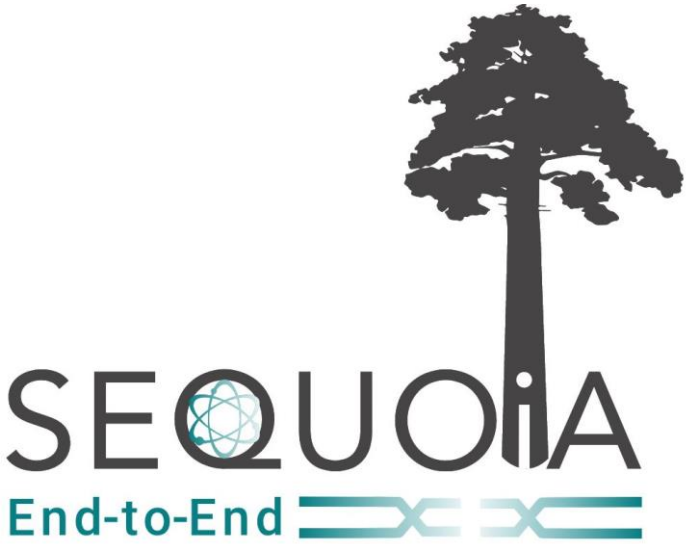
Abschlussbericht 2024

im Rahmen der zweiten Förderphase des
»Kompetenzzentrum Quantencomputing
Baden-Württemberg«



Transparentes Quanten-Software-Engineering und Algorithmendesign anwendungszentrierter End-to-End Lösungen

Eingereicht durch Fraunhofer IAO

Konsortialführung und koordinierendes Fraunhofer-Institut:

Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO
Prof. Dr.-Ing. Oliver Riedel
Projektleiter: Dr. Christian Tutschku

In Kooperation mit:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
Prof. Dr.-Ing. Marco Huber

Fraunhofer-Institut für Angewandte Festkörperphysik IAF
Prof. Dr. Rüdiger Quay

Fraunhofer-Institut für Kurzzeitdynamik, Ernst-Mach-Institut, EMI
Prof. Dr.-Ing. habil. Stefan Hiermaier

FZI Forschungszentrum Informatik
Prof. Dr. rer. nat. Ralf H. Reussner

Universität Stuttgart, Höchstleistungsrechenzentrum Stuttgart (HLRS)
Prof. Dr.-Ing. Dr. h.c. Dr. h.c. Prof. E.h. Michael M. Resch

Eberhard Karls Universität Tübingen (EKUT), Lehrstuhl Eingebettete Systeme
Prof. Dr. Oliver Bringmann

Albert-Ludwigs-Universität Freiburg (ALU), Lehrstuhl für Theoretische Physik
Prof. Dr. Andreas Buchleitner

Karlsruher Institut für Technologie (KIT)
Institut für Informationssicherheit und Verlässlichkeit
Prof. Dr. Ina Schaefer

Inhalt

Vorwort:

Im Folgenden werden die Arbeiten und Resultate aller neun Projektpartner nach abgeschlossener Projektphase präsentiert. Nach einer Kurzbeschreibung des Projektes und einer Management-Summary der erarbeiteten Resultate werden die in den Arbeitspaketen erarbeiteten Ergebnisse detailliert vorgestellt. Zuletzt schließt der Bericht mit einem Überblick über die im Projekt durchgeführte Öffentlichkeitsarbeit sowie die eingereichten und sich in Vorbereitung befindenden Publikationen.

Der fachliche Beitrag aller Forschenden ist in ihrer jeweiligen Muttersprache (Deutsch/Englisch) verfasst. Auf ein nachträgliches redaktionelles Übersetzen wurde verzichtet, um den Sinn und den Inhalt aller Aussagen exakt beizubehalten. Bei Unverständlichkeiten oder Unklarheiten kann eine Übersetzung ins Deutsche im Nachhinein aber natürlich jederzeit von der Projektleitung eingeholt werden.

1 Kurzbeschreibung und Projektstruktur

Bevor die erarbeiteten Ergebnisse des SEQUOIA End-to-End Projektes dargestellt werden, folgt zuerst eine Kurzbeschreibung des Projektes und dessen Aufbaus. Dies soll die Verortung der ab Kapitel 2 präsentierten Ergebnisse im Projekt erleichtern sowie als Überblick und Einstieg in die Projektthemen dienen.

1.1 Kurzbeschreibung

Wesentliches Ziel des Verbundforschungsprojektes »SEQUOIA End-to-End« ist es, die heutigen Engpässe im gesamten Quanten-Software-Entwicklungsprozess transparent zu machen und durch ganzheitliches Quanten-Software-Engineering performante, automatisierte und steuerbare End-to-End-Lösungen für industrielle Anwendungsfälle zu erforschen und bereitzustellen. Dazu wird in drei Fokusfeldern Forschungs- und Entwicklungsarbeit geleistet:

Im ersten Feld wird ein präzises Verständnis aller Prozesse in der Quanten-Software-Entwicklung und insbesondere in deren Schnittstellen entwickelt. Darauf basierend wird eine optimale, algorithmenspezifische Nutzung aktueller Quantencomputer (insbesondere dem IBM Q System One Ehningen) mittels performanter Transpilations- und Fehlermitigationspipelines erforscht, sowie eine fundierte Potenzialabschätzung für die industriellen Anwendungsfälle des projektbegleitenden Unternehmensnetzwerkes abgeleitet. Dieses besteht derzeit aus 32 assoziierten Partnern, die Ihr Interesse und mitwirken am Projekt mittels »Letters-of-Intent« bekräftigt haben.

Das zweite Fokusfeld ist dem Quanten-Software-Engineering gewidmet. Hier werden Werkzeuge und Werkzeugketten zur effizienten Entwicklung von Quantenapplikationen erforscht sowie Techniken für systematisches Testen und Debugging entwickelt, um Verlässlichkeit und damit industrielle Einsetzbarkeit zu ermöglichen. Ein vertieftes Verständnis aller Parametereinflüsse und deren gegenseitige Abhängigkeiten bei Auswahl und Auslegung von Quantenalgorithmen und deren Transpilation wird durch die Methode Design Space Exploration erforscht. Daraus werden algorithmenspezifisch optimale Parameter für hybride Quantenanwendung abgeleitet.

Im dritten Fokusfeld werden basierend auf den industriellen Anwendungsfällen des Unternehmensnetzwerkes Benchmarks entwickelt, welche den Stand des Quantencomputings für verschiedene industrielle Fragestellungen quantifizieren. Es werden End-to-End-Demonstratoren entwickelt und bereitgestellt, welche Best-Practices für den gesamten Quanten-Software-Entwicklungsprozess liefern und diese auch für Nicht-Quantencomputing-Experten nachvollziehbar machen.

1.2 Projektstruktur

Abb. 1 gibt einen Überblick, wie die zugrundeliegende Projektstruktur auf die verschiedenen Arbeitspakete aufgeteilt ist und wie die Arbeitspakete ineinandergreifen. In AP1 werden die Anwendungsfälle der Industriepartner behandelt. Es umfasst die mathematische Formulierung der Probleme, die Auswahl und Anpassung eines geeigneten Quantenalgorithmus (in Zusammenspiel mit AP2) und das Mapping auf das zugehörige QC-Problem sowie die Reduktion des Problems auf eine für heutige Quantencomputer geeignete Größe (Proof-of-Concept, POC) und schließlich die Entwicklung von End-To-End-Demonstratoren. In AP2 werden Quantenalgorithmen im Detail analysiert, implementiert und auf die Eignung für die gegebenen Anwendungsfälle untersucht bzw. optimiert. Auch der Einsatz von Hochleis-tungsrechnern in Zusammenspiel mit Quantencomputern wird hier untersucht. An AP2 schließt AP3 nahtlos an, in welchem die optimale Ausführung der identifizierten Quan-tenalgorithmen auf heutigen fehlerbehafteten Quantencomputern erforscht wird. Hier werden Methoden und Werkzeuge sowohl für den Transpilationsprozess als auch für die Fehlermitigation entwickelt und zur Verfügung gestellt. Die Resultate aus AP1 - 3 werden in AP4 durch Benchmarks systematisch auf mögliche Vorteile, z.B. in Bezug auf Qualität oder Laufzeit, untersucht. In diesem AP wird zudem die Performance gängiger SDKs auf klassischer Hardware gebenchmarkt und ein systematischer Vergleich von klassischen Optimierern, welche für variationelle Quantenalgorithmen essenziell sind, erstellt. AP5 ist dem Quanten-Software-Engineering gewidmet, in welchem grundlegende Arbeitsschritte in modulare Softwarewerkzeuge abstrahiert werden, die Qualität von Quantensoftware durch Test- und Debugging-Methoden abgesichert wird und Quanten-Applikationen über die PlanQK Plattformen deployed werden. Zudem werden die Effekte und das Zusammenspiel der Parameter und Freiheitsgrade bei der Auswahl und der Auslegung von Algorithmen und ihrer Transpilationspipeline analysiert (Design Space Exploration). In AP6 sind Transfer-, Kooperations- und Schulungsaktivitäten im Rahmen des Anwendungszentrums zusammengefasst. Hier wird der Wissenstransfer (u.a. in das Unternehmensnetzwerk und das weiter gespannte Quantum^{BW}-Netzwerk) beispielsweise mithilfe der entwickelten End-To-End-Demonstratoren geleistet.

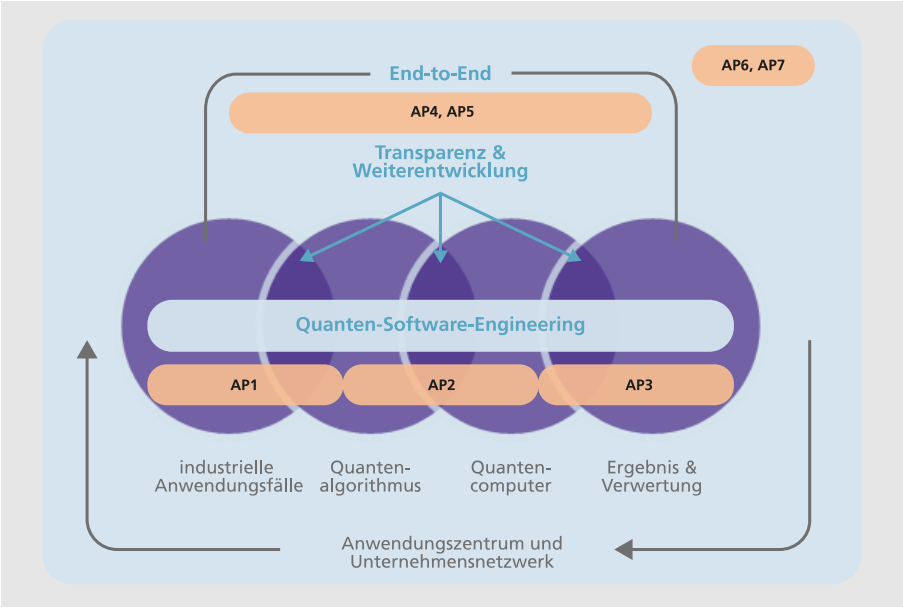


Abb. 1 Zugrundeliegende Projektstruktur des Verbundforschungsvorhabens »SEQUOIA End-to-End«

Abb. 2 zeigt weiter das zugehörige und im Dezember 2020 bewilligte Gantt-Diagramm, auf dessen Meilensteine M8 und M15 in den folgenden Kapiteln explizit eingegangen wird.

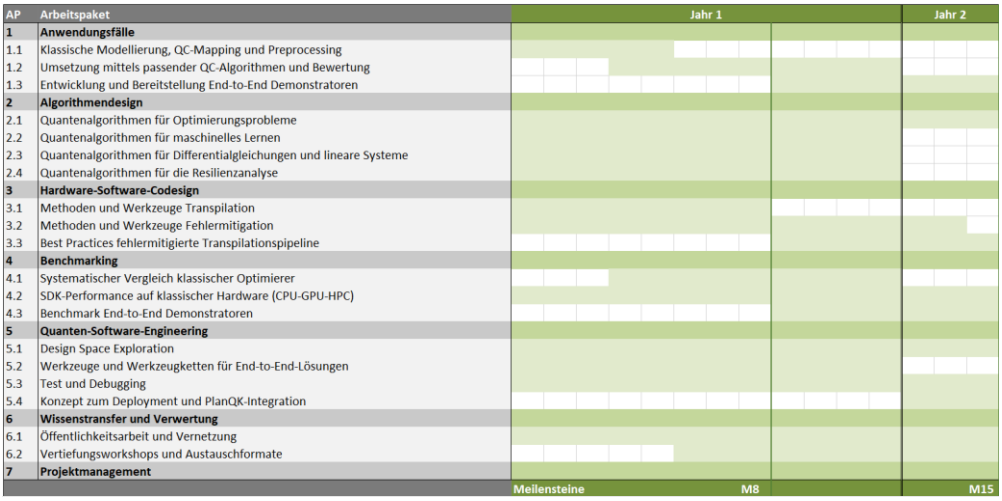


Abb. 2 Gantt-Diagramm und Meilensteine des Verbundforschungsprojektes »SEQUOIA End-to-End«.

1.3 Key Performance Indikatoren (KPIs)

Bevor ab Kapitel 2 die Ergebnisse der dargestellten Projektstruktur im Detail erörtert werden, soll an dieser Stelle noch einmal zusammenfassend auf die Key Performance Indikatoren des »SEQUOIA End-to-End« Projektes eingegangen werden.

(1) **Insgesamt werden im Projekt 7 Anwendungsfälle analysiert** sowie prototypisch umgesetzt bzw. weiterentwickelt (Vorprojekt »SEQUOIA«). Deren finaler Stand wird detailliert in den Kapiteln 2.1 und 2.2 diskutiert.

1.1. **Quantenbasierte numerische Strömungssimulation**
(Forschung Fraunhofer IAO | Unternehmen Simerics & Pfizer)

Im Kontext der Fluidsimulation sowie der Simulation von Tabletierungsprozessen wird das quantenbasierte Lösen von gekoppelten Differentialgleichungen mittels Circuit Learning Methoden analysiert, bewertet und gegen VQLS-Ansätze verglichen.

1.2. **Routenplanung von LKW-Flotten im Supply Chain Management**
(Forschung Fraunhofer IAO | Unternehmen Schwarz IT & LamA)

Die bereits im Vorprojekt »SEQUOIA« als QUBO formulierten Optimierungsprobleme »Routenoptimierung« und »Ladesäulenoptimierung« werden mittels VQE und Quantum-Annealing (Dwave) Lösungen umgesetzt und gegen die erarbeiteten QAOA (+) Lösungen gebenchmarkt.

1.3. **Szenario-basierte Routenplanung zur Absicherung von Automotive Fahrfunktionen** (Forschung EKUT + Unternehmen Bertrandt)

Testszenarien für automatisierte Fahrfunktionen sollen in der realen Umgebung möglichst effizient durchlaufen werden. Zu diesem Zweck wird, vom Standpunkt des Testfahrzeugs aus, eine Strecke auf Basis von Kartendaten ermittelt, auf der in einer vorgegebenen Zeit möglichst viele Ampelszenarios passiert werden.

1.4. **Resilienzanalyse kritischer Infrastrukturnetze**
(Forschung EMI und ALU)

Die Quantennetzwerke, die im Vorprojekt EFFEKTIF entwickelt wurden und die Dynamik eines Infrastrukturnetzwerks, mit Ausfällen und Wiederherstellungen, widerspiegeln, wurden auf dem Quantencomputer simuliert und erweitert. Außerdem wurde ein neuer Ansatz, basierend auf QAE, getestet, um bessere Skalierung für größere Netzwerke zu erreichen.

1.5. **Verifikation neuronaler Netze in der Verkehrszeichenerkennung**
(Forschung IPA)

Neuronale Netze in sicherheitskritischen Anwendungen unterliegen besonderer Sorgfaltspflicht. Die Verifikation ist ein Prozess, der sicherstellen soll, dass die Vorhersagen dieser Netze in sicherheitskonformen Schranken liegen. In diesem Anwendungsfall wurde ein quanten-klassischer Hybridalgorithmus zur Verifikation neuronaler Netze entwickelt und für das Szenario der Verkehrszeichenerkennung in selbstfahrenden Autos erprobt.

1.6. **Kostenoptimierung und Auslegung von Fertigungsstraßen**
(Forschung IPA + Unternehmen Denso)

Eine Fertigungsstraße soll unter Einhaltung gegebener Bedingungen möglichst kostengünstig ausgelegt und eingerichtet werden. Hierzu wurde das kombinatorische Optimierungsproblem als QUBO formuliert und mit einem hybriden Ansatz unter Verwendung von Quantum Annealing gelöst.

