargestellt. Die Spiele basieren auf der Baum-, Kreis- und Straßentoplogie. In 3.5d ist der Einfluss des <i>Num</i> <i>Repeats</i> Parameters in Bezug auf das Finden von reinen NE beispielhaft an einem Spiel mit 24 Spielern und Straßentopologie visualisiert.	62
3.6	Überblick über die Laufzeitkomponenten von Q-Nash mit QBSolv.	64
4.1	Beispielhafte Darstellung der KE-Methode für zehn Generatio- nen $(G = 10)$ . Die graue Schattierung repräsentiert die ungül- tigen Bereiche der abgeschnittenen Normalverteilung. Die El-	

4.2	Lösungslandschaften und Lösungsqualitäten der QAOA- Zielfunktion $\min \langle \psi(\beta, \gamma)   H_C   \psi(\beta, \gamma) \rangle$ für die Probleminstanz $\mathcal{A}$ mit $H_C$ als deren Kosten- bzw. Problem-Hamiltonian. Die Probleminstanz $\mathcal{A}$ besteht aus zwei Gegenständen $x_1$ und $x_2$ mit Gewichten $w_1 = 1$ bzw. $w_2 = 1$ . Die Rucksackkapazität $y_1$ beträgt $W = 1$ . Die Wertigkeit der Gegenstände wurden auf $c_1 = 2$ und $c_2 = 1$ gesetzt. Die BKS für die Variablen $ x_1, x_2, y_1\rangle$ ist demnach 101 und in dunkelblau markiert. Die grünen Kreise entsprechen den Parametern, die durch den klassischen Optimierer abgetastet wurden. Das rote Kreuz markiert das globale Minimum der Lösungs- bzw. Energielandschaft, das für die Abbildungen 4.2a und 4.2b bei $-2.85$ bzw. $-3.2$ liegt. In Abbildung 4.2a sieht man vergleichsweise schlechte und zufällig gewählte Penalty-Parameter, während in Abbildung 4.2b, die mittels der KE-Methode optimierten Parameter verwendet wurden. Die Histogramme in Abbildung 4.2c und 4.2d zeigen die entsprechenden Lösungsqualitäten in Form des	
	Näherungsverhältnisses.	85
4.3	Darstellung der Lösungsqualitäten für verschiedene QAOA- Schaltkreistiefen $p = 1, 2, 3$ (4.3a-4.3c). Für jede Schaltkreis- tiefe wurden fünf KP-Instanzen $\mathcal{A}$ - $\mathcal{E}$ verwendet. Die mit zu- fällig gekennzeichneten Boxplots repräsentieren das Näherungs- verhältnis von fünf zufällig gezogenen Penalty-Parameterpaaren (mit jeweils zehn Ausführungen), während die mit optimiert ge- kennzeichneten Boxplots das Näherungsverhältnis des $\rho$ -Anteils der Penalty-Parameterpopulation der 10. Generation des KE- Verfahrens darstellen.	86
4.4	Verschiedene D-Wave Hardware-Emebeddings und ihre Lö- sungsqualität in Form des Näherungsverältnisses zu der BKS für die MECP-Instanz $\mathcal{A}$	89
4.5	Eigenspektren und Lösungsqualitäten für die MECP- Probleminstanz $\mathcal{A}$ . Das Problem $\mathcal{A}$ besteht aus vier Teilmengen mit Zahlen von 1 bis 4, $[(1,2), (1,3), (1,2,4), (3)]$ . Die BKS (dunkelblau markiert) ist 0011, d.h., die Mengen mit Index 2 und 3 sind in der Lösung enthalten. $g_{min} = \delta$ markiert die minimale Spektrallücke, die für zufällige Penalty-Parameter $A$ und $B$ etwa 1 $h \times GHz$ und für optimierte Werte etwa 3 $h \times GHz$ beträgt, siehe Abbildungen 4.5a bzw. 4.5b. Die Abbildungen 4.5c und 4.5d zeigen die Wahrscheinlichkeiten, einen entsprechenden Zustand zu messen	91
4.6	Korrelation des Näherungsverhältnisses zur BKS und der mi- nimalen spektralen Lücke. Die blauen Kreise repräsentieren die Individuen (Penalty-Parameterpaare) über zehn Generatio- nen der KE-Methode (insgesamt 1000 Individuen pro Abbil- dung). Die erste Reihe repräsentiert die KP-Instanzen $\mathcal{A} - \mathcal{D}$ , während die zweite und dritte Reihe die MECP- bzw. SPP- Instanzen darstellen. Die rote Zahl gibt den jeweiligen Pearson- Korrelationskoeffizienten an.	92
-----	---	-----
4.7	Die Lösungsqualitäten für die vier Probleminstanzen $\mathcal{A}$ - $\mathcal{D}$ des KP, MECP und SPP sind jeweils in 4.7a-4.7c dargestellt. Die als <i>zufällig</i> gekennzeichneten Boxplots repräsentieren das Näherungsverhältnis von fünf zufällig ausgewählten Penalty-Parametern (mit jeweils zehn Ausführungen), während die mit <i>optimiert</i> gekennzeichneten Boxplots das Näherungsverhältnis des $\rho$ -Anteils der Penalty-Parameterpopulation der 10. Generation des KE-Verfahrens darstellen.	94
5.1	Links: Konzept eines einfachen Perzeptrons. Rechts: Mehr- schichtiges Perzeptron mit einer Eingabeschicht (rot), einer Aus- gabeschicht (grün), mehreren verdeckten Schichten (orange). Je- des Perzeptron ist mit jedem anderen der benachbarten Schicht verbunden	102
5.2	Visualisierung der Korrelationen zwischen der minimalen Spek- trallücke und ihrer Position im Annealing-Prozess und dem Penalty-Parameterverhältnis für 25 zufällige Instanzen pro Pro- blemklasse. Für jede Probleminstanz (dargestellt durch eine Farbe) wurden 50 gleichmäßig verteilte Penalty-Parameterpaare (dargestellt durch die Punkte) gezogen	107
5.3	Clustering Ergebnisse der minimalen Spektrallücke und ihrer Lage in Bezug auf die Penalty-Parameterverhältnisse. Um die 1000 Probleminstanzen in jeder Teilabbildung zu visualisieren, ist für jedes Cluster die gemittelte Trajektorie und ihr 95%- Konfidenzintervall als Fehlerbalken dargestellt. Jedes Cluster ist zudem eindeutig gefärbt und sein Label in Textform annotiert.	111
5.4	$R^2$ - (a) und $RMSE$ -Koeffizienten (b) während des Trainings inklusive des 95%-Konfidenzintervalls über 10 Durchläufe für die sechs untersuchten Problemklassen. Ein großer $R^2$ und ein kleiner $RMSE$ -Koeffizient sind günstig	113