1.7. **Konfigurationspriorisierung für variable Softwaresysteme**
(Forschung KIT, besonderes Interesse VW)

Die Priorisierung von Konfigurationen eines hochkonfigurierbaren Systems, z.B. zur Identifizierung von Testkonfigurationen, wurde mithilfe von Quanten Computing adressiert. Dazu muss das Optimierungsproblem in einer geeigneten Form (z.B. QUBO) formuliert und mit einem geeignetem hybriden Quantenalgorithmus (z.B. QAOA) umgesetzt werden.

(2) **Insgesamt wurden im Projekt 10 Werkzeuge entwickelt bzw.** für den anwendungsorientierten Einsatz **erprobt.**
(Die Ergebnisse werden in Kapitel 3 ausführlich erläutert)

2.1. Qutools für eine automatisierte Pulsoptimierung
2.2. Classiq für die automatisierte Schaltkreisgenerierung
2.3. Hamiltonian Simulation zur Zeitentwicklung von Energiefunktionalen
2.4. Zero-noise-extrapolation zur Fehlermitigation
2.5. Probabilistic-Error-Cancellation (PEC) zur Fehlermitigation
2.6. Reinforcement Learning für die automatisierte Schaltkreisgenerierung
2.7. Alb-qubo Werkzeug zur Formulierung von Fertigungsstraßen
2.8. Ganzheitlich integrierte QC-Entwicklungsumgebung (IDE)
2.9. Transpilationsmethoden auf Basis von Redundanzansätzen
2.10. Vorhersage von PDE-Lösungen mittels QNNs und PIQNNs

(3) Im Projekt befinden sich **16 Publikationen** und 2 Master-Abschlussarbeiten in Vorbereitung, in Review bzw. wurden schon akzeptiert. Detailliert ist dies Kapitel 3 zu entnehmen.

(4) Hinsichtlich Wissenstransfer wurden im Projekt **45 Schulungs- und Transferveranstaltungen mit einer Gesamtzahl von ca. 2700 geschulten Personen** durchgeführt.

Die neu-erarbeiteten Demonstratoren wurden in die bestehende Projektwebsite (https://www.sequoia-iao.de/) unter den Rubriken »Use Cases« und »Software Tools« integriert und als solche gekennzeichnet. Dies umfasst die Use-Case-Demonstratoren und Software Tools in Form von (interaktiven) Notebooks, die über die Infrastruktur des Fraunhofer-GitLab bereitgestellt und über Binder explorativ und frei zugänglich sind.

Im Folgenden werden die im Projekt erarbeiteten Ergebnisse detailliert und AP-spezifisch dargestellt. Wie der Projektstruktur in Kapitel 1.2 zu entnehmen ist, fließen jedoch viele Ergebnisse direkt in die Umsetzung der Anwendungen in AP1. Daher soll an dieser Stelle ausdrücklich darauf hingewiesen sein, dass hier ein starker Überlapp zwischen den APs besteht und auf einige AP2-5 Ergebnisse daher bereits in AP1 eingegangen wird.

2 Überblick der wissenschaftlichen Arbeiten

2.1 Arbeitspaket 1 – Anwendungsfälle

Definiertes Ziel des ersten Arbeitspakets ist laut Projektbeschreibung die Entwicklung und Bereitstellung von End-to-End-Demonstratoren für eine prototypische Umsetzung von industriellen Anwendungsfällen aus dem Unternehmensnetzwerk. Nachvollziehbarkeit und Dokumentation von Best Practices entlang der gesamten Anwendungsentwicklung stehen dabei in besonderem Fokus.

Unter Leitung des **Fraunhofer IAO** wurde dies für die in Kapitel 0 eingeführten Anwendungsfälle anhand dreier Arbeitsschritte erreicht (vgl. Gantt-Chart in Abb. 2)

AP 1.1 Klassische Modellierung, QC-Mapping und Preprocessing

- Übertragung der industriellen Anwendungsfälle in mathematische Modelle
- Reduktion der klassischen Modelle auf eine für heutige Quantencomputer adäquate Größe unter Beibehaltung der problemspezifischen Eigenheiten (POCs)
- Mapping der klassischen Modelle auf QC-Modelle und zugehöriges Preprocessing (z.B. Daten-Encodings, Pauli-Zerlegung von Matrizen, etc.)

AP 1.2 Umsetzung mittels passender QC-Algorithmen und Bewertung

- Wahl der Hyperparameter in klassischen und QC-Modellen
- Implementierung und Bereitstellung der klassischen und QC-Modelle
- Ausführung auf Simulatoren. Bewertung der QC-Potenziale mit Blick auf künftige Skalierungen der QC-Hardware, aber auch im Vergleich mit der Performance klassischer Lösungsmethoden

AP 1.3 Entwicklung und Bereitstellung End-to-End Demonstratoren

- Modularisierung der Anwendungsfälle mit Abkapselung technischer Details. Bereitstellung gut geeigneter Beispiele samt Parameter und Hyperparameter
- Erstellung von Jupyter Notebooks mit End-to-End-Durchläufen durch die Lösungs- und Entwicklungsschritte der Anwendungsfälle
- Dokumentation und Sicherstellung der Nachvollziehbarkeit von Modellierungs-, Entwurfs- und Implementierungsentscheidungen, Erarbeitung von Best Practices

Dabei liegen jenem Arbeitsplan insgesamt zwei Meilensteine zugrunde

- **M8:** Mathematisches Modell der Anwendungsfälle ist erarbeitet, geeigneter Quantenalgorithmus ist identifiziert und Mapping der mathematischen Modelle auf die QC-Modelle ist entwickelt.
- **M15:** End-to-End-Demonstratoren sind fertig entwickelt und bereitgestellt. Vorgehen ist dokumentiert.

Dabei sind beide Meilensteine (M8 und M15) planmäßig erreicht worden. Im Folgenden wird der Endstand der jeweiligen Use-Case-Forschung dargestellt, wobei nach den in Kapitel 0 erwähnten Anwendungsfällen unterschieden wird. Die folgenden Ergebnisse der Arbeitspakete 1.1 und 1.2 wurden zu wissenschaftlichen Publikationen ausgearbeitet und auf der »Quantum Effects« im Oktober 2023 vorgestellt (siehe Publikationen).

2.1.1 Routenplanung von LKW-Flotten im Supply Chain Management

Der bereits im Vorgängerprojekt SEQUOIA initial behandelte Anwendungsfall des Flottenroutings zur Reduktion von Leerkilometern ähnelt bzw. gleicht in seiner mathematischen Beschreibung stark dem der Optimierung von Ladeplänen [1] [2]. Beide Anwendungsfälle können als QUBO-Matrix formuliert und mittels QAOA-artiger oder VQE-basierter Lösungsansätze gelöst werden.
Daher repräsentieren die in den folgenden aufgezeigten Ergebnissen exemplarisch Benchmarks, Best Practices und Performance Charakteristika QUBO-basierter Optimierungsalgorithmen im Allgemeinen.

Basierend auf den SEQUOIA Use-Case-Ergebnissen »Optimierung von Ladeplänen« und »Routenplanung von LKW-Flotten« [1] wurde eine Reihe von systematischen Experimenten für QUBO-basierte Optimierungs-Use-Cases durchgeführt, um deren End-to-End Lösungen hinsichtlich verschiedener Input- Konfigurationen sowie Algorithmen- und / oder Backend-Charakteristika (Noise, Coupling-Map, etc.) zu bewerten bzw. zu benchmarken (Best-Practices). In [4] werden die Ergebnisse detailliert vorgestellt und diskutiert, so dass wir hier nur über einen kleinen Ausschnitt davon berichten.

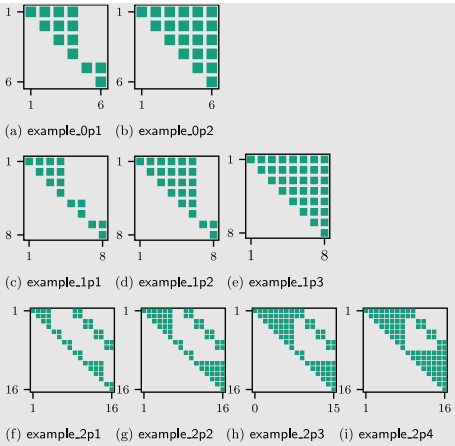


Abb. 3 Besetzungsstruktur der QUBO-Matrix dreier Beispielreihen mit 6 (oben), 8 (mitte) und 16 (unten) Qubits. Die Beispiele in den einzelnen Beispielreihen unterscheiden sich in ihrer Kopplungsstärke, was sich in der Besetzungsstruktur der QUBO-Matizen manifestiert.

In Abb. 3 sind beispielhaft die Besetzungsstruktur der QUBO-Matrizen dreier solcher Experimentreihen visualisiert. Es handelt sich um eine 6 Qubit, eine 8 Qubit und eine 16 Qubit Formulierung, wobei sich die einzelnen Experimente einer Reihe durch ihre Kopplungsstärke unterscheiden. Dies führt jeweils zu einer zunehmenden Anzahl an Nicht-Nulleinträgen in der QUBO-Matrix, wodurch die zugehörigen Probleminstanzen bei gleicher Qubit-Zahl schwieriger zu lösen sind. Bei gleicher Qubit-Anzahl können somit verschiedene Schwierigkeitsgrade abgebildet werden.
Durch Simulation verschiedenster Szenarien (auf den im KQCBW aufgebauten HPC-Ressourcen) wurden optimale Sets an variationellen Parametern für die Algorithmen QAOA und VQE bestimmt. In Abb. 4 sind beispielhaft einige Durchläufe des klassischen Optimierers COBYLA für eines der 8 Qubit Beispiele zu sehen.

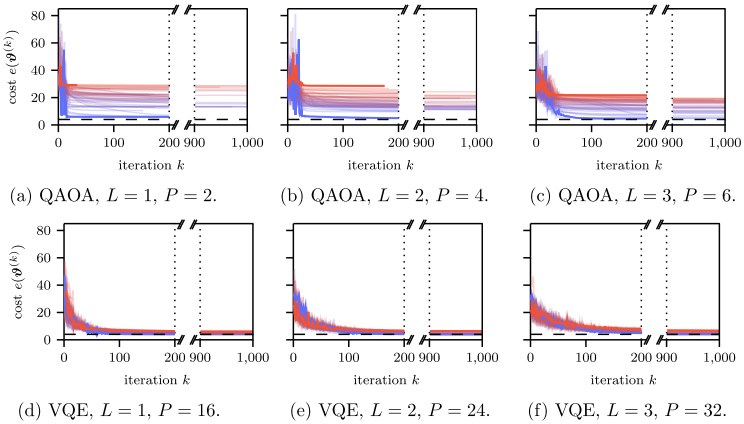


Abb. 4 Resultate der Optimierung der P variationellen Parameter der QAOA (obere Plots) bzw. VQE (untere Plots) Schaltkreise mit L=1, 2 und 3 Layern (links, mitte, rechts) für das 8 Qubit Beispiel example_1p3. Als Optimierer kam COBYLA zum Einsatz. Abgebildet ist die Zahl an Optimierungsiterationen gegen den Wert der Zielfunktion. Die Quantenschaltkreise wurden mit einem exakten Simulator ausgeführt. Für jeden Plot wurden 50 verschiedene, zufällig ausgewählte Startwerte für die Initialisierung des Optimierungsprozesses verwendet (eine Linie entspricht jeweils einem Startwert).

Aus diesen Ergebnissen können Quantenschaltkreise generiert werden, welche den Optimalfall eines Algorithmus abbilden, und mit welchen somit die Leistungsfähigkeit von aktuellen Quantencomputern bzw. deren Grenzen bewertet werden können. Hierbei wurden absichtlich Ergebnisse einer exakten Simulation verwendet, da die Optimierung der klassischen Parameter auf realen Quantencomputern einerseits sehr ressourcenintensiv und andererseits durch die Fehleranfälligkeit heutiger Quantencomputer ohnehin nur sehr unzureichend möglich ist.
In Abb. 5 ist eine solche Optimierung auf ibmq_ehningen abgebildet. Es ist sehr gut zu erkennen kann, dass für die QAOA-Schaltkreise im Prinzip keine Konvergenz des Optimierungsprozesses stattfindet. Dies kann unter anderem durch eine Betrachtung der Kostenlandschaft, in welcher die Optimierung stattfindet, erklärt werden. Eine solche ist in für den Fall einer exakten Simulation, einer Simulation basierend auf dem Fehlermodell von ibmq_ehningen und des realen ibmq_ehningen Backend gegeben.

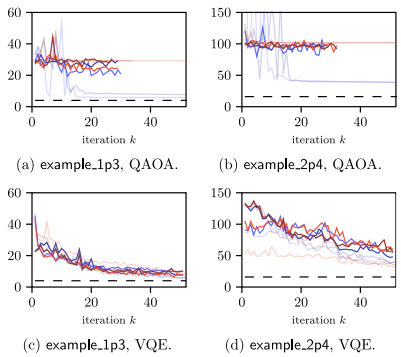


Abb. 5 Optimierung der variationellen Parameter der QAOA (obere Plots) bzw. VQE (untere Plots) Schaltkreise für L=1 Layer für das 8 Qubit Beispiel example_1p3 (links) und das 16 Qubit Beispiel example_2p4 (rechts) auf ibmq_ehningen. Die transparenten Linien stammen zum Vergleich von einer exakten Simulation.

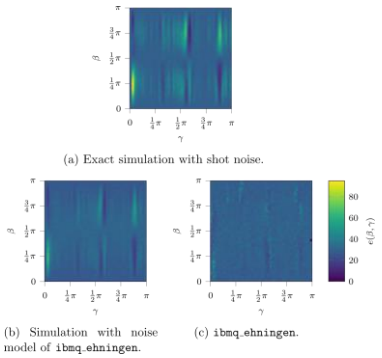
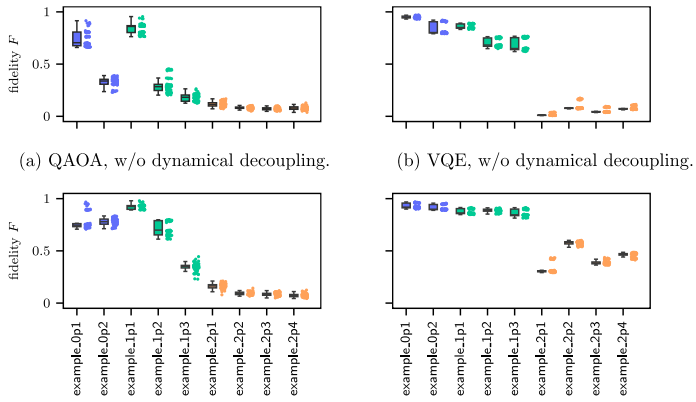


Abb. 6 Heatmap der Kostenfuntion für QAOA mit L=1 Layer für das 8 Qubit Beispiel example_1p3 für jeweils 50 verschiedene Werte der zwei variationellen Parameter β und γ. Die Schaltkreise wurden mit einem exakten Simulator (oben), einem Simulator mit dem Fehlermodell von ibmq_ehningen (unten links) und auf dem realen Backend ibmq_ehningen (unten rechts) ausgeführt.

Es ist sehr deutlich zu sehen, dass für das reale Backend nur ein sehr schwaches und deutlich verrauschtes Signal vorliegt, auf Basis dessen eine Optimierung äußerst schwierig ist.

Um End-to-End Durchläufe im Detail zu verstehen, wurden alle Beispielreihen auf den IBM Quantencomputern und insbesondere dem System ibmq_ehningen ausgeführt und bezüglich verschiedener Qualitätscharakteristiken verglichen. Ein Beispiel ist in Abb. 7 gegeben, in welchem die Fidelity bezüglich einer exakten Simulation zur Beurteilung der

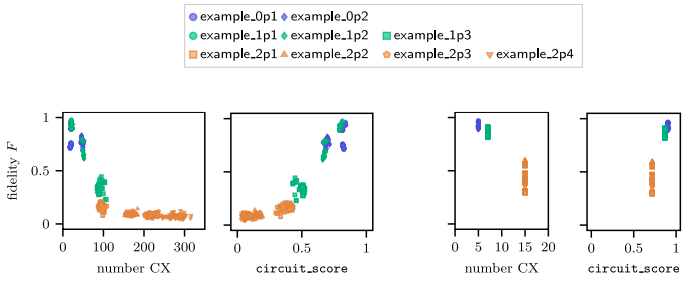


(a) QAOA, w/o dynamical decoupling. (b) VQE, w/o dynamical decoupling.

(c) QAOA, w/ dynamical decoupling. (d) VQE, w/ dynamical decoupling.

Abb. 7 Fidelity F der Ergebnisse der Ausführung der trainierten QAOA (links) und VQE (rechts) Schaltkreise mit L=1 layer auf ibmq_ehningen berechnet bezüglich einer exakten Simulation dieser Schaltkreise. Für jedes Beispiel wurde der logische Schaltkreis mit 75 verschiedenen seeds transpiliert und jeder der so erhaltenen Transpilationen wurde auf ibmq_ehningen ausgeführt (jeder Punkt in den Plots entspricht einem transpilierten Schaltkreis). Zur Verbesserung der Resilienz der Qubits wurde „dynamical decoupling" verwendet (untere Plots). In den oberen Plots wurde dies nicht verwendet.