5.5	Darstellung des durchschnittlichen BKS-Näherungsverhältnis	
	des Modells (rot) in Bezug auf die vorhergesagten Penalty-	
	Parameter gegenüber dem Näherungsverhältnis der zufällig aus-	
	gewählten Penalty-Parameter (grün), einschließlich des 95%-	
	Konfidenzintervalls über 50 Iterationen und jeweils 100 Proble-	
	minstanzen. Zu beachten ist, dass das durchschnittliche Nähe-	
	rungsverhältnis des Modells als Basiswert verwendet und somit	
	auf 1.0 gesetzt wurde, während das Näherungsverhältnis der zu-	
	fällig ausgewählten Penalty-Parameter relativ zu dem des Mo-	
	dells aufgetragen ist. Bezüglich der zufällig gewählten Parame-	
	ter wurde das aktuell bisher beste erzielte Näherungsverhältnis	
	über die Iterationen beibehalten.	. 114
5.6	DBSCAN-Clustering der minimalen Spektrallücke und ihrer La-	
	ge in Bezug auf die Penalty-Parameterverhältnisse. Die Proble-	
	minstanzen jeder Problemklasse unterscheiden sich in der Grö-	
	ße von 6 (blau), 8 (rot) und 10 (gelb) Variablen. Um die 2500	
	Probleminstanzen in jeder Teilabbildung zu visualisieren, wur-	
	de für jedes Cluster die Mittelwert-Trajektorie und das 95%-	
	Konfidenzintervall als Fehlerbalken berechnet.	. 118
5.7	Darstellung der Korrelationen zwischen den minimalen Spek-	
	trallücken und ihrer Lage im Annealing-Prozess über das	
	Penalty-Parameterverhältnis von 25 zufälligen Instanzen pro	
	Problemklasse. Für jede simulierte eingebettete Probleminstanz	
	(dargestellt durch eine Farbe) wurden 50 gleichmäßig verteil-	
	te Penalty-Parameterpaare (dargestellt durch die Punkte) gezo-	

gen. Es ist zu beachten, dass sich innerhalb der Teilabbildungen

einige Trajektorien überschneiden können.

## Literatur

- S. Aaronson und P. Rall. "Quantum approximate counting, simplified". In: Symposium on Simplicity in Algorithms. SIAM. 2020, S. 24–32.
- [2] C. C. Aggarwal, A. Hinneburg und D. A. Keim. "On the Surprising Behavior of Distance Metrics in High Dimensional Space". In: *Database Theory — ICDT 2001*. Hrsg. von J. Van den Bussche und V. Vianu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, S. 420–434. ISBN: 978-3-540-44503-6.
- D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd und O. Regev. "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation". In: SIAM Journal on Computing 37.1 (2007), S. 166–194. DOI: 10.1137/S0097539705447323.
- T. Albash und D. A. Lidar. "Adiabatic quantum computation". In: *Rev. Mod. Phys.* 90 (1 2018), S. 015002. DOI: 10.1103/RevModPhys.90.015 002.
- G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-5 Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Łukasz Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya und C. Zoufal. Qiskit: An Open-source Framework for Quantum Computing. Version 0.7.2. Jan. 2019. DOI: 10.5281/zenodo.2562111.

- [6] P. Amara, D. Hsu und J. E. Straub. "Global energy minimum searches using an approximate solution of the imaginary time Schroedinger equation". In: *The Journal of Physical Chemistry* 97.25 (1993), S. 6715– 6721. DOI: 10.1021/j100127a023.
- [7] B. Apolloni, N. Cesa-Bianchi und D. De Falco. "A numerical implementation of "quantum annealing"". In: Stochastic Processes, Physics and Geometry: Proceedings of the Ascona-Locarno Conference. 1990, S. 97– 111.
- [8] R. Babbush, B. O'Gorman und A. Aspuru-Guzik. "Resource efficient gadgets for compiling adiabatic quantum optimization problems". In: *Annalen der Physik* 525.10-11 (2013), S. 877–888. DOI: 10.1002/andp .201300120.
- [9] J.-H. Bae, P. M. Alsing, D. Ahn und W. A. Miller. "Quantum circuit optimization using quantum Karnaugh map". In: *Scientific reports* 10.1 (2020), S. 1–8. DOI: 10.1038/s41598-020-72469-7.
- [10] F Barahona. "On the computational complexity of Ising spin glass models". In: Journal of Physics A: Mathematical and General 15.10 (Okt. 1982), S. 3241. DOI: 10.1088/0305-4470/15/10/028.
- [11] B. Barak, A. Moitra, R. O'Donnell, P. Raghavendra, O. Regev, D. Steurer, L. Trevisan, A. Vijayaraghavan, D. Witmer und J. Wright. "Beating the Random Assignment on Constraint Satisfaction Problems of Bounded Degree". In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. Bd. 40. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, S. 110–123. DOI: 10 .4230/LIPIcs.APPROX-RANDOM.2015.110.
- [12] W. P. Baritompa, D. W. Bulger und G. R. Wood. "Grover's Quantum Algorithm Applied to Global Optimization". In: SIAM Journal on Optimization 15.4 (2005), S. 1170–1184. DOI: 10.1137/040605072.
- P. Batra, M. H. Ram und T. Mahesh. "Recommender system expedited quantum control optimization". In: *Physics Open* 14 (2023), S. 100127. ISSN: 2666-0326. DOI: 10.1016/j.physo.2022.100127.
- P. Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (Mai 1980), S. 563–591. ISSN: 1572-9613. DOI: 10.1007/BF01011339.
- [15] C. H. Bennett, E. Bernstein, G. Brassard und U. Vazirani. "Strengths and Weaknesses of Quantum Computing". In: SIAM Journal on Computing 26.5 (1997), S. 1510–1523. DOI: 10.1137/S0097539796300933.
- [16] S. Berninghaus, K. Ehrhart und W. Güth. Strategische Spiele: Eine Einführung in die Spieltheorie. Springer-Lehrbuch. Springer Berlin, Heidelberg, 2010. ISBN: 9783642116513. DOI: 10.1007/978-3-642-11651-3.

- [17] N. A. R. Bhat und K. Leyton-Brown. "Computing Nash Equilibria of Action-Graph Games". In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. UAI '04. Arlington, Virginia, USA: AUAI Press, 2004, 35–42. ISBN: 0974903906.
- [18] J. D. Biamonte. "Nonperturbative k-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins". In: *Phys. Rev. A* 77 (5 Mai 2008), S. 052331. DOI: 10.1103/Ph ysRevA.77.052331.
- [19] M. Bierlaire. Optimization: Principles and Algorithms. 2nd. Lausanne: EPFL Press, 2018. ISBN: 9782940222780.
- [20] M. Booth, S. P. Reinhardt und A. Roy. Partitioning Optimization Problems for Hybrid Classical Quantum Execution. Techn. Ber. D-Wave Systems Inc., 2017. URL: https://docs.ocean.dwavesys.com/proje cts/qbsolv/en/latest/\_downloads/bd15a2d8f32e587e9e5997ce9d 5512cc/qbsolv\_techReport.pdf (besucht am 30.01.2023).
- [21] H. Borchani, G. Varando, C. Bielza und P. Larrañaga. "A survey on multi-output regression". In: WIREs Data Mining and Knowledge Discovery 5.5 (2015), S. 216–233. DOI: 10.1002/widm.1157.
- [22] M. Born und V. Fock. "Beweis des adiabatensatzes". In: Zeitschrift für Physik 51.3 (1928), S. 165–180. DOI: 10.1007/BF01343193.
- [23] E. Boros, P. L. Hammer und G. Tavares. "Local Search Heuristics for Quadratic Unconstrained Binary Optimization (QUBO)". In: *Journal* of Heuristics 13.2 (Apr. 2007), 99–132. ISSN: 1381-1231. DOI: 10.1007 /s10732-007-9009-3.
- [24] M. Boyer, G. Brassard, P. Høyer und A. Tapp. "Tight bounds on quantum searching". In: Fortschritte der Physik: Progress of Physics 46.4-5 (1998), S. 493–505.
- [25] L. T. Brady, C. L. Baldwin, A. Bapat, Y. Kharkov und A. V. Gorshkov. "Optimal Protocols in Quantum Annealing and Quantum Approximate Optimization Algorithm Problems". In: *Phys. Rev. Lett.* 126 (7 Feb. 2021), S. 070505. DOI: 10.1103/PhysRevLett.126.070505.
- [26] G. Brassard, P. Hoyer, M. Mosca und A. Tapp. "Quantum amplitude amplification and estimation". In: *Contemporary Mathematics* 305 (2002), S. 53–74.
- [27] J. Brown, M. Paternostro und A. Ferraro. "Optimal quantum control via genetic algorithms for quantum state engineering in driven-resonator mediated networks". In: *Quantum Science and Technology* 8.2 (Jan. 2023), S. 025004. DOI: 10.1088/2058-9565/acb2f2.

- M. Brusco, C. P. Davis-Stober und D. Steinley. "Ising formulations of some graph-theoretic problems in psychological research: Models and methods". In: *Journal of Mathematical Psychology* 102 (2021), S. 102536. ISSN: 0022-2496. DOI: 10.1016/j.jmp.2021.102536.
- [29] J. Cai, W. G. Macready und A. Roy. "A practical heuristic for finding graph minors". In: *arXiv preprint arXiv:1406.2741* (2014).
- [30] J. Cai, W. G. Macready und A. Roy. "A practical heuristic for finding graph minors". In: *arXiv preprint arXiv:1406.2741* (2014).
- [31] E. Campbell, A. Khurana und A. Montanaro. "Applying quantum algorithms to constraint satisfaction problems". In: *Quantum* 3 (Juli 2019), S. 167. ISSN: 2521-327X. DOI: 10.22331/q-2019-07-18-167.
- Y. Cao, S. Jiang, D. Perouli und S. Kais. "Solving Set Cover with Pairs Problem using Quantum Annealing". In: *Scientific Reports* 6 (2016). DOI: 10.1038/srep33957.
- [33] D. Carfi und F. Musolino. "Fair redistribution in financial markets: a game theory complete analysis". In: *Journal of Advanced Studies in Finance* 2.2 (2011), S. 74–100. ISSN: 2068-8393.
- [34] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio et al. "Variational quantum algorithms". In: *Nature Reviews Physics* 3.9 (2021), S. 625–644. DOI: 10.1038/s42254-021-00348-9.
- [35] N. J. Cerf, L. K. Grover und C. P. Williams. "Nested quantum search and NP-hard problems". In: Applicable Algebra in Engineering, Communication and Computing 10.4 (2000), S. 311–338. DOI: 10.1007/s0 02000050134.
- [36] G. Chapuis, H. Djidjev, G. Hahn und G. Rizk. "Finding Maximum Cliques on the D-Wave Quantum Annealer". In: Journal of Signal Processing Systems 91.3 (März 2019), S. 363–377. ISSN: 1939-8115. DOI: 10.1007/s11265-018-1357-8.
- [37] V. Choi. "The Effects of the Problem Hamiltonian Parameters on the Minimum Spectral Gap in Adiabatic Quantum Optimization". In: *Quantum Information Processing* 19.3 (Jan. 2020). ISSN: 1570-0755. DOI: 10.1007/s11128-020-2582-1.
- [38] R. Cleve und J. Watrous. "Fast parallel circuits for the quantum Fourier transform". In: Proceedings 41st Annual Symposium on Foundations of Computer Science. 2000, S. 526–536. DOI: 10.1109/SFCS.2000.89214
  0.
- [39] M. W. Coffey. "Adiabatic quantum computing solution of the knapsack problem". In: *arXiv preprint arXiv:1701.05584* (2017).