Qualität verwendet wird. Es ist sehr deutlich zu sehen, dass für kleine Probleminstanzen noch eine hohe Fidelity von ibmq_ehningen geliefert wird, während für die größeren Probleme eine Abnahme festzustellen ist. Dies wird durch Hardwarefehler verursacht. Eine simple Methode die Qubits resilienter gegen Fehler zu machen ist „dynamical decoupling", welches stets in den Hardwareexperimenten verwendet wurde. Um seinen Effekt zu verdeutlichen, wurde alle Experimente auch ohne „dynamical decoupling" durchgeführt. In erwähnter Abb. 7 ist der positive Effekt auf die Fidelity sehr gut zu beobachten (vergleiche obere und untere Plots).

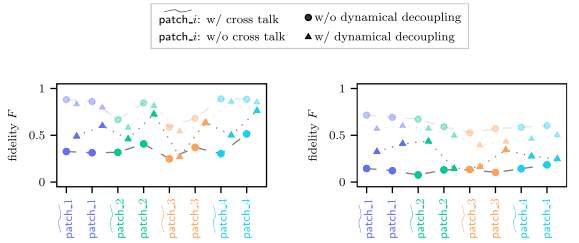


(a) QAOA, w/ dynamical decoupling. (b) VQE, w/ dynamical decoupling.

Abb. 8 Anzahl der CX Gates (erster und dritter Plot) und circuit_score (zweiter und vierter Plot) zur Beurteilung der transpilierten Schaltkreise basierend auf QAOA (linke Plots) bzw. VQE (rechte Plots) mit L=1 Layern. Die transpilierten Schaltkreise wurden auf ibmq_ehningen ausgeführt.

Es ist jedoch auch eine hohe Varianz innerhalb der Beispiele der Problemreihen zu erkennen. Hier führten verschiedenen Transpilationen des gleichen logischen Schaltkreises zu verschiedenen Ergebnissen, wenn diese auf ibmq_ehningen ausgeführt

wurden. Natürlicherweise ist dies sehr unerwünscht für zukünftige Quanten-Applikationen ist, da es die Zuverlässigkeit und Vorhersagbarkeit untergräbt. Deshalb wurden verschiedene Metriken für die transpilierten Schaltkreise betrachtet, insbesondere solche zur Beurteilung der zu erwartenden Qualität nach der Ausführung auf einem realen Backend. In Abb. 8 zeigen wir mit der Anzahl an CX-Gates und dem **circuit_score** zwei solcher Metriken. Wir sehen, dass diese bis zu einem gewissen Maße die Fidelity eines Schaltkreises vorhersagen können. Jedoch ist auch hier in jedem Beispiel eine Varianz vorhanden. Es muss also Fehlerquellen geben, die durch diese Metriken nicht erkannt werden. Eine mögliche solche Quelle ist „Cross Talk", welcher in [3] für **ibmq_ehningen** untersucht wurde. Hierbei handelt es sich um Cross Talk bei supraleitenden Transmon-Qubits, der auf der Kollision von Übergangsfrequenzen zwischen benachbarten Qubits beruht. Basierend darauf und einer Reihe eigener Experimente wurden die Qubits identifiziert, welche besonders stark unter dieser Form von Cross Talk leiden. Für diese wurde untersucht, ob es eine Auswirkung auf die Qualität der QAOA-Resultate gibt. In Abb. 11 sind die Ergebnisse visualisiert. Für einige Beispiele ist tatsächlich eine verringerte Fidelity zu beobachten, wenn Cross Talk Qubits involviert sind. Jedoch gilt dies nicht allgemein, woraus geschlussfolgert werden kann, dass Cross Talk nur ein Teil der Erklärung der Varianz in den Resultaten ist.



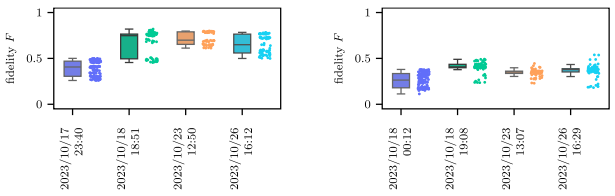
(a) QAOA, example_0p2, experiment ran on 2023/11/22 18:22.

(b) QAOA, example_1p3, experiment ran on 2023/10/27 20:18

Abb. 9 Vergleich verschiedener Ausschnitte der Qubits von **ibmq_ehningen**, wobei jeweils ein Ausschnitt Cross Talk Qubits enthält und der andere nicht (siehe Legende). Es wurde ein QAOA Schaltkreis von L=1 Layern verwendet. Die transparenten Marker stammen von einer Simulation mit dem Fehlermodell von **ibmq_ehningen**.

Es ist bekannt, dass die Fehler von aktuellen Quantencomputern nicht konstant sind, sondern sich zeitlich verändern. Dies wirkt sich selbstverständlich auch auf die Qualität der Resultate aus, was die Reproduzierbarkeit von Experimenten und daraus folgend auch die Beurteilung der Quantencomputer selbst deutlich erschwert. In Abb. 9 ist dies beispielhaft für zwei QAOA-Schaltkreise dargestellt, die an vier verschiedenen Tagen auf **ibmq_ehningen** ausgeführt wurden. Es sind sehr deutliche Unterschiede über die Tage hinweg festzustellen, nicht nur in der Verteilung der Fidelity sondern auch in ihrer Varianz.

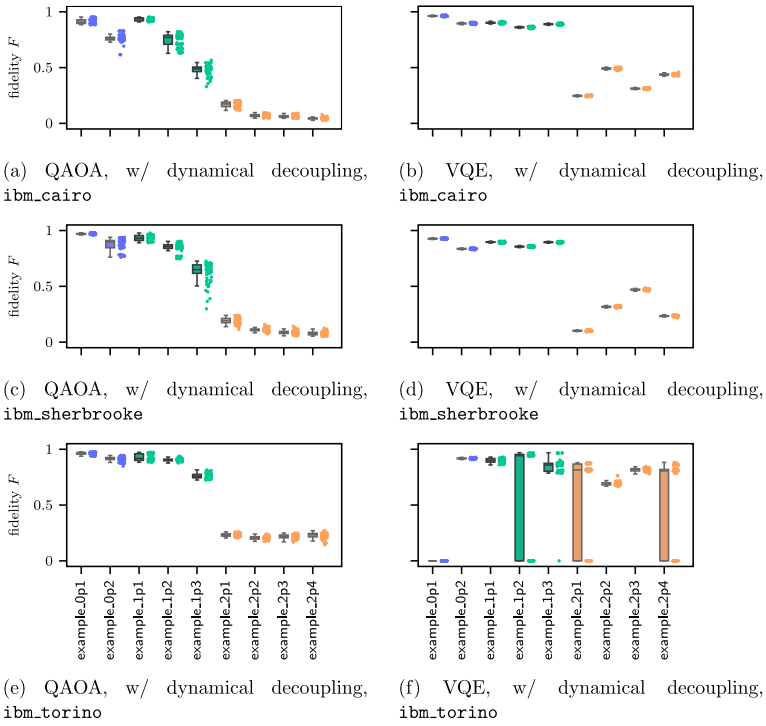
Die Quantencomputer-Hardware wird stetig fortentwickelt und verbessert. Um diesen Fortschritt abzubilden und zu bewerten, wurden alle QAOA und VQE-Beispiele auf drei weiteren IBM Backends ausgeführt, nämlich: **ibm_cairo** (gleiche QPU wie **ibmq_ehningen**), **ibm_sherbrooke** (127 Qubits, vorgestellt Dezember 2022) und **ibm_torino** (133 Qubits, vorgestellt Dezember 2023, aktuell bester verfügbare IBM QPU). In Abb. 10 sind die Fidelities der verschiedenen Backends dargestellt. Ein sehr deutlicher Sprung in der Qualität ist insbesondere für **ibm_torino** auszumachen. Wobei auch hier die größten QAOA-Instanzen nur mit einer unzureichenden Qualität ausgeführt werden können.



(a) QAOA, w/ dynamical decoupling, example_1p2

(b) QAOA, w/ dynamical decoupling, example_1p3

Abb. 11 Fidelity von QAOA mit L=1 Layern und dynamical decoupling für zwei verschiedene Beispiele an vier unterschiedlichen Tagen. Der logische QAOA Schaltkreis wurde jeweils 75 Mal transpiliert und jede Transpilation wurde auf **ibmq_ehningen** ausgeführt (jeder Punkt in den Plots stammt von einem transpilierten Schaltkreis).



(a) QAOA, w/ dynamical decoupling, ibm_cairo

(b) VQE, w/ dynamical decoupling, ibm_cairo

(c) QAOA, w/ dynamical decoupling, ibm_sherbrooke

(d) VQE, w/ dynamical decoupling, ibm_sherbrooke

(e) QAOA, w/ dynamical decoupling, ibm_torino

(f) VQE, w/ dynamical decoupling, ibm_torino

Abb. 10 Gleiches Setup wie in **Abb. 7** ,jedoch wurden die Schaltkreise auf **ibm_cairo** (oben), **ibm_sherbrooke** (mitte) und **ibm_torino** (unten) ausgeführt.

Im Sinne des Benchmarkings wurden jene Probleminstanzen ebenfalls systematisch auf D-Wave Quanten-Annealern ausgeführt, um somit Best Practices für das Hardware-Software-Codesign zu erarbeiten. Da die Quanten-Annealing-Technologie bereits fortgeschrittener ist, wurden zusätzliche größere Use Cases entworfen (mit bis zu 48 logischen bzw. 192 physischen Qubits). In Abb. 12 zeigen wir beispielhaft Resultate. Im linken oberen Plot ist eine Einbettung des logischen QUBO-Problems auf die D-Wave Hardware zu sehen. Oben rechts wird die Qualität aller LamA-Beispiele ausgeführt auf dem D-Wave Annealer miteinander verglichen. Wir sehen, dass kleine Probleme gut gelöst werden können und wie mit Wachstum der Problemgröße die Qualität nachlässt. In den beiden unteren Plots werden Hardwareparameter gebenchmarkt. Ein detailliertes Verständnis dieser ist wichtig, um die bestmögliche Qualität von der Quanten-Hardware zu erhalten.

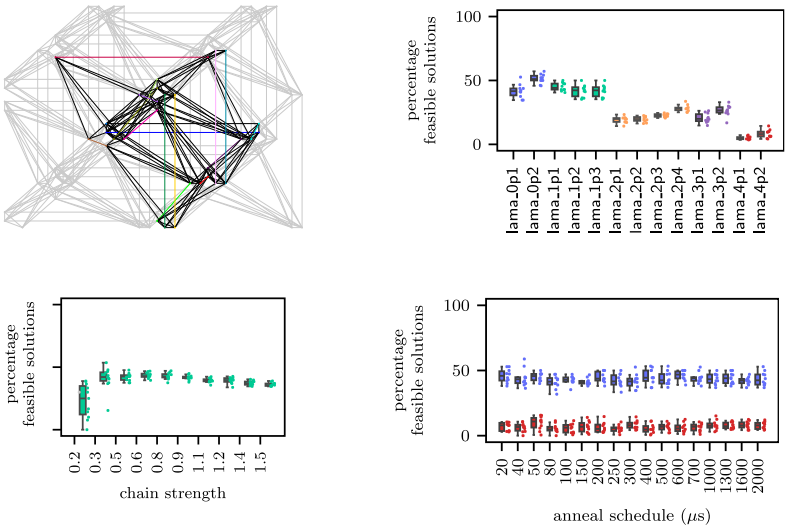
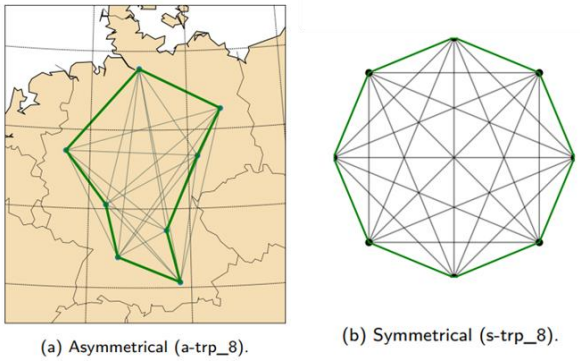


Abb. 12 LamA Use Case auf dem D-Wave Quanten-Annealer Advantage 4.1. Oben links ist die Einbettung der QUBO Matrix auf die QPU für das Beispiel lama_2p4 gegeben. Für den oberen rechten Plot wurden alle LamA Beispiele auf dem Quanten-Annealer ausgeführt. Die Qualität der Resultate wird hier mit dem Prozentsatz zulässiger Lösungen bezogen auf alle ausgelesenen Bitstrings beurteilt. Die unteren Plots dienen zum Benchmark verschiedener Hardwareparameter in Bezug auf die Resultatqualität. Dabei wird unten links die chain strength für das Beispiel lama_1p3 und unten rechts die anneal time für die Beispiele lama_0p1 (blau) and lama_4p2 (rot) betrachtet.

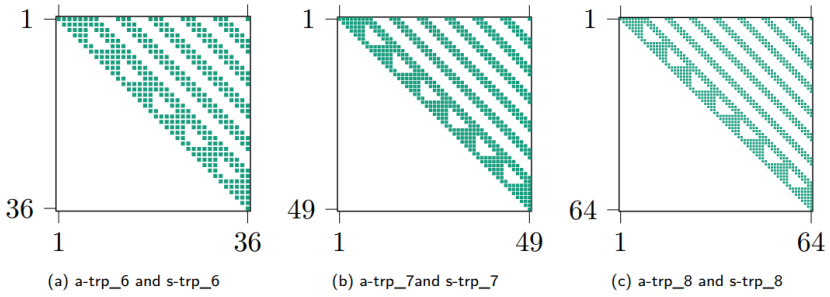
The alternative computing to the universal quantum computing like IBM systems is based on the physical concept called quantum annealing. Unlike the universal quantum computers, we do not perturb the system with the quantum gates while the quantum state is evolving. Instead, a general superposition state is initialized and then system slowly anneal into the final problem ground state solution. The truck routing problem is exciting problem for industrial partners as it is crucial logistics problem for many industries. We reduced the truck routing problem to travelling salesman problem by considering only one truck. We built the qubo matrix by using Dwave's ocean inherent qubo formulation.

In this study, we considered three different problem sizes. Each problem size has two different distributions, with one being asymmetric and the other radially symmetric. The asymmetrical distribution refers to a scenario where cities are placed randomly, resulting in a unique distance between each pair of cities (Abb. 13). This randomness introduces complexity to the optimization problem, presenting a more challenging scenario for solution algorithms. On the other hand, symmetrical distribution implies that cities are equidistantly placed on a ring (Abb. 13). This uniformity simplifies the optimization landscape, potentially leading to more straight-forward solutions. The asymmetric and symmetric distribution problem classes are referred as **a-trp_m** and **s-trp_m**, respectively, where **m** corresponds to number of cities. The asymmetric and symmetric distributions are given Abb. 13. We built QUBO matrices for our problems and their sparsity patterns are given in the Abb. 14.



(a) Asymmetrical (a-trp_8). (b) Symmetrical (s-trp_8).

Abb. 13 Examples for the two types of distributions that we consider in this paper: asymmetrical a-trp_8



(a) a-trp_6 and s-trp_6 (b) a-trp_7and s-trp_7 (c) a-trp_8 and s-trp_8

Abb. 14 Sparsity pattern of a QUBO matrix for 6, 7, and 8 cities.

Penalty Parameters

We analyze two different penalty parameters effecting the quality of solution. On the QUBO level, we used ϱ and on the hardware level the chain strength. The QUBO penalty parameter ϱ controls the constraints in the problem formulation. The chain strength is responsible for keeping the chains consistent, penalizing chain breaks, and is implemented on a lower level concerning the actual Hamiltonian that describes the hardware operation. The number of anneal reads were 1500 for each combination of penalty and chain strength value. The heat map plot for the chain strength and the penalty parameter is instructive as these parameters ensure the integrity of the solution and the satisfaction of constraints. The percentage of feasible and optimal solutions for the asymmetrical distribution of cities are given in Abb. 15.

One common observation for all the different number of cities is that for small chain strength, quantum annealing can hardly find and any feasible solutions, irrespective of strength of penalty. In the case of a-trp_6, the chain strength of 0.6 yields the highest percentage of solutions for nearly all penalties. On the contrary, the higher percentage of feasible solutions for a-trp_7 and a-trp_8 are obtained for chain strength of 0.9 and only for larger penalty values. Since the problems a-trp_7 and a-trp_8 needs to maintain the chain length of 6, 7 to hold 284 and 436 physical qubits, respectively, Tabelle 1.