- [40] G. Cornuéjols, J. Peña und R. Tütüncü. Optimization Methods in Finance. 2. Aufl. Cambridge University Press, 2018. DOI: 10.1017/9781 107297340.
- [41] G. E. Crooks. "Performance of the quantum approximate optimization algorithm on the maximum cut problem". In: arXiv preprint ar-Xiv:1811.08419 (2018).
- [42] W. Cruz-Santos, S. E. Venegas-Andraca und M. Lanzagorta. "A QUBO Formulation of Minimum Multicut Problem Instances in Trees for D-Wave Quantum Annealers". In: *Scientific Reports* 9.1 (Nov. 2019). DOI: 10.1038/s41598-019-53585-5.
- [43] D-Wave Systems Inc. Error Sources for Problem Representation. URL: https://docs.dwavesys.com/docs/latest/c\_qpu\_ice.html#qpu-i ce-error-h-scale-interqubits (besucht am 30.01.2023).
- [44] D-Wave Systems Inc. Programming the D-Wave QPU: Setting the Chain Strength. 2020. URL: https://www.dwavesys.com/media/vsufwv1 d/14-1041a-a\_setting\_the\_chain\_strength.pdf (besucht am 30.01.2023).
- [45] D-Wave Systems Inc. QPU-Specific Characteristics. 2020. URL: https: //docs.dwavesys.com/docs/latest/doc\_physical\_properties.ht ml (besucht am 30.01.2023).
- [46] D-Wave Systems Inc. QPU-Specific Characteristics. 2020. URL: https: //docs.dwavesys.com/docs/latest/c\_solver\_parameters.html#a uto-scale (besucht am 30.01.2023).
- [47] D-Wave Systems Inc. What is Quantum Annealing? 2020. URL: https ://docs.dwavesys.com/docs/latest/c\_gs\_2.html#annealing-inlow-energy-states (besucht am 30.01.2023).
- [48] D-Wave Systems sells its first Quantum Computing System to Lockheed Martin Corporation. URL: https://www.dwavesys.com/news/d-wave -systems-sells-its-first-quantum-computing-system-lockheed -martin-corporation (besucht am 30.01.2023).
- [49] W. van Dam, M. Mosca und U. Vazirani. "How powerful is adiabatic quantum computation?" In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science. 2001, S. 279–287. DOI: 10.1109/SF CS.2001.959902.
- [50] W. van Dam, M. Mosca und U. Vazirani. "How powerful is adiabatic quantum computation?" In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science. 2001, S. 279–287. DOI: 10.1109/SF CS.2001.959902.

- [51] C. Daskalakis und C. H. Papadimitriou. "Computing Pure Nash Equilibria in Graphical Games via Markov Random Fields". In: *Proceedings of* the 7th ACM Conference on Electronic Commerce. EC '06. New York, NY, USA: Association for Computing Machinery, 2006, 91–99. ISBN: 1595932364. DOI: 10.1145/1134707.1134718.
- [52] E. R. David, E. H. Geoffrey und J. W. Ronald. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), S. 533– 536. DOI: 10.1038/323533a0.
- [53] M. J. Dinneen, A. Mahasinghe und K. Liu. "Finding the Chromatic Sums of Graphs Using a D-Wave Quantum Computer". In: *The Journal* of Supercomputing 75.8 (Aug. 2019), 4811–4828. ISSN: 0920-8542. DOI: 10.1007/s11227-019-02761-5.
- [54] K. Domino, A. Kundu, Ö. Salehi und K. Krawiec. "Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing". In: *Quantum Information Processing* 21.9 (Sep. 2022), S. 337. ISSN: 1573-1332. DOI: 10.1007/s11128-022-0367 0-y.
- [55] R. Drechsler. "Pseudo-Kronecker expressions for symmetric functions". In: *IEEE Transactions on Computers* 48.9 (1999), S. 987–990. DOI: 10.1109/12.795226.
- [56] D. J. Egger, J. Mareček und S. Woerner. "Warm-starting quantum optimization". In: *Quantum* 5 (Juni 2021), S. 479. ISSN: 2521-327X. DOI: 10.22331/q-2021-06-17-479.
- [57] E. Elkind, L. A. Goldberg und P. Goldberg. "Nash Equilibria in Graphical Games on Trees Revisited". In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. EC '06. New York, NY, USA: Association for Computing Machinery, 2006, 100–109. ISBN: 1595932364. DOI: 10.1145/1134707.1134719.
- [58] M. Ester, H.-P. Kriegel, J. Sander und X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD'96. AAAI Press, 1996, 226–231.
- [59] E. Farhi, J. Goldston, D. Gosset, S. Gutmann, H. B. Meyer und P. Shor. "Quantum Adiabatic Algorithms, Small Gaps, and Different Paths". In: *Quantum Info. Comput.* 11.3 (März 2011), 181–214. ISSN: 1533-7146.
- [60] E. Farhi, J. Goldstone und S. Gutmann. "A quantum approximate optimization algorithm". In: arXiv preprint arXiv:1411.4028 (2014).
- [61] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Mauerer und C. Linnhoff-Popien. "A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer". In: *Frontiers* in ICT 6 (2019). ISSN: 2297-198X. DOI: 10.3389/fict.2019.00013.

- [62] D. Fernandes und I. Dutra. "Using Grover's Search Quantum Algorithm to Solve Boolean Satisfiability Problems: Part I". In: XRDS 26.1 (Sep. 2019), 64–66. ISSN: 1528-4972. DOI: 10.1145/3358251.
- [63] R. P. Feynman. "Simulating physics with computers". In: International Journal of Theoretical Physics 21.6 (Juni 1982), S. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179.
- [64] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath und C. Monroe. "Complete 3-qubit Grover search on a programmable quantum computer". In: *Nature communications* 8.1 (2017), S. 1–9. DOI: 10.103 8/s41467-017-01904-7.
- [65] A. Finnila, M. Gomez, C. Sebenik, C. Stenson und J. Doll. "Quantum annealing: A new method for minimizing multidimensional functions". In: *Chemical Physics Letters* 219.5 (1994), S. 343–348. ISSN: 0009-2614. DOI: 10.1016/0009-2614(94)00117-0.
- [66] T. Fösel, M. Y. Niu, F. Marquardt und L. Li. "Quantum circuit optimization with deep reinforcement learning". In: arXiv preprint ar-Xiv:2103.07585 (2021).
- [67] E. Fredkin und T. Toffoli. "Conservative logic". In: International Journal of theoretical physics 21.3 (1982), S. 219–253. DOI: 10.1007/BF018 57727.
- [68] D. Fudenberg und J. Tirole. *Game Theory*. Translated into Chinesse by Renin University Press, Bejing: China. Cambridge, MA: MIT Press, 1991.
- [69] T. Gabor, M. Zorn und C. Linnhoff-Popien. "The Applicability of Reinforcement Learning for the Automatic Generation of State Preparation Circuits". In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, 2196–2204. ISBN: 9781450392686. DOI: 10.1145/3520304.3534039.
- [70] M. R. Garey und D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. USA: W. H. Freeman & Co., 1979. ISBN: 0716710447.
- [71] A. Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. Ö'Reilly Media, Inc.", 2022. ISBN: 9781491962299.
- [72] E. Gibney. "Inside Microsoft's quest for a topological quantum computer". In: *Nature* (2016). DOI: 10.1038/nature.2016.20774.
- [73] C. Gidney und M. Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". In: *Quantum* 5 (Apr. 2021), S. 433. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-15-433.

- [74] A. Glos, A. Krawiec und Z. Zimborás. "Space-efficient binary optimization for variational quantum computing". In: *npj Quantum Information* 8.1 (Apr. 2022), S. 39. ISSN: 2056-6387. DOI: 10.1038/s41534-022-00 546-y.
- [75] F. Glover, G. Kochenberger und Y. Du. "Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models". In: 4OR 17.4 (Dez. 2019), S. 335–371. DOI: 10.1007/s10288-019-00424-.
- [76] G. Gottlob, G. Greco und F. Scarcello. "Pure Nash Equilibria: Hard and Easy Games". In: J. Artif. Int. Res. 24.1 (Sep. 2005), 357–406. ISSN: 1076-9757.
- [77] E. Grant, L. Wossnig, M. Ostaszewski und M. Benedetti. "An initialization strategy for addressing barren plateaus in parametrized quantum circuits". In: *Quantum* 3 (Dez. 2019), S. 214. ISSN: 2521-327X. DOI: 10.22331/q-2019-12-09-214.
- [78] C. Grimme und J. Bossek. Einführung in die Optimierung. Springer, 2018. ISBN: 978-3-658-21150-9.
- [79] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, 212–219. ISBN: 0897917855. DOI: 10.1 145/237814.237866.
- [80] A. Hagar und M. Cuffaro. "Quantum Computing". In: *The Stanford Encyclopedia of Philosophy*. Hrsg. von E. N. Zalta und U. Nodelman. Fall 2022. Metaphysics Research Lab, Stanford University, 2022.
- [81] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas und H. S. Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit". In: *nature* 405.6789 (2000), S. 947–951. DOI: 10.1038/35016072.
- [82] T. D. Hansen, H. Kaplan, O. Zamir und U. Zwick. "Faster K-SAT Algorithms Using Biased-PPSZ". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. New York, NY, USA: Association for Computing Machinery, 2019, 578–589. ISBN: 9781450367059. DOI: 10.1145/3313276.3316359.
- [83] P. R. Hegde, G. Passarelli, A. Scocco und P. Lucignano. "Genetic optimization of quantum annealing". In: *Phys. Rev. A* 105 (1 Jan. 2022), S. 012612. DOI: 10.1103/PhysRevA.105.012612.
- [84] R. Herrman, L. Treffert, J. Ostrowski, P. C. Lotshaw, T. S. Humble und G. Siopsis. "Globally Optimizing QAOA Circuit Depth for Constrained Optimization Problems". In: *Algorithms* 14.10 (2021). ISSN: 1999-4893. DOI: 10.3390/a14100294.

- [85] M. J. Holler, G. Illing und S. Napel. "Zur Geschichte der Spieltheorie". In: *Einführung in die Spieltheorie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 403–441. ISBN: 978-3-642-31963-1. DOI: 10.1007/9 78-3-642-31963-1\_9.
- [86] M. Homeister. Quantum Computing verstehen. Springer, 2008. DOI: 10 .1007/978-3-8348-9241-6.
- [87] K. Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), S. 251–257. ISSN: 0893-6080.
  DOI: 10.1016/0893-6080(91)90009-T.
- [88] K. Hornik, M. Stinchcombe und H. White. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". In: *Neural Networks* 3.5 (1990), S. 551–560. ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90005-6.
- [89] R. Hua und M. J. Dinneen. "Improved QUBO Formulation of the Graph Isomorphism Problem". In: SN Computer Science 1.1 (Sep. 2019), S. 19. ISSN: 2661-8907. DOI: 10.1007/s42979-019-0020-1.
- [90] IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation. URL: https://www-03.ibm.com/press/us/en/pressrele ase/49661.wss (besucht am 30.01.2023).
- [91] IBM Quantum (2022). IBM Quantum Compute Resources ibmq\_kolkata. URL: https://quantum-computing.ibm.com/servi ces/resources?tab=systems (besucht am 30.01.2023).
- [92] A. J., A. Adedoyin, J. Ambrosiano, P. Anisimov, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, N. Lemons, S. Lin, A. Malyzhenkov, D. Mascarenas, S. Mniszewski, B. Nadiga, D. O'malley, D. Oyen, S. Pakin, L. Prasad, R. Roberts, P. Romero, N. Santhi, N. Sinitsyn, P. J. Swart, J. G. Wendelberger, B. Yoon, R. Zamora, W. Zhu, S. Eidenbenz, A. Bärtschi, P. J. Coles, M. Vuffray und A. Y. Lokhov. "Quantum Algorithm & Implementations for Beginners". In: 3.4 (Juli 2022). ISSN: 2643-6809. DOI: 10.1145/3517340.
- [93] A. X. Jiang und K. Leyton-Brown. "Computing Pure Nash Equilibria in Symmetric Action Graph Games". In: Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1. AAAI'07. AAAI Press, 2007, 79–85. ISBN: 9781577353232.
- [94] J. Jiang, X. Sun, S.-H. Teng, B. Wu, K. Wu und J. Zhang. "Optimal Space-Depth Trade-Off of CNOT Circuits in Quantum Logic Synthesis". In: Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA), S. 213–229. DOI: 10.1137/1.9781611975994.13.

- [95] M. Jiang. "Finding Pure Nash Equilibrium of Graphical Game Via Constraints Satisfaction Approach". In: Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. Hrsg. von B. Chen, M. Paterson und G. Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 483–494. ISBN: 978-3-540-74450-4.
- [96] S. G. Johnson. The NLopt nonlinear-optimization package. 2014. URL: http://github.com/stevengj/nlopt.
- [97] S. Johri, S. Debnath, A. Mocherla, A. Singk, A. Prakash, J. Kim und I. Kerenidis. "Nearest centroid classification on a trapped ion quantum computer". In: *npj Quantum Information* 7.1 (2021), S. 1–11. DOI: 10 .1038/s41534-021-00456-5.
- [98] T. Kadowaki und H. Nishimori. "Quantum annealing in the transverse Ising model". In: *Phys. Rev. E* 58 (5 Nov. 1998), S. 5355–5363. DOI: 10.1103/PhysRevE.58.5355.
- [99] M. J. Kearns, M. L. Littman und S. P. Singh. "Graphical Models for Game Theory". In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. UAI '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, 253–260. ISBN: 1558608001.
- [100] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy und C. Weedbrook. "Strawberry fields: A software platform for photonic quantum computing". In: *Quantum* 3 (2019), S. 129. DOI: 10.22331/q-2019-03 -11-129.
- [101] D. P. Kingma und J. Ba. "Adam: A Method for Stochastic Optimization". In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. Hrsg. von Y. Bengio und Y. LeCun. 2015. DOI: 10.48550 /arxiv.1412.6980.
- [102] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang und Y. Wang. "The Unconstrained Binary Quadratic Programming Problem: A Survey". In: J. Comb. Optim. 28.1 (Juli 2014), 58–81. ISSN: 1382-6905. DOI: 10.1007/s10878-014-9734-0.
- [103] D. Koller und B. Milch. "Multi-Agent Influence Diagrams for Representing and Solving Games". In: Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, 1027–1034. ISBN: 1558608125.
- [104] A. S. Koshikawa, M. Ohzeki, T. Kadowaki und K. Tanaka. "Benchmark test of black-box optimization using D-wave quantum annealer". In: *Journal of the Physical Society of Japan* 90.6 (2021), S. 064001. DOI: 10.7566/JPSJ.90.064001.