Since the stronger chain strength can keep the longer chains and thus the original problem intact, they lead to higher percentages of feasible and optimal solutions. As the problem size increases the smaller chain strength values fail to yield any solutions. Higher percentage of optimal solutions for a-trp_6 are obtained for the same values of chain strength values as for feasible solutions. On the other hand, for a-trp_7 and a-trp_8, stronger chain strengths and smaller penalty values yielded optimal solutions. The higher percentage of feasible and optimal solutions do not share the same chain strength values. Overall, the percentage of feasible and optimal solutions reduces as we increase the problem size. In the case of symmetrical distribution of the cities, the percentage of feasible solutions for all cities is higher than in the asymmetrical distribution of cities, see Abb. 15 and Abb. 16. For s-trp_6, with a chain strength of 0.6 and for nearly all penalty

values, we obtain the highest number of feasible solutions. The same chain strength yields maximum optimal solutions. The feasible solutions for s-trp_7 are at higher for larger chain strength values which is similar to the asymmetrical distribution of cities. On the contrary to the asymmetrical distribution, the same chain strength is enough to obtain a higher percentage of both feasible and optimal solution for s-trp_7 and s-trp_8.

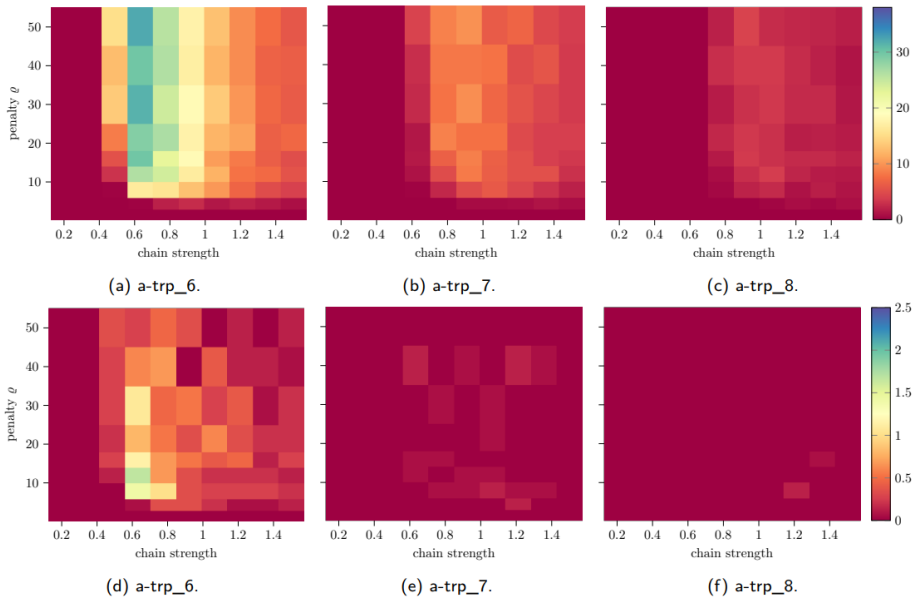


Abb. 15 Percentage of feasible (a) – (c) and optimal (d) – (f) solutions for asymmetrical distribution of cities with variable chain strength and penalty values. The number of anneal reads are 1500 for each combination of penalty and chain strength. Annealing time was chosen to be 300 µs. Clique embedding was used to embed the QUBO. The calculations were executed on Advantage 4.1 and results were obtained on 2023/11/06. (For the (a) – (c) the color scale ranges from 0 to 38 and for (d) – (f) from 0 to 2.5)

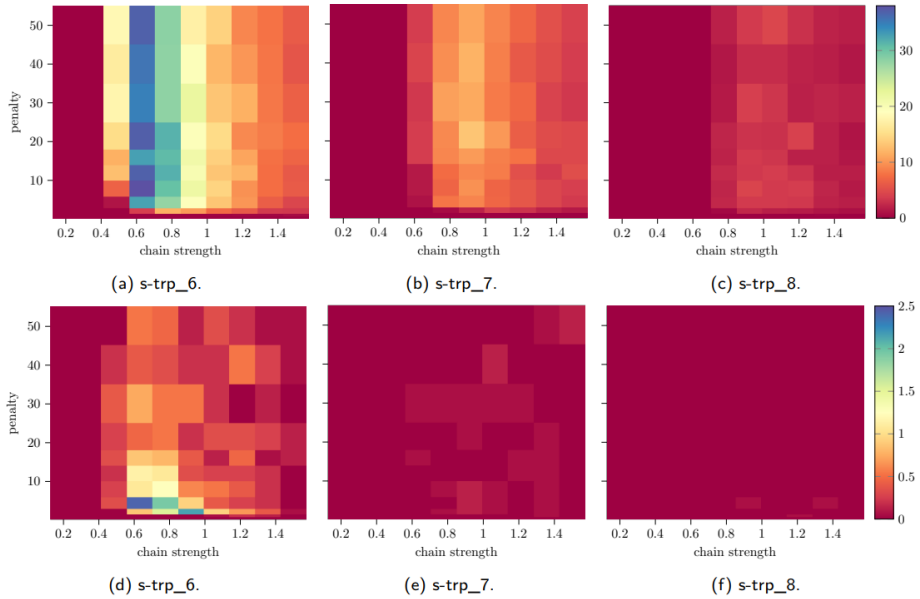


Abb. 16 Percentage of feasible (a) – (c) and optimal (d) – (f) solutions for symmetrical distribution of cities with variable chain strength and penalty values. The number of anneal reads are 1500 for each value of penalty and chain strength. Annealing time was chosen to be 300 µs. Clique embedding was used to embed the QUBO. The calculations were executed on Advantage 4.1 and results were obtained on 2023/11/11. (For the (a) – (c) the color scale ranges from 0 to 38 and for (d) – (f) from 0 to 2.5)

example	# logical qubits	# physical qubits	# of chains	chain length
6 cities	36	172	36	5
7 cities	49	284	49	6
8 cities	64	436	64	7

Tabelle 1 The number of logical qubits, physical qubits, chains, and chain length for all the problem sizes in the TRP use case. The symmetric and asymmetric problems have the same numbers.

Anneal schedule: We discuss the impact of the annealing schedule on enhancing the likelihood of obtaining feasible and optimal solutions. We employed multiple anneal schedules for the annealing to observe the solution quality. Optimal values of chain strength and QUBO penalty were chosen from the heatmap analysis for all problem cases. Different anneal schedules were chosen from shorter times to longer times. We plotted the percentage of feasible and optimal solutions for asymmetric and symmetric distributions. Each marker in Abb. 17 and correspond to the percentage of feasible solutions for 400 anneal reads. There are 50 such markers for each anneal schedule resulting in total of 20000 anneal reads. The feasible solutions for a-trp_6 has an inverted parabola type behavior, see Abb. 17. Rapid annealing disrupts the system's ability to maintain the ground state, leading to transitions to higher energy states associated with suboptimal or infeasible solutions. Conversely, longer anneal schedules beyond 600 µs also reduce the percentage of feasible solutions, possibly due to factors like thermal relaxation or qubit noise. Additionally, over a higher likelihood of feasible solutions but do not always guarantee maximal outcomes. We observe the same behavior in a-trp_7 but with a lesser intensity. For a-trp_8 the impact of the anneal schedule is not as significant as in the other cases. The inverted parabola behavior is less pronounced in optimal solutions.

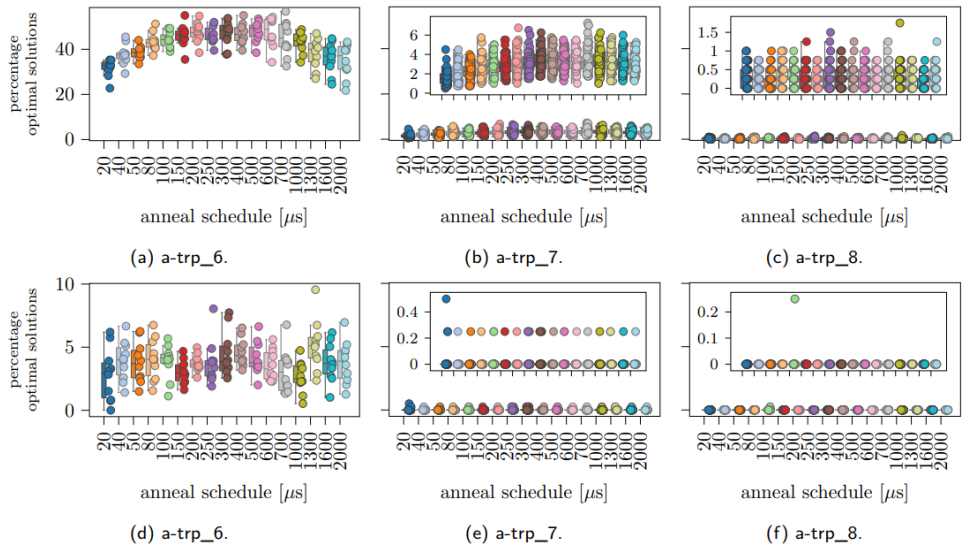


Abb. 17 Percentage of feasible and optimal solutions for asymmetrical distribution of cities with different anneal schedules. The chain strength value is 0.6 for a-trp_6 and 0.9 for a-trp_7 and a-trp_8. The number of anneal reads are 20000 for each anneal schedule. Clique embedding was used to embed the QUBO. The calculations were executed on Advantage 4.1 and results were obtained on 2023/11/15.

For the symmetric distribution we can see the inverted parabola behavior in symmetrical distribution of cities (Abb. 18). For all the problem sizes the feasible solutions peak at 300–400 µs and drop at shorter or longer anneal schedules. In contrast to the asymmetrical distribution, in the s-trp_7 and s-trp_8 cases the number of optimal

solutions were found more often in anneal schedules which are longer in duration. This behavior is possibly caused by scaling effects of the hardware coupling strengths calculated from the quadratic terms in the QUBO. In the case of a radial symmetric placement of cities, the difference between shortest and longest route is smaller than for an asymmetric placement. After scaling the maximum QUBO coefficient (corresponding to the longest possible route) to the maximum hardware coupling term, the symmetric TRP will result in a larger coupling term for the shortest route than in the asymmetric case. This has the effect that the energy scale of all hardware couplings is larger and thus we have a larger spacing between ground and excited states. A wider gap is directly connected to the success of staying in the ground state during annealing and finding a optimal solution. Therefore, symmetric distributions of cities have a higher percentage of optimal solutions.

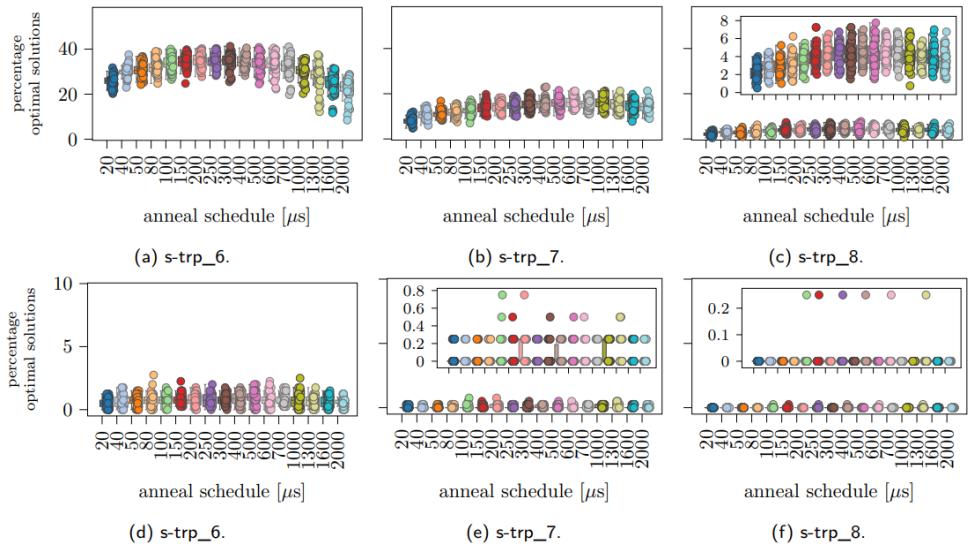


Abb. 18 Percentage of feasible and optimal solutions for symmetrical distribution of cities with different anneal schedules. The chain strength values of 0.6 is used for s-trp_6 and 0.9 for s-trp_7, and s-trp_8. The number of anneal reads are 20000 for each anneal schedule. Clique embedding was used to embed the QUBO. The calculations were executed on Advantage 4.1 and results were obtained on 2023/11/24.

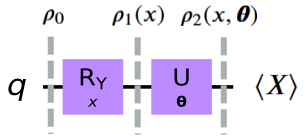
End-to-end demonstrators were developed on the code base (IBM and Dwave), with which the results can be reproduced.

Referenzen

[1] C. K. Tutschku, A. Sturm, F. Knäble, et al." Quantencomputing in der industriellen Applikation" 2023. doi: 10.24406/publica-805.
[2] A. Sturm, "Theory and Implementation of the Quantum Approximate Optimization Algorithm: A Comprehensible Introduction and Case Study Using Qiskit and IBM Quantum Computers" arXiv, Jan. 23, 2023. Available: https://arxiv.org/abs/2301.09535
[3] A. Ketterer, T. Wellens, "Characterizing crosstalk of superconducting transmon processors" arXiv, Mar. 24, 2023. Available: https://arxiv.org/abs/2303.14103
[4] A. Sturm, B. Mummaneni, L. Rullkötter, "Unlocking Quantum Optimization: A Use Case Study on NISQ Systems". arXiv, Apr. 10, 2024, Available: https://arxiv.org/abs/2404.07171

2.1.2 Quantenbasierte numerische Strömungssimulation

Für den Use Case „Prozesseffizienzsteigerung bei Tablettierung in der Pharmaindustrie“ wurde die Algorithmenklasse „Quantum Circuit Learning (QCL)“ [1], [2], mit klassischen Simulationen und auf dem IBM Quantencomputer in Ehningen analysiert und evaluiert. Insbesondere ihre Verwendung zur Lösung von Differentialgleichungen wurde ausgiebig untersucht.
In [3] und [4] werden die Ergebnisse detailliert vorgestellt und diskutiert, so dass wir hier nur einen Überblick zu den Ergebnissen daraus geben.
Ein Workshop zur Umsetzung des Use Case mit der Firma Pfizer hat stattgefunden. Zudem wurde ein End-to-End Demonstrator zu diesem Themengebiet erstellt.
Quantum Circuit Learning Schaltkreise beginnen mit einem Datenkodierungsblock. Hier wird eine Variable mit Quanten-Feature-Map-Kodierung in den Quantenzustand kodiert. Darauf folgt ein variationeller Block, der aus parametrisierten Quantengattern besteht. Zum Schluss, wird ein Erwartungswert gemessen. Der Erwartungswert beschreibt den Wert einer Funktion, die von der kodierten Variable und von den Parametern des variationellen Blocks abhängt. Diese Funktion bezeichnen wir als Quantum Model Function f_{QC}. Ein einfaches Beispiel für einen Quantum Circuit Learning Schaltkreis ist in Abb. 19 gegeben.



$$\rho_0 = \tfrac{1}{2}\left(\mathrm{I} + \mathrm{Z}\right)$$

$$\rho_1(x) = \mathrm{R_Y}(x)\rho_0\,\mathrm{R_Y}(x)^\dagger = \tfrac{1}{2}\left(\mathrm{I} + \sin x\,\mathrm{X} + \cos x\,\mathrm{Z}\right)$$

$$\begin{aligned}\rho_2(x,\boldsymbol{\theta}) &= U(\boldsymbol{\theta})\rho_1(x)U(\boldsymbol{\theta})^\dagger \\ &= \tfrac{1}{2}\left(I + a(x,\boldsymbol{\theta})\,\mathrm{X} + b(x,\boldsymbol{\theta})\,\mathrm{Y} + c(x,\boldsymbol{\theta})\,\mathrm{Z}\right)\end{aligned}$$

$$\langle X\rangle_{\rho_2(x,\boldsymbol{\theta})} = a(x,\boldsymbol{\theta}) = u_X(\boldsymbol{\theta})\sin x + u_Z(\boldsymbol{\theta})\cos x$$

Abb. 19 Simples Beispiel für ein Quantum-Circuit-Learning-Schaltkreis mit einem Qubit, einem Datenkodierungsblock und einem variationellen Block. Dieser Schaltkreis kann Funktionen der Art $f_{QC}(x) = \mu\sin x + \nu\cos x$ (mit $\mu^2 + \nu^2 \leq 1$) durch Anpassung der variationellen Parameter θ beschreiben

Dieser Schaltkreis kann Linearkombinationen aus $\sin x$ und $\cos x$ beschreiben. Werden mehrere Qubits verwendet, resultiert das Tensorprodukt der einzelnen Qubitzustände in Ansatzfunktionen höherer Ordnung. Ansatzfunktionen eines anderen Typs können über veränderte Datenkodierungsblöcke erhalten werden (z.B. führt die Verwendung einer $R_Y(\arcsin x)$-Kodierung zu polynomartigen Ansatzfunktionen). In Abb. 20 ist ein typischer Quantum Circuit Learning Schaltkreis mit $R_Y(\arcsin x)$-Kodierung abgebildet.