- [105] F. Leymann und J. Barzen. "The bitter truth about gate-based quantum algorithms in the NISQ era". In: *Quantum Science and Technology* 5.4 (2020), S. 044007. DOI: 10.1088/2058-9565/abae7d.
- [106] F. Leymann und J. Barzen. "The bitter truth about gate-based quantum algorithms in the NISQ era". In: *Quantum Science and Technology* 5.4 (Sep. 2020), S. 044007. DOI: 10.1088/2058-9565/abae7d.
- [107] M. L. Littman, M. Kearns und S. Singh. "An Efficient, Exact Algorithm for Solving Tree-Structured Graphical Games". In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. NIPS'01. Cambridge, MA, USA: MIT Press, 2001, 817–823.
- [108] A. Lucas. "Ising formulations of many NP problems". In: Frontiers in Physics 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005.
- [109] A. Mandal, A. Roy, S. Upadhyay und H. Ushijima-Mwesigwa. "Compressed Quadratization of Higher Order Binary Optimization Problems". In: Proceedings of the 17th ACM International Conference on Computing Frontiers. CF '20. New York, NY, USA: Association for Computing Machinery, 2020, 126–131. ISBN: 9781450379564. DOI: 10.1 145/3387902.3392627.
- [110] A. Mandviwalla, K. Ohshiro und B. Ji. "Implementing Grover's Algorithm on the IBM Quantum Computers". In: 2018 IEEE International Conference on Big Data (Big Data). 2018, S. 2531–2537. DOI: 10.1109 /BigData.2018.8622457.
- [111] Y. Manin. "Computable and uncomputable". In: Sovetskoye Radio, Moscow 128 (1980).
- [112] C. C. McGeoch. "Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice". In: Synthesis Lectures on Quantum Computing 5.2 (2014), S. 1–93. DOI: 10.2200/S00585ED1V01Y201407QMC00 8.
- C. Moore und M. Nilsson. "Parallel Quantum Computation and Quantum Codes". In: SIAM Journal on Computing 31.3 (2001), S. 799–815.
  DOI: 10.1137/S0097539799355053.
- [114] C. Moussa, J. N. van Rijn, T. Bäck und V. Dunjko. "Hyperparameter Importance of Quantum Neural Networks Across Small Datasets". In: arXiv preprint arXiv:2206.09992 (2022).
- S. Mukherjee. "A Grover Search-Based Algorithm for the List Coloring Problem". In: *IEEE Transactions on Quantum Engineering* 3 (2022), S. 1–8. DOI: 10.1109/TQE.2022.3151137.
- [116] P. L. Mura. "Game Networks". In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. UAI '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, 335–342. ISBN: 1558607099.

- [117] K. G. Murty. Operations Research: Deterministic Optimization Models. USA: Prentice-Hall, Inc., 1994. ISBN: 0130565172.
- J. F. Nash Jr. "Equilibrium points in n-person games". In: Proceedings of the national academy of sciences 36.1 (1950), S. 48-49. DOI: 10.107 3/pnas.36.1.48.
- [119] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni und B. Parney. "Traffic Flow Optimization Using a Quantum Annealer". In: *Frontiers in ICT* 4 (2017). ISSN: 2297-198X. DOI: 10.3389/fict.2017 .00029.
- [120] J. von Neumann und O. Morgenstern. Theory of Games and Economic Behavior. Princeton: Princeton University Press, 2007. ISBN: 9781400829460. DOI: 10.1515/9781400829460.
- [121] T. Nicholson. Optimization in Industry: Volume 1, Optimization Techniques. Routledge, 2017. DOI: 10.4324/9781315125824.
- [122] M. A. Nielsen und I. L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 2010. DOI: 10.1017/CB09780511976667.
- [123] M. Y. Niu, S. Boixo, V. N. Smelyanskiy und H. Neven. "Universal quantum control through deep reinforcement learning". In: *npj Quantum Information* 5.1 (Apr. 2019), S. 33. ISSN: 2056-6387. DOI: 10.1038/s415 34-019-0141-3.
- [124] J. Nüßlein, T. Gabor, C. Linnhoff-Popien und S. Feld. "Algorithmic QUBO Formulations for K-SAT and Hamiltonian Cycles". In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, 2240–2246. ISBN: 9781450392686. DOI: 10.1145/352 0304.3533952.
- [125] L. E. Ortiz und M. Kearns. "Nash Propagation for Loopy Graphical Games". In: Proceedings of the 15th International Conference on Neural Information Processing Systems. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, 817–824.
- [126] M. J. Osborne und A. Rubinstein. A course in game theory. MIT press, 1994. ISBN: 9780262650403.
- [127] A. Palmieri und A. Lallouet. "Constraint Games revisited". In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. 2017, S. 729–735. DOI: 10.24963/ijcai.2017 /101.
- [128] I. Parmee und P. Hajela. Optimization in Industry. Springer London, Jan. 2002. DOI: 10.1007/978-1-4471-0675-3.