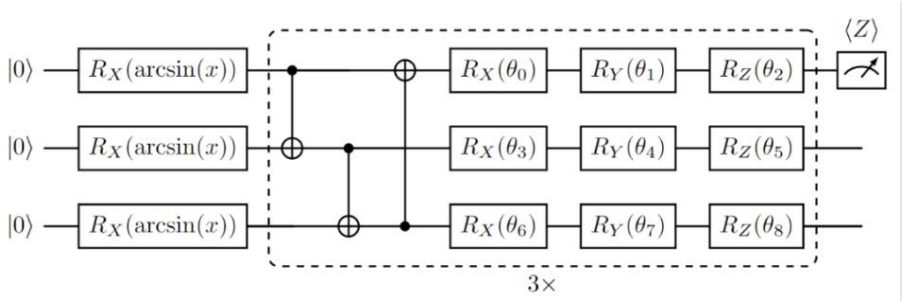


Abb. 20 Quantum-Circuit-Learning-Schaltkreis mit drei Qubits und $R_Y(\arcsin x)$-Kodierung. Der Z-Erwartungswert des ersten Qubits wird gemessen und definiert die Quantum Model Function $f_{QC}(x)$.

Eine Anwendung dieser Schaltungen ist die Approximation eindimensionaler Funktionen. Zu diesem Zweck werden die Parameter **θ** des variationellen Blocks mit einem klassischen Optimierer so bestimmt, dass die Quantum Model Function $f_{QC}(x)$ eine gegebene Funktion f(x) approximiert. Dazu wird eine Kostenfunktion definiert und an mehreren Punkten ausgewertet, die dann mit dem klassischen Optimierer minimiert wird.
Zu Beginn wurden die Schaltkreise mit Hilfe von klassischen Simulationen untersucht. Abb. 21 zeigt drei verschiedene Funktionen, die jeweils mit zwei unterschiedlichen Kostenfunktionen approximiert wurden (siehe [3] für die detaillierten Kostenfunktionen).

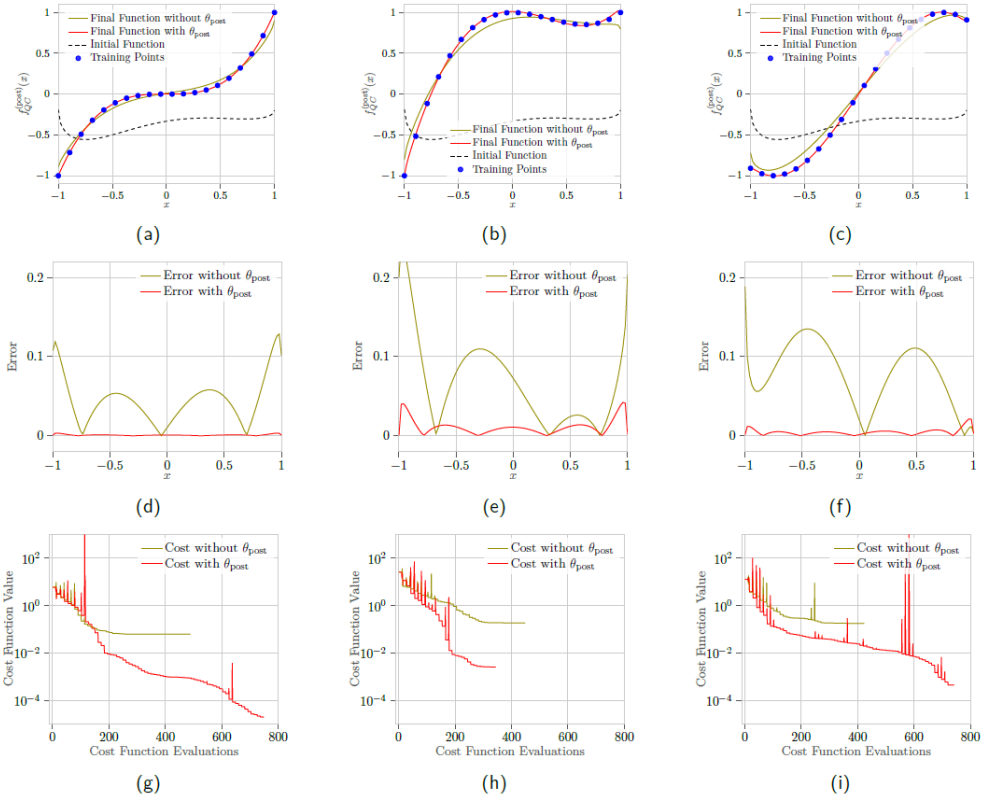


Abb. 21 Drei Funktionsapproximationen unter Verwendung von den QCL-Schaltkreisen in **Abb. 20** auf einem exakten Simulator mit (rote Linien) und ohne (grüne Linien) einen Nachbearbeitungsparameter θ_{post}. Die approximierten Funktionen sind $f_1(x) = x^3$ (a), $f_2(x) = x^3 - x^2 + 1$ (b) und $f_3(x) = \sin(2x)$ (c). Die Kostenfunktion wird für 20 äquidistante Trainingspunkte ausgewertet und klassisch mit SLSQP minimiert. Die Ausgangsfunktion (gestrichelte schwarze Linie) zeigt $f_{QC}(x)$ mit den zufällig gewählten Startparametern vor dem Optimierungsprozess. Zusätzlich sind in (d)-(f) die Absolutwerte der jeweiligen Fehler $|f(x) - f_{QC}(x)|$ dargestellt. In (g)-(i) sind die jeweiligen Werte der Kostenfunktion gegen die Anzahl der Kostenfunktionsauswertungen aufgetragen.

Es zeigt sich, dass es möglich ist verschiedene Funktionen zu approximieren. Zudem erhöht die Einführung eines Nachbearbeitungsparameters θ_{post}, der mit dem Erwartungswert multipliziert wird und auch klassisch optimiert wird, die Genauigkeit erheblich.
Daraufhin wurde die Ausführbarkeit auf dem IBM Quantencomputer in Ehningen erprobt. Die gleichen Schaltkreise wurden erneut verwendet, um dieselben Funktionen zu approximieren. Die Resultate lassen sich in Abb. 22 erkennen.

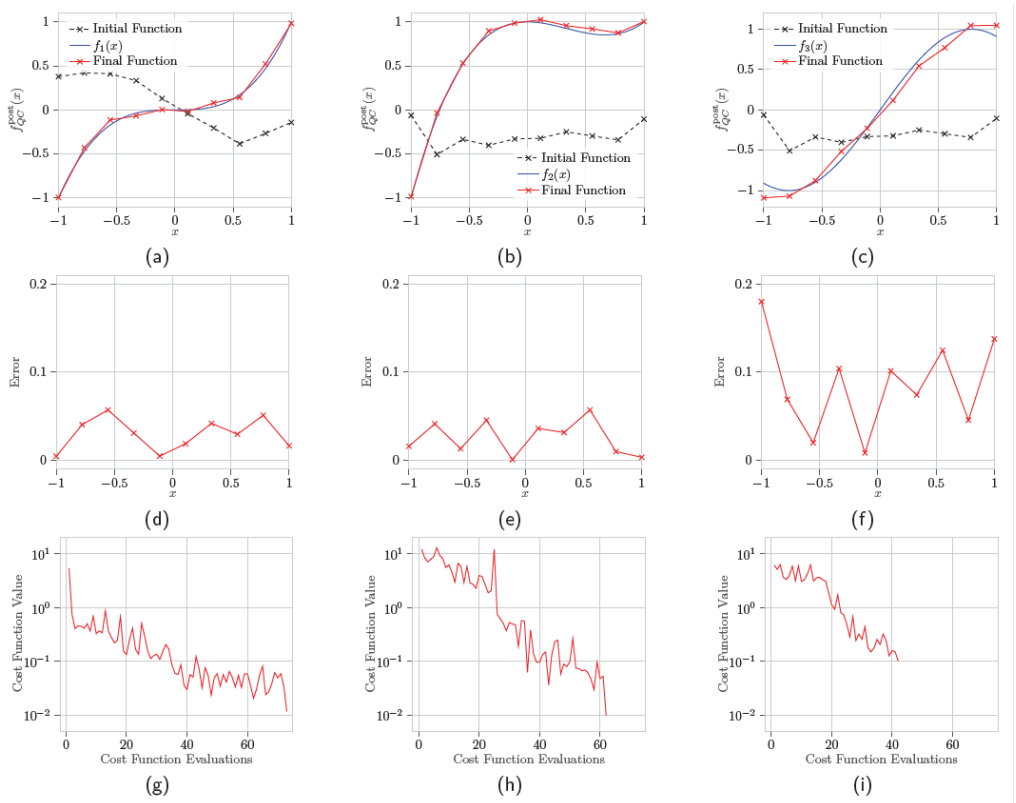


Abb. 22 Drei Funktionsapproximationen mit den QCL-Schaltungen in **Abb. 20** auf dem IBM Quantum System One Ehningen mit einem Nachbearbeitungsparameter θ_{post}. Die approximierten Funktionen sind $f_1(x) = x^3$ (a), $f_2(x) = x^3 - x^2 + 1$ (b) und $f_3(x) = \sin(2x)$ (c). Die Kostenfunktion wird an 10 äquidistanten Trainingspunkten mit jeweils 2000 Shots ausgewertet und klassisch mit COBYLA minimiert. Zusätzlich sind in (d)-(f) die Absolutwerte der jeweiligen Fehler $|f(x) - f_{QC}(x)|$ dargestellt. In (g)-(i) sind die jeweiligen Werte der Kostenfunktion in Abhängigkeit von der Anzahl der Kostenfunktionsauswertungen aufgetragen.

Die Funktionen können mit guter Genauigkeit auf dem realen Quantencomputer approximiert werden. Insbesondere die beiden Polynome können mit hoher Genauigkeit approximiert werden. Der Fehler, der am Ende überwiegt, ist hauptsächlich auf Shot-Noise zurückzuführen.
Aufbauend auf der Idee von Funktionsapproximationen kann QCL in Kombination mit der Parameter-Shift-Rule zur Lösung von Differentialgleichungen verwendet werden. Die Parameter-Shift-Rule ist ein Ansatz, um Gradienten einer parametrisierten Quantenschaltung zu erhalten [1, 5].
Um zu untersuchen, ob es realistisch ist Differentialgleichungen auf dem echten Quantencomputer zu lösen, wurde zuerst die Parameter-Shift-Rule auf dem IBM

Quantencomputer in Ehningen getestet. Die Ergebnisse für die erste und zweite Ableitung eines Beispiels sind in Abb. 23 zu erkennen.

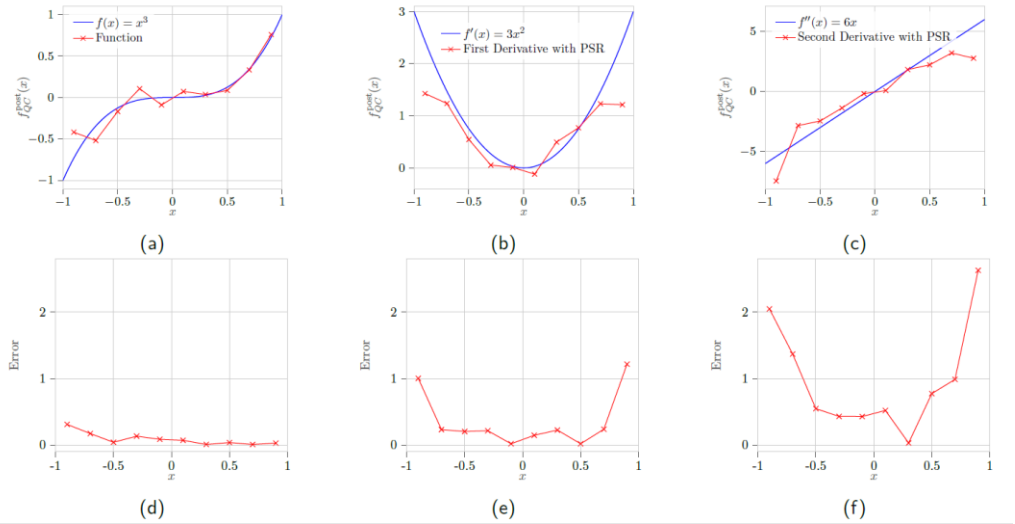


Abb. 23 Ableitungen, die mit der Parameter-Shift-Regel auf dem IBM Quantum System One in Ehningen mit 1024 Shots unter Verwendung der finalen Parameter aus dem ersten Beispiel in **Abb. 22** mit der Funktion f(x) = x^3 (a), der ersten Ableitung f'(x) = 3x^2 (b) und der zweiten Ableitung f''(x) = 6x (c) erhalten wurden. Zusätzlich, in (d)-(f) die Absolutwerte der jeweiligen Fehler |f (x) - f_{QC}(x)| dargestellt.

Das qualitative Verhalten der Ableitungen lässt sich gut bestimmen. Allerdings gibt es große Fehler, insbesondere in den Bereichen um x = -1 und x = 1, die weit über Shot-Noise hinausgehen. Die Fehler sind hauptsächlich auf Hardware-Fehler zurückzuführen, die sich aufgrund der hohen Anzahl von ausgewerteten Schaltungen aufsummieren.
Diese hohen Fehler sorgen dafür, dass das Lösen von Differentialgleichungen auf dem echten Quantencomputer schwierig ist. Wir wählen daher das sehr simple Beispiel u'(x) = 3x^2 mit u(0) = 0. Die Lösung auf dem IBM Quantencomputer in Ehningen lässt sich in Abb. 24 erkennen.

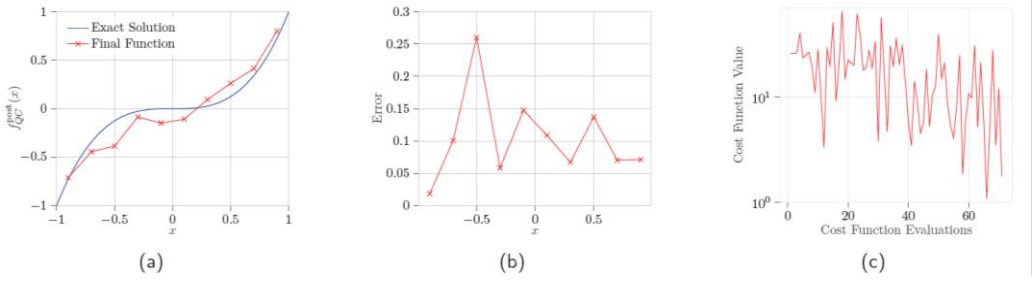


Abb. 24 (a) Lösung der Differentialgleichung u'(x) = 3x^2 mit u(0) = 0 auf dem IBM Quantum System One Ehningen mit 10 äquidistanten Trainingspunkten und 2000 Shots unter Verwendung von QCL Schaltungen mit mit RY(arcsin(x)) Kodierung. (b) Die absoluten Werte des Fehlers. (c) Werte der Kostenfunktion in Abhängigkeit von der Anzahl der Kostenfunktionsbewertungen.

Es zeigt sich, dass die Differentialgleichung auf dem realen Quantencomputer lösbar ist. Allerdings sind die Fehler hier deutlich höher als im Fall der Funktionsapproximationen in Abb. 22. Dies liegt daran, dass die Ableitungen in die Kostenfunktion einbezogen werden, was zu höheren Fehlern führt, wie in Abb. 23 zu sehen ist.
Zudem haben wir mit dieser Methode Differentialgleichungen mit Hilfe eines klassischen Simulators gelöst. Hier wählen wir als Beispiel einen gekoppelten harmonischen

Oszillator. Die genaue Gleichung und die Randbedingungen lassen sich in [3] finden und wir werden hier nicht im Detail darauf eingehen. Der gekoppelte harmonische Oszillator soll mit Hilfe des Schaltkreises in Abb. 25 gelöst werden.

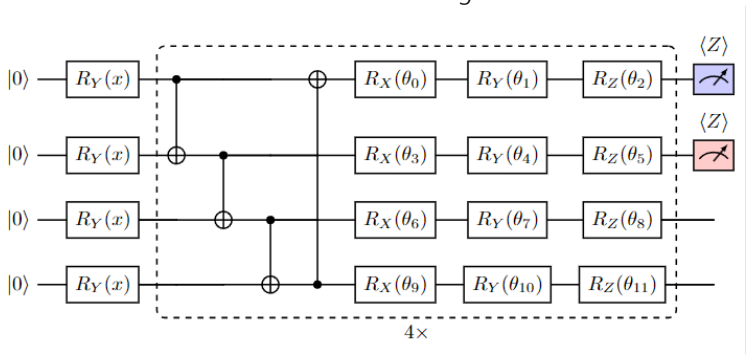


Abb. 25 Vier-Qubit-Schaltkreis mit $R_Y(x)$ Kodierung. Das erste und das zweite Qubit werden gemessen.

Der erste Term der gekoppelten Differentialgleichung wird mit dem ersten Qubit beschrieben und der zweite Term mit dem Zweiten. Dadurch kann die gekoppelte Differentialgleichung mit nur einem Schaltkreis gelöst werden. Das Ergebnis ist in Abb. 26 zu sehen.

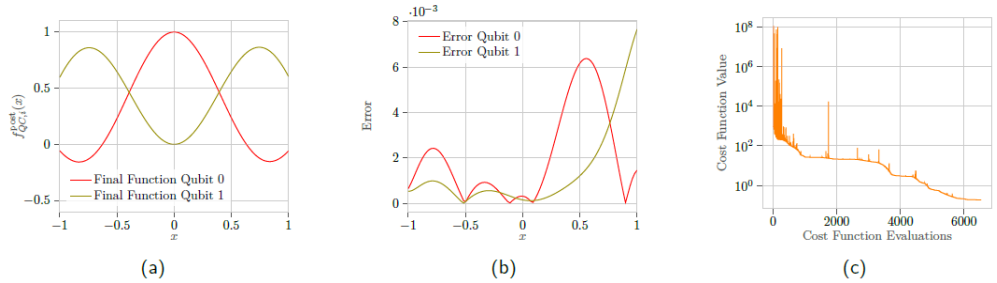


Abb. 26 (a) Lösung der Differentialgleichungen für einen gekoppelten harmonischen Oszillator (siehe [3]). Das Ergebnis wird mit der QCL-Schaltung in **Abb. 25** in Kombination mit der Parameter-Shift-Rule erzeugt. Die Parameter werden klassisch mit COBYLA optimiert. (b) Absolutwerte der jeweiligen Fehler $|f(x)-f_{QC}(x)|$. (c) Werte der Kostenfunktion in Abhängigkeit von der Anzahl der Kostenfunktionsauswertungen.

Referenzen

[1] K. Mitarai, M. Negoro, M. Kitagawa, et al. "Quantum circuit learning". In: Physical Review A 98.3 (2018). doi: 10.1103/PhysRevA.98.032309.

[2] Oleksandr Kyriienko, Annie E. Paine, and Vincent E. Elfving. "Solving nonlinear differential equations with differentiable quantum circuits". In: Physical Review A 103.5 (2021). doi: 10.1103/PhysRevA.103.052416

[3] Niclas Schillo, Andreas Sturm, "Quantum Circuit Learning on NISQ Hardware". arXiv, May 3, 2024, Available: https://arxiv.org/abs/2405.02069

[4] Niclas Schillo. "Quantum algorithms and quantum machine learning for differential equations" (2023).

[5] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. Physical Review A, 99, 2019. ISSN 2469-9926. doi: 10.1103/PhysRevA.99.032331.

Hamiltonian Simulation using Quantum Eigenvalue Transformation and Variational Block Encoding

The simulation of quantum mechanical systems is one of the most promising ways to deepen our understanding of nature and develop new technologies. Ranging from solid state research over particle physics to quantum chemistry, the ability to explore quantum phenomena in a simulated manner can fruit R&D in various ways. Classical simulations of these systems are ever improving and have been essential in science and industry. But in principle, emulating a quantum mechanical system on a classical computer is exponentially hard with a growing system size and entanglement. Current prominent threshold examples even for supercomputers are a sufficiently accurate simulation of high temperature superconductors or the process of nitrogen fixation in plants. It thus makes sense to try to map a quantum mechanical problem onto a quantum mechanical system which can be controlled and observed more easily – a Quantum computer. This problem has been at the front of a list of possible use-cases for quantum hardware since the first idea of such machines came up. The situation is rather uncomplicated. An initial quantum state should be evolved in time. Mathematically, this behaviour is described by the Schrödinger equation:

$$i\hbar \frac{d}{dt}|\Psi(t)\rangle = H|\Psi(t)\rangle,$$

which is a differential equation of first order and is governed by the system Hamiltonian H. For time independent Hamiltonians the operator that should be implemented on the quantum computer is the time evolution operator

$$U(t) = e^{-iHt} \quad \text{and it evolves the state as} \quad |\Psi(t)\rangle = e^{-iHt}|\Psi(0)\rangle.$$

Implementing $U(t)$ is the main task of Hamiltonian simulation on a quantum computer. This is arbitrarily hard, depending on the structure of the quantum mechanical system that makes up the Hamiltonian.

Historically, Trotterization is the most common way to decompose the exponential function into operators that can be implemented directly as a gate sequence. Recently, a new algorithm has been developed, improving the scaling of the simulation from polynomial in t to linear in t.

This algorithm uses a pulse sequence of parameterized rotations and encoding of the Hamiltonian to apply functions to each eigenvalue of the system. By choosing the right parameterization, an exponential function resembling the time evolution operator is executed in all eigenstate subspaces and the initial state is propagated in time as

$$e^{-iHt}|\Psi(0)\rangle = f(H)|\Psi(0)\rangle = \sum_i \alpha_i f(\lambda_i)|\lambda_i\rangle = \sum_i \alpha_i e^{-i\lambda_i t}|\lambda_i\rangle = |\Psi(t)\rangle.$$

Since the operation of the algorithm can be expressed in terms of each eigenvalue, it is called Quantum Eigenvalue Transformation.

The pulse sequence is described in Quantum Signal Processing and is a chain of operators encoding a Hamiltonian and interspersed, parameterized Rz rotations expressed as the QET operator

$$U(H) = e^{i\psi_0 Z}\prod_{i=0}^{N}\left(U_H e^{i\psi_i Z}\right) = \begin{pmatrix} f(H) & * \\ * & * \end{pmatrix}.$$

Abb. 27 displays the quantum circuit that is obtained from the implementation of the QET operator.

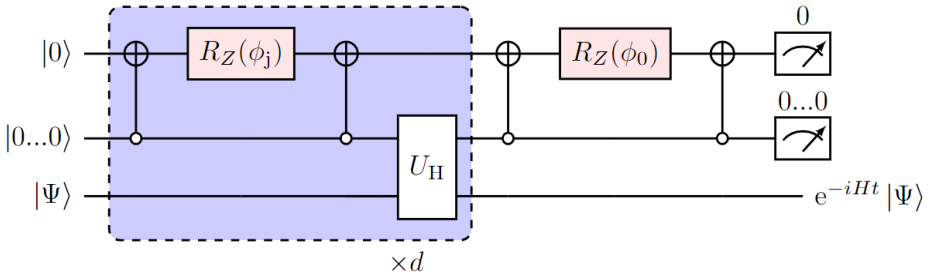


Abb. 27 Circuit diagram of the QET algorithm

A general Hamiltonian is not necessarily unitary, and its encoding is accompanied by expanding the Hilbert space by a specific amount of ancilla qubits that are found in the middle register. Together with this implementation of the multidimensional Rz rotation that adds one more ancilla, propagating the system register by the time evolution operator is only achieved when measuring the ancillas in the zero state. This is equivalent to applying the top left corner of the U-matrix onto the initial state. Rotation angles are calculated from a polynomial series expansion to the exponential function, in the case of Hamiltonian simulation, a Jacobi-Anger expansion. The number of iterations N is given by the polynomial degree of the approximation.

Even though the circuit looks easy, the main complexity is found in the operator U_H, that encodes the Hamiltonian. Just as U, it encodes the Hamiltonian in a subspace of the Hilbert space where a number of ancillas have been added, into the top left corner. Hence the name block-encoding. The additional ancillas are used to absorb the non-unitary part of H. In essence

$$H|\Psi\rangle = (\langle 0| \otimes \mathbb{I})U_H(|0\rangle \otimes |\Psi\rangle),$$

where the first register includes the ancilla qubits and the second register contains the system qubits.

Exact block encodings are for example based on encoding every matrix element of H using oracles or the linear combination of unitaries (LCU) that decomposes H into strings of Pauli operators and then combines the smaller operators. The later method uses arbitrary state preparation to encode the coefficients of the Pauli strings and several multi-qubit controls related to the number of Pauli strings.

Implementing such operators in the NISQ-era is almost impossible even for small systems. A solution is to approximate the block encoding circuit. One of these methods consists of a variational circuit that can be built and optimized to the appropriate degree of approximation in the QET algorithm. We found that the two-qubit gate count and circuit depth can be heavily reduced compared to exact methods while still maintaining a sufficient accuracy.

The variational block encoding (VBE) circuits are optimized with the cost function:

$$C(\vec{\theta}) = \left\| H_{VBE}(\vec{\theta}) - H \right\|_F,$$

that uses the Frobenius norm of the exact Hamiltonian and the Hamiltonian from the VBE. Up to now the variational circuits are optimized without noise, so that gradient-based classical optimizers like SLSQP perform very well in updating the parameters.

Variational circuits display a high amount of freedom in their design. For the implementation on a NISQ quantum computer, the most straight-forward option are hardware efficient circuits using only the hardware specific gate set.

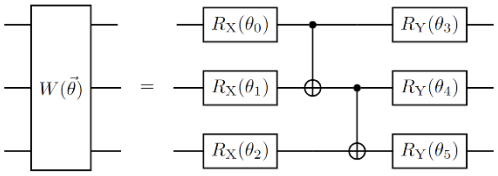


Abb. 28 Ansatz circuit for a 2-qubit system

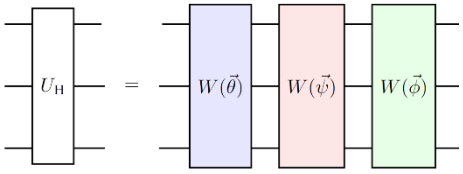


Abb. 29 Variational circuit for U_H

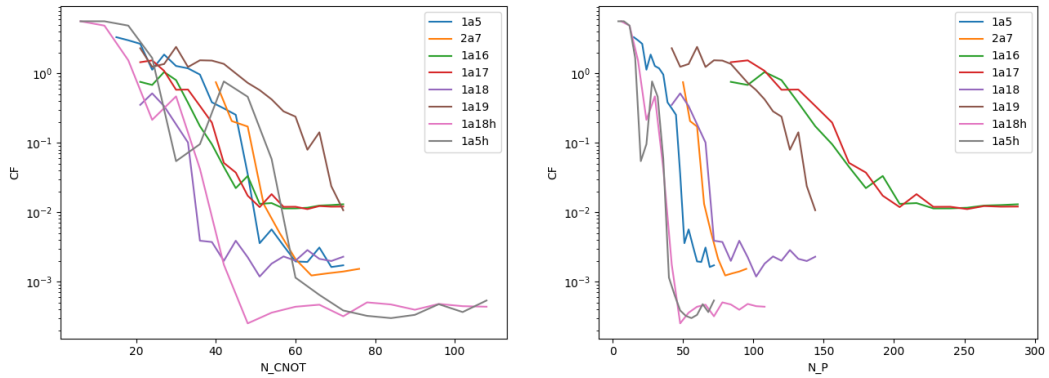


Abb. 30 Accuracy of Ansatz related to the number of CNOT gates (left) and number of parameters (right).

An example is shown in Abb. 28 and this ansatz can then be iterated several times as shown in Abb. 29. This enlarges the expressability and entanglement capabilities. Different ratios of single to multi-qubit gates are used to very both quantities. It was observed that real Hamiltonians can be very efficiently approximated by only using Ry rotations and consequently staying in the ZX-plane of the Bloch sphere. Additional degrees of freedom are the amount of ancillas, compared to a fixed amount in the exact LCU encoding and if the variational circuit is either Hermitian or non-Hermitian.

Abb. 30 displays results for a VBE of a three site transverse field Ising Hamiltonian. Different Ansatz circuits were optimized with a different amount of ancillas and layers. These results are still preliminary, but the show that ansatz circuits behave very distinct in their ability to encode the Hamilitian to a cirtain degree, particulary hermitian ansätze show good results with a low CNOT and parameter count. As a reference, the CNOT count of an LCU ansatz for this particular system lays at 279.

Further research will provide data as to which end a specific ansatz is suited for a specific type of Hamiltonian. Variational circuits could be very useful in approximating chemical Hamiltonians, which are often very unstructured. Furthermore, one goal is to compare

how well variational circuits work for different hardware architectures. Especially on Rydberg platforms, that have a triangular/square qubit connectivity and a CCZ gate.
There are various possible uses for VBEs. Block encoding operators can be used for Quantum Phase estimation, eliminating the need for a full Hamiltonian simulation algorithm in the controlled phase operators. Apart from that, Quantum Signal Processing opens the door to implement various matrix functions, for example the inverse. This can be used to solve linear system problems and discretised differential equations.
In conclusion, VBEs can aid in the implementation of small scale QET algorithms on NISQ hardware.

2.1.3 Kostenoptimierung und Auslegung von Fertigungsstraßen

Im Anwendungsfall „Fertigungsstraßen“ wurde ein sog. “Assembly Line Balancing Problem” untersucht. Dabei handelt es sich um eine in der Produktion häufig auftretende Optimierungsaufgabe. Ziel der Optimierung ist es eine Anzahl an Arbeitsplätzen möglichst günstig mit verfügbaren Maschinen zu bestücken, um Aufgaben zu bearbeiten. Die Aufgaben müssen in richtiger Reihenfolge und unter Einhalten einer Taktzeit abgearbeitet werden. Es sind verschiedene Maschinen verfügbar, welche gewisse Aufgaben in gegebenen Zeiten bearbeiten können. Neben der Lösung des Problems selbst, sollte anhand des Anwendungsfalls prototypisch beleuchtet werden, inwiefern heutige Methoden und Systeme praktikabel zum Lösen kombinatorischer Optimierungsprobleme einsetzbar sind. Hierbei sollte insbesondere die Skalierung hin zu großen Probleminstanzen untersucht und mit konventionellen Verfahren verglichen werden. Die Umsetzung des Anwendungsfalls in dessen kleinster Form wird in diesem Kapitel erläutert. Die Ergebnisse zur Skalierung und Praktikabilität sind in der Dokumentation zu AP 4.3 “Analyse von QUBO-Kapazitäten und Bewertung von Testumgebungen” zu finden.

(**1**)	$\mathbf{min}_{y\in\{0,1\}} \sum_{j=1}^{r}\sum_{k=1}^{m} EC_j \cdot y_{jk}\,,\ s.t.$
(2)	$\sum_{j=1}^{r}\sum_{k=1}^{m} z_{ijk} = 1\ \forall i$
(3)	$\sum_{i=1}^{n} t_{ij} \cdot z_{ijk} \leq ct \cdot y_{jk}\ \ \forall jk$
(4)	$\sum_{i=1}^{n}\sum_{j=1}^{r} t_{ij} \cdot z_{ijk} \leq ct\ \forall k$
(5)	$\sum_{j=1}^{r}\sum_{k=1}^{m} k \cdot z_{gjk} \leq \sum_{j=1}^{r}\sum_{l=1}^{m} l \cdot z_{hjl}\ \forall (g \quad h) \in P$

Tabelle 2 Formulierung des Assembly Line Balancing Problems aus [1] als lineares Optimierungsproblem mit Gleichheits- und Ungleichheitsnebenbedingungen. (1) formuliert die Kosten zum Einrichten der Fertigungsstraße. (2) stellt sicher, dass jede Aufgabe nur einmal bearbeitet wird. (3) stellt sicher, dass die Taktzeit an jeder zugeordneten Maschine eingehalten wird. (4) stellt sicher, dass die Taktzeit an jedem Arbeitsplatz eingehalten wird. (5) stellt sicher, dass die Reihenfolge der Aufgaben eingehalten wird.

Tabelle 2 zeigt die Formulierung als lineares Optimierungsproblem mit Nebenbedingungen analog zu [1]. Das resultierende kombinatorische Optimierungsproblem ist NP-Schwer und somit mit klassischen Lösungsverfahren bei großen Instanzen nur schwer lösbar.