- [129] G. Passarelli, V. Cataudella und P. Lucignano. "Improving quantum annealing of the ferromagnetic *p*-spin model through pausing". In: *Phys. Rev. B* 100 (2 Juli 2019), S. 024302. DOI: 10.1103/PhysRevB.100.02 4302.
- [130] E. Pelofske, G. Hahn und H. N. Djidjev. "Advanced anneal paths for improved quantum annealing". In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). 2020, S. 256–266. DOI: 10.1109/QCE49297.2020.00040.
- [131] B. Poggel, N. Quetschlich, L. Burgholzer, R. Wille und J. M. Lorenz. "Recommending Solution Paths for Solving Optimization Problems with Quantum Computing". In: arXiv preprint arXiv:2212.11127 (2022).
- [132] R. Porotti, A. Essig, B. Huard und F. Marquardt. "Deep Reinforcement Learning for Quantum State Preparation with Weak Nonlinear Measurements". In: *Quantum* 6 (Juni 2022), S. 747. ISSN: 2521-327X. DOI: 10.22331/q-2022-06-28-747.
- [133] M. J. D. Powell. "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. Hrsg. von S. Gomez und J.-P. Hennart. Dordrecht: Springer Netherlands, 1994, S. 51–67. ISBN: 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5\_4.
- [134] "Quantum stochastic optimization". In: Stochastic Processes and their Applications 33.2 (1989), S. 233-244. ISSN: 0304-4149. DOI: 10.1016/0 304-4149(89)90040-9.
- [135] R. A. Quintero und L. F. Zuluaga. Characterizing and Benchmarking QUBO Reformulations of the Knapsack Problem. Techn. Ber. Technical Report. Department of Industrial und Systems Engineering, Lehigh University, 2021.
- M. Rabin. "Incorporating Fairness into Game Theory and Economics". In: *The American Economic Review* 83.5 (1993), S. 1281–1302. ISSN: 00028282.
- [137] R. L. Rardin und R. L. Rardin. Optimization in operations research.
  Bd. 166. Prentice Hall Upper Saddle River, NJ, 1998. ISBN: 0023984155.
- [138] B. W. Reichardt. "The Quantum Adiabatic Optimization Algorithm and Local Minima". In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing. STOC '04. New York, NY, USA: Association for Computing Machinery, 2004, 502–510. ISBN: 1581138520. DOI: 10.1145/1007352.1007428.
- [139] E. Rieffel und W. Polak. Quantum Computing: A Gentle Introduction. 1st. The MIT Press, 2011. ISBN: 9780262015066.

- [140] C. Roch, A. Impertro und C. Linnhoff-Popien. "Cross Entropy Optimization of Constrained Problem Hamiltonians for Quantum Annealing". In: *Computational Science – ICCS 2021*. Cham: Springer International Publishing, 2021, S. 60–73. ISBN: 978-3-030-77980-1. DOI: 10.1007/97 8-3-030-77980-1\_5.
- [141] C. Roch, A. Impertro, T. Phan, T. Gabor, S. Feld und C. Linnhoff-Popien. "Cross Entropy Hyperparameter Optimization for Constrained Problem Hamiltonians Applied to QAOA". In: 2020 International Conference on Rebooting Computing (ICRC). 2020, S. 50–57. DOI: 10.110 9/ICRC2020.2020.00009.
- [142] C. Roch, S. Londoño Castillo und C. Linnhoff-Popien. "A Grover based Quantum Algorithm for Finding Pure Nash Equilibria in Graphical Games". In: 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C). 2022, S. 147–151. DOI: 10.1109/ICSA-C54293.2022.00036.
- [143] C. Roch, T. Phan, S. Feld, R. Müller, T. Gabor, C. Hahn und C. Linnhoff-Popien. "A Quantum Annealing Algorithm for Finding Pure Nash Equilibria in Graphical Games". In: *Computational Science – ICCS 2020.* Cham: Springer International Publishing, 2020, S. 488–501. ISBN: 978-3-030-50433-5. DOI: 10.1007/978-3-030-50433-5\_38.
- [144] C. Roch, D. Ratke, J. Nüßlein, T. Gabor und S. Feld. "The Effect of Penalty Factors of Constrained Hamiltonians on the Eigenspectrum in Quantum Annealing". In: ACM Transactions on Quantum Computing (Dez. 2022). ISSN: 2643-6809. DOI: 10.1145/3577202.
- M. Roetteler, M. Naehrig, K. M. Svore und K. Lauter. "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms". In: Advances in Cryptology – ASIACRYPT 2017. Hrsg. von T. Takagi und T. Peyrin. Cham: Springer International Publishing, 2017, S. 241–270. ISBN: 978-3-319-70697-9.
- [146] F. Rosenblatt. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
- [147] T. Roughgarden. Twenty Lectures on Algorithmic Game Theory. Cambridge University Press, 2016. DOI: 10.1017/CB09781316779309.
- [148] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya und Q. Wu. "A Survey of Game Theory as Applied to Network Security". In: 2010 43rd Hawaii International Conference on System Sciences. 2010, S. 1–10. DOI: 10.1109/HICSS.2010.35.
- [149] R. Y. Rubinstein und D. P. Kroese. The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics). Berlin, Heidelberg: Springer-Verlag, 2004. ISBN: 038721240X.

- [150] R. Rubinstein. "The cross-entropy method for combinatorial and continuous optimization". In: *Methodology and computing in applied probability* 1.2 (1999), S. 127–190. DOI: 10.1023/A:1010091220143.
- [151] S. H. Sack und M. Serbyn. "Quantum annealing initialization of the quantum approximate optimization algorithm". In: *Quantum* 5 (Juli 2021), S. 491. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-01-491.
- [152] M. Salm, J. Barzen, U. Breitenbücher, F. Leymann, B. Weder und K. Wild. "The NISQ Analyzer: Automating the Selection of Quantum Computers for Quantum Algorithms". In: *Service-Oriented Computing*. Hrsg. von S. Dustdar. Cham: Springer International Publishing, 2020, S. 66–85. ISBN: 978-3-030-64846-6.
- [153] T. Sandholm, A. Gilpin und V. Conitzer. "Mixed-Integer Programming Methods for Finding Nash Equilibria". In: *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*. AAAI'05. Pittsburgh, Pennsylvania: AAAI Press, 2005, 495–501. ISBN: 157735236x.
- [154] G. E. Santoro und E. Tosatti. "Optimization using quantum mechanics: quantum annealing through adiabatic evolution". In: *Journal of Physics A: Mathematical and General* 39.36 (Aug. 2006), R393. DOI: 10.1088 /0305-4470/39/36/R01.
- [155] B. Schmitt. Tweedledum. 2018. URL: https://github.com/boschmitt /tweedledum.git (besucht am 30.01.2023).
- [156] E. Schubert, J. Sander, M. Ester, H. P. Kriegel und X. Xu. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: ACM Trans. Database Syst. 42.3 (Juli 2017). ISSN: 0362-5915. DOI: 10.1145/3068335.
- [157] R. Shaydulin, I. Safro und J. Larson. "Multistart methods for quantum approximate optimization". In: 2019 IEEE High Performance Extreme Computing Conference (HPEC). IEEE. 2019, S. 1–8.
- [158] P. W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: SIAM review 41.2 (1999), S. 303–332.
- [159] C. H. da Silva Santos, M. S. Gonçalves und H. E. Hernández-Figueroa. "Designing Novel Photonic Devices by Bio-Inspired Computing". In: *IEEE Photonics Technology Letters* 22.15 (2010), S. 1177–1179. DOI: 10.1109/LPT.2010.2051222.
- [160] V. Soni, S. Singh und M. P. Wellman. "Constraint Satisfaction Algorithms for Graphical Games". In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. AA-MAS '07. New York, NY, USA: Association for Computing Machinery, 2007. ISBN: 9788190426275. DOI: 10.1145/1329125.1329206.