In AP 1.1 wurde die mathematische Formulierung als lineares Optimierungsproblem mit Nebenbedingungen in Tabelle 2 in ein quadratisches, unrestringiertes, binäres Optimierungsproblem (*quadratic unconstrained binary optimization* – QUBO)

umgeformt. Die lineare Kostenfunktion kann für den binären Fall leicht in eine quadratische Formulierung umgewandelt werden, da für binäre Variablen die Gleichheit

$$x = x^2 \ (x \in \{0, 1\})$$

gilt. Gleichheitsnebenbedingungen können durch die Umformulierung

$$\min_{x \in \{0,1\}^n, Ax=b} x^T C x = \min_{x \in \{0,1\}^n} x^T C x + \lambda \cdot (Ax - b)^2$$

unter Hinzunahme von Lagrange-Parametern λ zur Kostenfunktion hinzugefügt werden. Hier werden insgesamt vier Lagrange-Parameter hinzugefügt.
Ungleichheitsnebenbedingungen werden durch die Verwendung von Schlupfvariablen in Gleichheitsnebenbedingungen umgeformt.

$$Ax \leq b \Leftrightarrow Ax + s = b, \qquad s > 0$$

Für die automatisierte Überführung des linearen Optimierungsproblems in eine QUBO-Formulierung wurde eine Python Bibliothek entwickelt und implementiert. Die Bibliothek erlaubt die Formulierung beliebiger Probleminstanzen auch über den Rahmen des Projekts hinaus. Die Formulierung als QUBO kann einfach in eine quantenmechanische Formulierung als Ising-Modell übertragen werden und dann mit gängigen Methoden wie QAOA oder Quantum Annealing [4, 6] gelöst werden. Die Bibliothek ist auf GitLab frei verfügbar (https://gitlab.cc-asp.fraunhofer.de/ipa-quantum/alb-qubo).

Die Verwendung der Schlupfvariablen in der Formulierung als QUBO kann bereits bei sehr kleinen Probleminstanzen zu großen QUBOs führen. In AP 2.1 wurde deswegen ein hybrider Ansatz namens „QBSolv" [2] als passender Quantenalgorithmus identifiziert. Der Solver verwendet eine Kombination aus Partitionierung der QUBOs und dem heuristischen Suchalgorithmus Tabu-Suche zum Lösen der großen QUBO-Instanzen. Die maximale Größer der Subsysteme wird auf 20 Entscheidungsvariablen festgelegt. Zum Lösen der kleinen, partitionierten Sub-QUBOs kann ein beliebiger Annealing-Algorithmus verwendet werden. Hier wurde ein DWAVE Quantum Annealer [5] verwendet. Dieser Ansatz erlaubt es auch große Probleminstanzen zu lösen.
Weiterhin fügt die Umformulierung als QUBO dem Problem vier zusätzliche (Hyper-) Parameter hinzu. Die Wahl der Lagrange-Parameter ist nicht trivial und beeinflusst die Gewichtung der jeweiligen Nebenbedingungen auf die zu optimierende Kostenfunktion. Die Parameter werden mittels einer Rastersuche und der Simulated-Annealing Implementierung in Neal [3] gelöst. Hierbei wird überprüft welche Lagrange-Parameter-Kombination aus 1000 Samples die meisten optimalen Lösungen produziert. Als optimale Lösung werden jene gewertet, die alle Nebenbedingungen erfüllen und dabei die niedrig-möglichsten Kosten zum Einrichten der Fertigungsstraße verursachen. Beim Lösen mit Simulated Annealing wird auf die Verwendung von QBSolv verzichtet. Dies ist AP 1.2 zuzuordnen.

Hier wird im Folgenden exemplarisch die Lösung einer kleinen Beispielinstanz mit vier Aufgaben, zwei Maschinen und zwei Arbeitsplätzen vorgestellt. Weitere Instanzen und deren Lösungen sind in AP 4.3 aufgeführt. Die hier dargestellte Instanz ist Beispielinstanz 1 in AP 4.3 und umfasst 64 Qubits in der QUBO-Formulierung.

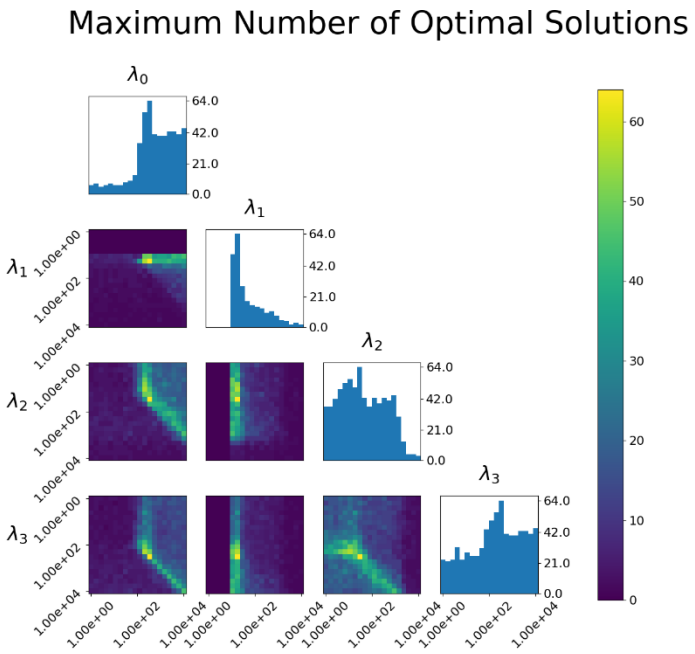


Abb. 31 Anzahl der Optimalen Lösungen für jeweils fixierte Kombinationen der Lagrange Parameter für Beispiel 1 aus 1000 Samples mit Neal. Die Plots auf der Diagonale zeigen jeweils die meisten möglichen optimalen Lösungen für einen fixierten Lagrange Parameter. Die Plots auf der unteren Dreiecksmatrix zeigen jeweils fixierte Kombinationen aus zwei Lagrange Parametern.

Abb. 31 zeigt, exemplarisch für Beispiel 1, den Suchraum und die jeweiligen Ergebnisse der verwendeten Lagrange Parameter Kombinationen.
Mit dem oben beschriebenen Verfahren mit QBSolv und dem DWAVE Quantum Annealer wurden jeweils 1000 Samples produziert. Unter diesen Samples finden sich viele Lösungen, welche die Nebenbedingungen erfüllen, und auch einige optimale Lösungen. Abb. 32 zeigt die Anzahl sowie die Kosten zur Einrichtung der Fertigungsstraße der 50 am häufigsten vorkommenden Lösungen.

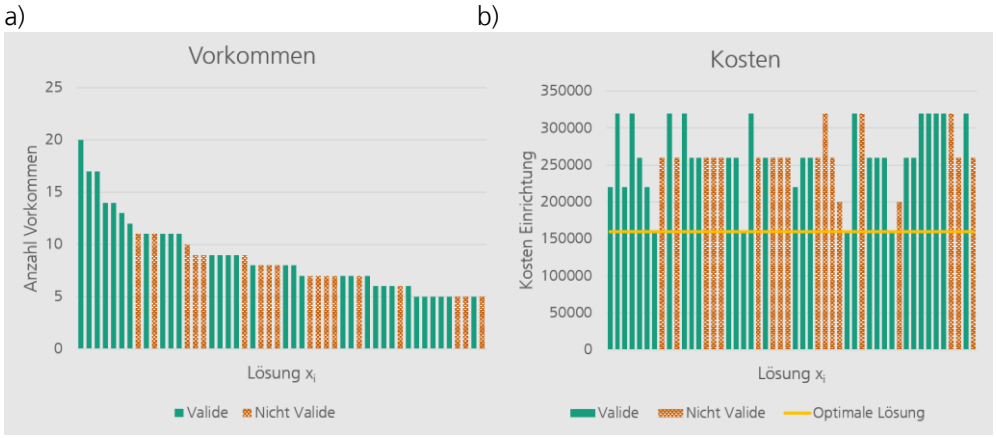


Abb. 32 Top 50 Ergebnisse von 1000 Samples von QBSolv mit DWAVE Quantum Annealer, angewendet auf Beispielinstanz 1. Beide Plots zeigen, ob eine Lösung die Nebenbedingungen erfüllt oder nicht erfüllt. a) zeigt die Häufigkeit der jeweiligen Lösungen, b) zeigt die Werte der ursprünglichen Kostenfunktion aus [1] der jeweiligen Lösungen. Außerdem ist der Wert der bestmöglichen Lösung markiert.

Die Ergebnisse zeigen, dass man mit diesem Lösungsverfahren für kleine Probleminstanzen gute Lösungen finden kann. Es ist eine ausreichende Menge valider

Lösungen unter den wahrscheinlichsten Ergebnissen zu finden. Die optimale Lösung wird ebenfalls unter den ersten 20 wahrscheinlichsten Lösungen zu finden. Da die Evaluierung der Kostenfunktion grundsätzlich günstig ist und die Schwierigkeit durch die exponentielle Anzahl an möglichen Zuständen entsteht, besteht die Lösungsstrategie, die $m \ll 2^M$ wahrscheinlichsten Zustände zu prüfen, um so die optimale Lösung zu finden. Hier ist M die Anzahl an Entscheidungsvariablen des ursprünglichen Problems. Es ist jedoch anzumerken, dass ein erheblicher klassischer Aufwand betrieben werden musste, um eine gute Kombination aus Lagrange Parametern zu finden. Diese Strategie setzt eine grundsätzlich gutartige Optimierungslandschaft und die Möglichkeit zum Auffinden valider und optimaler Lösungen mit hinreichender Wahrscheinlichkeit voraus. Diese Voraussetzung ist für größere Probleminstanzen nicht zwangsweise erfüllt, wie in den Ergebnissen zu AP 4.3 zu sehen ist.
Es wäre wünschenswert gewesen, die Lösungsgüte des oben beschriebenen Annealing-Ansatzes mit digitalen Algorithmen wie z.B. QAOA zu vergleichen. Die Problemgröße des Minimalbeispiels ist mit 64 Qubits jenseits der klassischen Simulierbarkeit. Versuche das Problem direkt mit QAOA auf 127-Qubit IBM Systemen zu lösen schlugen leider fehl. Probleme entstehen hierbei dadurch, dass die hardwarebedingten Fehler deutlich mit der Anzahl an verwendeten Qubits wächst (z.B. durch Cross-Talk und Frequency Crowding). Die derzeitig verfügbaren supraleitenden Systeme von IBM bieten zwar eine ausreichende Menge an Qubits, unsere Erfahrungen mit größerem Problem zeigen allerdings, dass trotz der nominell hohen Qubit-Anzahl Probleme, bei denen eine Größenordnung von mehr ~20 gemeinsam genutzten Qubits aufgrund der oben beschriebenen Fehler nicht praktikabel sind. Weiterhin zeigt sich im Fall des QAOA-Algorithmus, dass die klassische Optimierung (finden der QAOA Parameter) mit steigender Qubit Anzahl deutlich erschwert wird. Bei größeren Problemen entstehen daher nicht nur hardwarebedingte Probleme, sondern auch Schwierigkeiten in der Bewältigung der Optimierungslandschaft. Neben einer Reduzierung des Hardwarerauschens sind deshalb auch algorithmische Fortschritte nötig, um die Anwendung von QAOA für ein Problem dieser Größe praktikabel zu machen. Abb. 33 zeigt die Werte der ursprünglichen Kostenfunktion für die 50 häufigsten Lösungen einer Probleminstanz. Die Optimierung konnte nicht bis hin zur Konvergenz durchgeführt werden. Daher konnte mit unter 5.000 Stichproben keine valide Lösung gefunden werden. Wir stellen daher fest, dass das Assembly Line Balancing Problem mit dieser Größenordnung auf derzeitigen IBM Systemen mit nicht zufriedenstellend lösbar sind. Das deckt sich mit den Ergebnissen anderer Anwendungsfälle in diesem Projekt (siehe Anwendungsfall "Routenplanung von LKW-Flotten im Supply Chain Management").

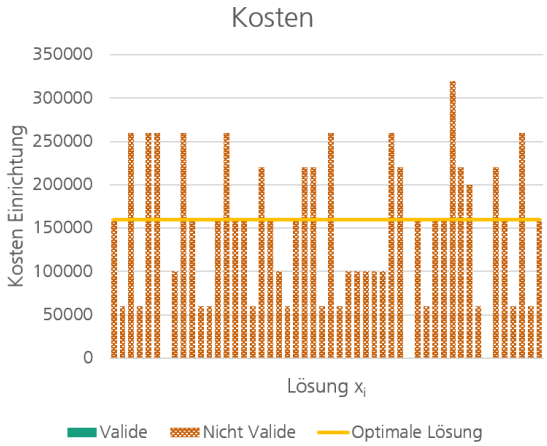


Abb. 33 Werte der ursprünglichen Kostenfunktion für die 50 wahrscheinlichsten Lösungen des Assembly Line Balancing Problem mittels QAOA auf dem IBM Quantencomputer Kyoto. Insgesamt wurde keine valide Lösung gefunden

Verwertung

Die Python Bibliothek zur Formulierung des Assembly Line Balancing Problem als QUBO wird auf GitLab (https://gitlab.cc-asp.fraunhofer.de/ipa-quantum/alb-qubo) öffentlich zugänglich gemacht und kann über den Fokus des Projekts hinaus für beliebige Instanzen verwendet werden. Außerdem wurde ein Demonstrator zur Veranschaulichung des verwendeten Lösungsverfahren anhand des hier verwendeten Beispiel 1 veröffentlicht. Weiterhin wurde der Anwendungsfall in verschiedenen Vorträgen, u.a. in der Webinar Reihe "Quantum Brunch" vom 26.01.2024 vorgestellt.
(https://www.ipa.fraunhofer.de/de/veranstaltungen-messen/veranstaltungen/2023/quantum_brunch.html).

Referenzen

[1] Albus, M. & Seeber, C. 2021. Linear optimization for dynamic selection of resources in constrained assembly line balancing problems. *Procedia CIRP* 104, S. 134–139
[2] Booth, M., Reinhardt, S.P., & Roy, A. 2017. Partitioning Optimization Problems for Hybrid Classical/Quantum Execution TECHNICAL REPORT.
[3] D-Wave Systems Inc. (2022). neal (Version 0.6.0) [Software]. Available from https://github.com/dwavesystems/dwave-neal
[4] Finnila, A.B., Gomez, M.A., Sebenik, C., Stenson, C. & Doll, J.D. 1993 Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics letters* 219, S. 343-348
[5] Johnson, M., Amin, M., Gildert, S. *et al.* 2011 Quantum annealing with manufactured spins. *Nature* 473, S. 194–198
[6] Kadowaki, T. & Nishimori, H. 1998 Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58, S. 5355-5363

2.1.4 Resilienzanalyse kritischer Infrastrukturnetze

Im Rahmen der Entwicklung eines Quantenalgorithmus zur Resilienzsteigerung von Infrastrukturnetzen wurde ausgehend von einem Knoten und Kanten-Modell zur Darstellung von voneinander abhängigen kritischen Infrastrukturnetzen (klassische Anwendung durch ein C++ basiertes Netzwerk-Simulationstool) ein stark vereinfachtes Netzwerk aufgestellt, das die spezifische Eigenschaft von Kaskadenentwicklungen abbildet.

Im Projekt EFFEKTIF wurde bereits begonnen diese Systeme zu untersuchen, wobei jedoch nur Netzwerke ohne Schädigungen oder einzelne Knoten mit Schädigung und Wiederherstellung simuliert werden konnten. Das größte Netzwerk, welches derzeit mit den Methoden aus AP 2.4 umgesetzt werden kann, besteht aus drei Knoten, die in einer Kette angeordnet sind und eine Schädigung, sowie die Wiederherstellung, eines Knotens erlauben.

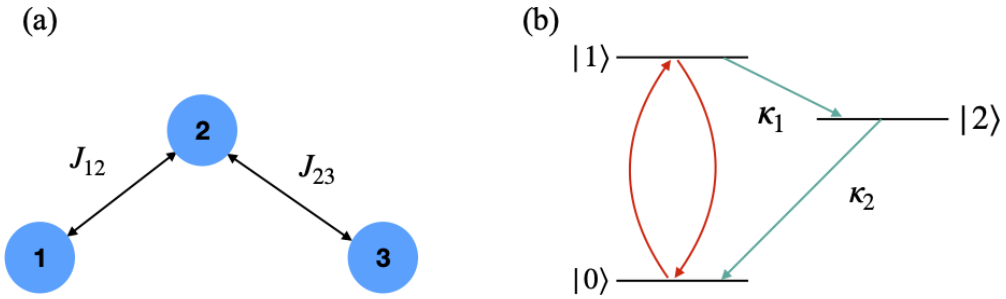


Abb. 34 (a) Three-node network whose nodes 1,2 and 2,3 are connected by links with strengths J_{12} and J_{23}, respectively. (b) The nodes are implemented as two-level (nodes 2 and 3) or three-level quantum systems (node

1), with levels |1⟩ and |2⟩ being qubit states that are coherently coupled by external driving with strength Ω. Level |3⟩ of node 1 models the network's defect state, which can be populated via an incoherent transition ("disruptive event") from level |2⟩ , with rate κ_1. Another incoherent transition |3⟩ → |1⟩ ("repair process"), with rate κ_2, returns the node to the qubit subspace.