- [161] J. Su, T. Tu und L. He. "A quantum annealing approach for Boolean Satisfiability problem". In: 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC). 2016, S. 1–6. DOI: 10.1145/2897937.28 97973.
- [162] R. Sutor. Dancing with Qubits: How Quantum Computing Works and how it Can Change the World. Expert Insight. Packt Publishing, 2019.
   ISBN: 9781838827366.
- [163] M. Suzuki. "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems". In: *Communications in Mathematical Physics* 51.2 (1976), S. 183–190. DOI: 10.1007/BF01609348.
- [164] G. Tavares. New algorithms for Quadratic Unconstrained Binary Optimization (QUBO) with applications in engineering and social sciences. Rutgers The State University of New Jersey-New Brunswick, 2008. DOI: 10.7282/T3XK8FS2.
- [165] A. Thomas und J. van Leeuwen. "Pure Nash equilibria in graphical games and treewidth". In: *Algorithmica* 71.3 (2015), S. 581–604. DOI: 10.1007/s00453-014-9923-3.
- [166] F. Truger, M. Beisel, J. Barzen, F. Leymann und V. Yussupov. "Selection and Optimization of Hyperparameters in Warm-Started Quantum Optimization for the MaxCut Problem". In: *Electronics* 11.7 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11071033.
- [167] H. Ushijima-Mwesigwa, C. F. A. Negre und S. M. Mniszewski. "Graph Partitioning Using Quantum Annealing on the D-Wave System". In: *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*. PMES'17. New York, NY, USA: Association for Computing Machinery, 2017, 22–29. ISBN: 9781450351263. DOI: 10.1145/3 149526.3149531.
- [168] V. Vargas-Calderón, N. Parra-A., H. Vinck-Posada und F. A. González. "Many-Qudit Representation for the Travelling Salesman Problem Optimisation". In: *Journal of the Physical Society of Japan* 90.11 (2021), S. 114002. DOI: 10.7566/JPSJ.90.114002.
- [169] D. Vickrey und D. Koller. "Multi-Agent Algorithms for Solving Graphical Games". In: Eighteenth National Conference on Artificial Intelligence. USA: American Association for Artificial Intelligence, 2002, 345–351. ISBN: 0262511290.
- Q. Wang, Y. Ma, K. Zhao und Y. Tian. "A comprehensive survey of loss functions in machine learning". In: Annals of Data Science 9.2 (2022), S. 187–212. DOI: 10.1007/s40745-020-00253-5.

- [171] X. Wang und T. Sandholm. "Reinforcement Learning to Play an Optimal Nash Equilibrium in Team Markov Games". In: Proceedings of the 15th International Conference on Neural Information Processing Systems. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, 1603–1610. DOI: 10.5555/2968618.2968817.
- [172] L. Wasserman. All of nonparametric statistics. Springer Science & Business Media, 2006. DOI: 10.1007/0-387-30623-4.
- [173] A. Weinstein und M. L. Littman. "Open-Loop Planning in Large-Scale Stochastic Domains". In: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. AAAI'13. AAAI Press, 2013, 1436–1442.
- [174] C. P. Williams, S. H. Clearwater et al. Explorations in quantum computing. Springer, 1998. DOI: 10.1007/978-1-84628-887-6\_2.
- [175] M. Willsch, D. Willsch, F. Jin, H. De Raedt und K. Michielsen. "Benchmarking the quantum approximate optimization algorithm". In: *Quan*tum Information Processing 19.7 (2020), S. 1–24. DOI: 10.1007/s1112 8-020-02692-8.
- [176] D. Xu und Y. Tian. "A comprehensive survey of clustering algorithms". In: Annals of Data Science 2.2 (2015), S. 165–193. DOI: 10.1007/s407 45-015-0040-1.
- [177] A. Yalaoui, H. Chehade, F. Yalaoui und L. Amodeo. Optimization of logistics. John Wiley & Sons, 2012. DOI: 10.1002/9781118569597.
- [178] S. Yang, W. Zi, B. Wu, C. Guo, J. Zhang und X. Sun. "Quantum logic synthesis for Satisfiability Problems". In: arXiv preprint ar-Xiv:2101.05430 (2021).
- [179] S. Yarkoni, A. Plaat und T. Back. "First Results Solving Arbitrarily Structured Maximum Independent Set Problems Using Quantum Annealing". In: 2018 IEEE Congress on Evolutionary Computation (CEC). 2018, S. 1–6. DOI: 10.1109/CEC.2018.8477865.
- [180] S. Yarkoni, H. Wang, A. Plaat und T. Bäck. "Boosting quantum annealing performance using evolution strategies for annealing offsets tuning". In: International Workshop on Quantum Technology and Optimization Problems. Springer. 2019, S. 157–168.
- T. Yokota. "Infinite range Ising spin glass model with a transverse field".
  In: *Physics Letters A* 125.9 (1987), S. 482–484. ISSN: 0375-9601. DOI: 10.1016/0375-9601(87)90190-3.
- [182] E. Zahedinejad und A. Zaribafiyan. "Combinatorial optimization on gate model quantum computers: A survey". In: arXiv preprint ar-Xiv:1708.05294 (2017).

- [183] S. Zbinden, A. Bärtschi, H. Djidjev und S. Eidenbenz. "Embedding Algorithms for Quantum Annealers with Chimera and Pegasus Connection Topologies". In: *High Performance Computing*. Hrsg. von P. Sadayappan, B. L. Chamberlain, G. Juckeland und H. Ltaief. Cham: Springer International Publishing, 2020, S. 187–206. ISBN: 978-3-030-50743-5.
- [184] L. Zhou, S.-T. Wang, S. Choi, H. Pichler und M. D. Lukin. "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices". In: *Phys. Rev. X* 10 (2 Juni 2020), S. 021067. DOI: 10.1103/PhysRevX.10.021067.