The nodes of the network are coupled by interactions of dipolar type with strengths J_{12} and J_{23} (see Abb. 34 (a)). These interactions connect qubit levels |0⟩ and |1⟩ of distinct nodes and mediate a coherent excitation exchange. In addition to the qubit levels, nodes include level |2⟩ (Abb. 34 (b)), which models nodes' damage and repair, via incoherent processes described by rates κ_1 and κ_2, assumed to be the same for each node.

The incoherent processes indicate that the network is an open quantum system undergoing a nonunitary evolution. Under the standard approximations [1], the density operator ρ of the network is governed by the following Markovian master equation:

$$\dot{\rho}=-i[H,\rho]-\sum_{\alpha=1}^{3}\sum_{m=1}^{2}\left(L_{\alpha m}^{\dagger}L_{\alpha m}\rho+\rho L_{\alpha m}^{\dagger}L_{\alpha m}\rho-2L_{\alpha m}\rho L_{\alpha m}^{\dagger}\right), \quad (1)$$

where $H=H_F+H_{Int}$ is the Hamiltonian operator describing the coherent field driving and the dipolar interactions, while $L_{\alpha m}$ and $L_{\alpha m}^{\dagger}$ are the Lindblad operators and their Hermitian adjoints. For the Hamiltonian H_F, we considered two cases: (i) pulsed laser field, and (ii) continuous wave laser field. For the case (i), we took the area of π for a pulse applied to a single node of the network (case (ii) is considered in AP 2.4). This amounts to setting the driven and undriven nodes to be initially in states |1⟩ and |0⟩, respectively. The dipolar interactions are given by the Hamiltonian

$$H_{Int}=\sum_{\alpha=1}^{2}J_{\alpha,\alpha+1}(\sigma_{01}^{\alpha}\sigma_{10}^{\alpha+1}+\sigma_{10}^{\alpha}\sigma_{01}^{\alpha+1}),$$

where $\sigma_{ij}^{\alpha}=|i\rangle_{\alpha}\langle j|_{\alpha}$. Finally, the Lindblad operators are given by $L_{\alpha 1}=\sqrt{\kappa_1}\sigma_{21}^{\alpha}$, $L_{\alpha 2}=\sqrt{\kappa_2}\sigma_{02}^{\alpha}$.

A general state of such three-node quantum network can be described by a density operator $\rho(t)\in\mathcal{H}_1\otimes\mathcal{H}_2\otimes\mathcal{H}_3$, with $\mathcal{H}_n$ being the Hilbert space of node n , spanned by vectors $\{|0\rangle,|1\rangle,|2\rangle\}_{\alpha}$; thus, $\dim\rho(t)=3^3=27$.

Mit Hilfe des ibmq_qasm_simulator konnte gezeigt werden, dass ein solches System auf einem Quantencomputer berechnet werden kann. Bei Benutzung des Simulators ohne ein künstliches Rausch-Modell werden die analytischen Ergebnisse, wie in Abb. 35 zu sehen, sehr gut reproduziert, was die Methode auf theoretischer Ebene verifiziert.

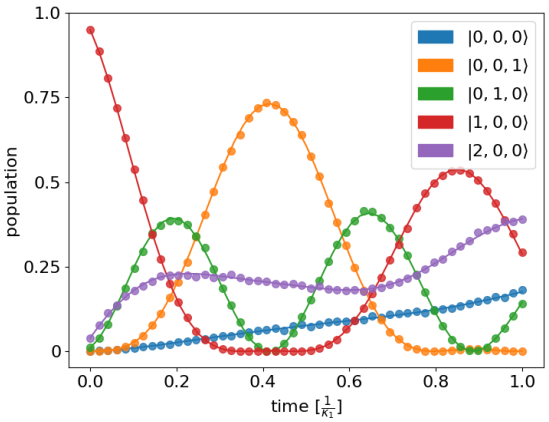


Abb. 35 Vergleich der Zeitentwicklungen von verschiedenen Zuständen im Drei-Knoten System des ibmq_qasm_simulator (Punkte) mit der analytischen Lösung mit Mathematica (durchgezogene Linien).

Diese Ergebnisse konnten auf der Quanten-Hardware von IBM nicht erzeugt werden. Stattdessen führt dieses System zu Quantenschaltkreisen, die so lang sind und so viele CNOT-Gatter enthalten, dass die Populationen aller Zustände einer nahezu zeitlich konstanten Gleichverteilung entsprechen und nicht dem qualitativen simulierten Verlauf folgen. Daher wurde die Größe des Systems auf zwei Knoten, von denen einer einen Defekt-Level besitzt, reduziert. In diesem System lassen sich bereits interessante dynamische Effekte beobachten, wobei die Fehler auf der Quanten-hardware von IBM moderat bleiben. In Abb. 36 sind verschiedene Zeitskalen zu beobachten. Zum einen gibt es den Zerfall des angeregten Zustands in den Defekt-Level, den man durch den Anstieg der Population des Zustands $|2,0\rangle$ und Abfall der Maxima der beiden angeregten Zustände $|0,1\rangle$ und $|1,0\rangle$ erkennt. Außerdem zerfällt der Defekt-Zustand in den Grundzustand und die Anregung wechselt zwischen den Knoten hin und her. Speziell bei letzterem Vorgang wird die Frequenz auf dem IBM Q System One sehr präzise reproduziert.

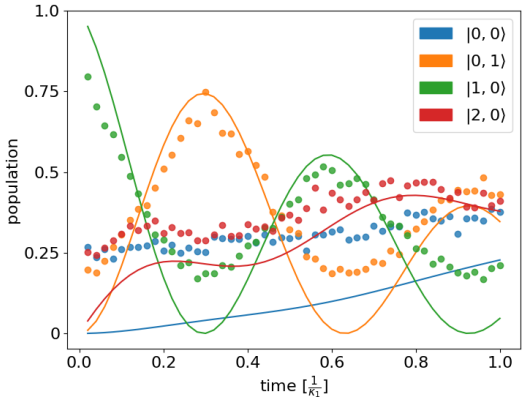


Abb. 36 Vergleich der Zeitentwicklungen von verschiedenen Zuständen im Zwei-Knoten System des IBM Q System One (Punkte) mit der analytischen Lösung mit Mathematica (durchgezogene Linien).

Zusätzlich zur Weiterentwicklung des Ansatzes aus EFFEKTIF wurde als neuer Ansatz ein Mapping auf ein kleines Quantenmodell durchgeführt, das mit bekannten Quanten-Optimierungsalgorithmen gelöst werden kann.
Ausgangspunkt ist auch hier das oben erwähnte Knoten und Kanten-Modell zur Darstellung von voneinander abhängigen kritischen Infrastrukturnetzen, das seine klassische Umsetzung in einem C++ basierten Netzwerk-Simulationstool findet. Der Ausfall einer kritischen Infrastrukturkomponente – im Modell abgebildet als Knoten –

führt mit einer gewissen Wahrscheinlichkeit zu einem Ausfall anderer angebundener Komponenten. Die Performanz des Systems wird aus der Menge der funktionierenden Knoten im Netz abgeleitet. Ausgeprägte Ausfallkaskaden führen zu einem starken Performanzverlust des Infrastrukturnetzes. Im klassischen Netzwerk-Simulationstool wird die zeitliche Systemperformanz unter verschiedenen Störungsereignissen durch die Monte-Carlo Methode ermittelt.

In AP 2.4 wurde auf Basis von [2] ein Quanten-Algorithmus entwickelt, der eine Netzwerkanalyse bezüglich der Netzkomponente oder Verbindung ermöglicht, deren Ausfall den größten Einfluss auf die Performanz des Netzwerkes hat. Dazu wurde der klassische Ansatz der Monte-Carlo Simulation durch eine „Quantum Amplitude Estimation" (QAE) ersetzt. Zur Suche nach der einflussreichsten Netzwerkkomponente wurde eine Grover-Suche durchgeführt. Weitere Details zum Quantenalgorithmus sind in Abschnitt 2.2.4 beschrieben.
Die Anzahl der Knoten und Kanten sind systemspezifisch und bestimmen mit ihrer Anzahl und den Übertragungswahrscheinlichkeiten die Komplexität des abgebildeten kritischen Infrastrukturnetzes. Im Falle dieses QC-Ansatzes sind sie auf eine einstellige Zahl (6 Komponenten und ein „Threat"-Knoten) reduziert. Ein untersuchtes Testnetz ist in Abb. 37 dargestellt.

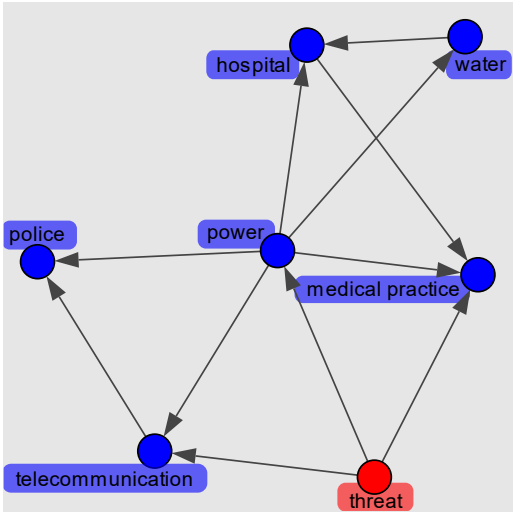


Abb. 37 Verwendetes Beispielnetz aus voneinander abhängigen kritischen Infrastrukturkomponenten, das auf seine einflussreichste Komponente (Knoten oder Kante) hin untersucht wird

Die Ausführung des Quantenalgorithmus auf QC-Simulatoren (ibmq_qasm_simulator) identifiziert für eine gegebene Netzstruktur und ein spezifisches Bedrohungsszenario, abhängig von den Wahrscheinlichkeitsparametern, jenen Parameter mit dem größten Einfluss auf die Gesamtperformanz des Netztes. Im Falle des Beispielnetzes aus Abb. 37 identifiziert der Quantenalgorithmus korrekt Ausfallinitialwahrscheinlichkeit P_1 von Knoten 1 (power), siehe Abb. 38.

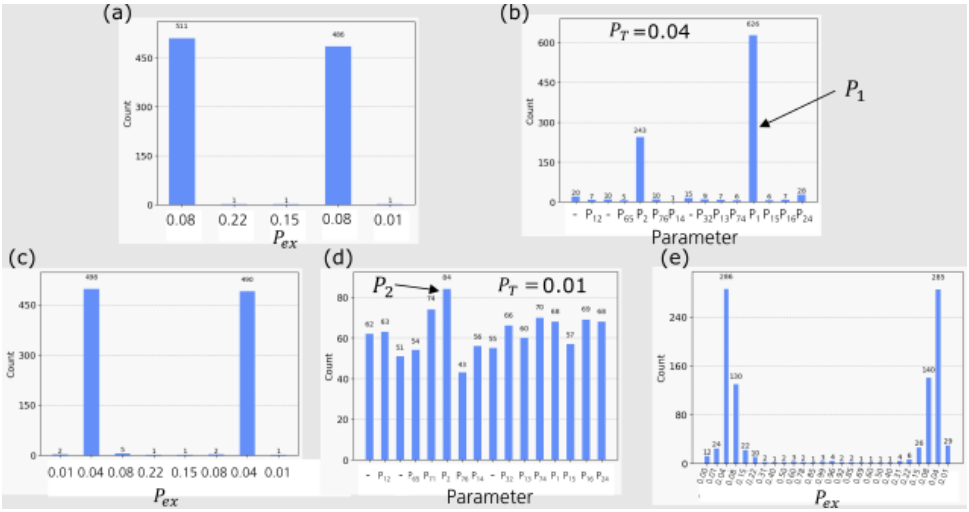


Abb. 38 (a) Wahrscheinlichkeit eines Performanzabfalls P_{ex} unter einen gesetzten Schwellwert für das untersuchte Netzwerk ermittelt durch QAE. (b) Grover-Suche nach dem Parameter, dessen Variation P_{ex} auf den nächsten kleineren Wert $P_T = 0.04$ reduziert (hier 0.04 aufgrund der genutzten Auflösung). Identifiziert wird P_1, die Ausfallinitialwahrscheinlichkeit von Knoten 1 (power). (c) P_{ex} für das untersuchte Netzwerk mit Variation von Parameter P_1. (d) Grover-Suche nach dem Parameter, dessen Variation P_{ex} auf den nächsten kleineren Wert $P_T = 0.01$ reduziert. Das Ergebnis ist nicht eindeutig. Da die Ausfallinitialwahrscheinlichkeit P_2 von Knoten 2 (hospital) den höchsten Wert erreicht, wird unter (e) P_{ex} für das untersuchte Netzwerk mit Variation von Parameter P_2 bestimmt (mithilfe von QAE). Der Mittelwert der Wahrscheinlichkeiten $P_{ex}(variation\ von\ P_2)$ liegt dabei höher als $P_{ex}(variation\ von\ P_1)$. Damit ist P_1 als Parameter mit dem größten Einfluss auf die Gesamtperformanz des Netztes identifiziert.

Verwertung

Zwei Demonstratoren wurden auf der Projektwebsite veröffentlicht.

References

[1] H. P. Breuer and F. Petruccione. The Theory of Open Quantum Systems. Oxford University Press, Oxford, 2002.
[2] M.C. Braun, T. Decker, N. Hegemann, S.F. Kerstan and C. Schäfer. A Quantum Algorithm for the Sensitivity Analysis of Business Risks. arXiv, 2021.

2.1.5 Entwurfsentscheidungen im Quantencomputing

Im Rahmen des ersten Arbeitspakets wurden existierende Anwendungsfälle aus SEQUOIA betrachtet und zusammengefasst. Die resultierenden Arbeitsartefakte sind als Vorarbeit für Arbeitspaket 5 zu verstehen, in dem ein tieferes Verständnis von Hyperparametern und weiteren Freiheitsgraden im Quantencomputing nötig ist. Hierbei wurden insbesondere Freiheitsgrade innerhalb der mathematischen Problemdefinition und Abbildung auf QC-Modelle betrachtet. Die Erkenntnisse der Arbeit wurden veröffentlicht, siehe [1]. Die Erkenntnisse der Arbeit werden maßgeblich für AP5 verwendet, um eine gezielte Entwurfsraumexploration zu gewährleisten.

Referenzen

[1] M. Scheerer, J. Klamroth, S. Garhofer, F. Knäble and O. Denninger, "Experiences in Quantum Software Engineering," *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, St. Petersburg, FL, USA, 2023, pp. 552-559, doi: 10.1109/IPDPSW59300.2023.00095.

2.1.6 Konfigurationspriorisierung für variable Softwaresysteme

Eine Software-Produktlinie modelliert die Variabilität eines hochkonfigurierbaren Systems, mit Hilfe von auswählbaren Features. Eine Menge von ausgewählten Features stellt eine einzigartige Produkt-Konfiguration dieses Systems dar. Zusätzlich können einzelne Features noch mit Attributen (z.B. den Kosten) assoziiert werden. Solche Software-Produktlinien lassen sich auf Basis von attributierten Feature-Modellen [1] modellieren und analysieren. Die Einschränkungen, welche durch das Feature-Modell enkodiert werden, lassen sich in eine aussagelogische Formel in konjunktiver Normalform (CNF) überführen. Z. B. für die aussagelogische Formel $(x_1 \vee x_2) \wedge (x_3)$ mit den Features x_1, x_2, x_3 wären {1,0,1}, {0,1,1} und {1,1,1} gültige Belegungen bzw. valide Konfigurationen, wobei 0 für Feature nicht gewählt und 1 für Feature gewählt steht.

Eichhorn et al. [2] analysieren algorithmische Verfahren, die in der klassischen Produktlinienanalyse für Konfigurationsprobleme verwendet werden, um Potenziale und Herausforderungen für das Quantencomputing zu identifizieren. Bei dem konkreten Anwendungsfall „Konfigurationspriorisierung" soll eine bestimmte Anzahl an Konfigurationen ausgegeben werden, welche folgende Eigenschaften erfüllen sollen: (1) die Konfigurationen sollen valide für die Einschränkungen des Feature-Modells sein und (2) die Konfigurationen sollen bezüglich einer Kostenfunktion priorisiert sein. Bei der Konfigurationspriorisierung handelt es sich um ein Bedingungserfüllungs- und Optimierungsproblem (CSOP). Die Kostenfunktion bestimmt dabei die Kosten einer Konfiguration anhand der Attribute der ausgewählten Features. Die Priorisierung erfolgt dabei nach aufsteigenden Kosten, somit hätte die erste Konfiguration minimale Kosten.

Folgende Ergebnisse sind Teil der SEQUOIA-Veröffentlichung von Ammermann et al. [3]:

In AP 1.1 wurde nun die klassische Modellierung des Konfigurationsproblems als attributiertes Feature-Modell wie folgt auf eine quantenmechanische Formulierung für Optimierungsprobleme als Ising-Modell überführt. Da die CNF-Darstellung eines Feature-Modells (1) nicht-quadratische Terme enthalten kann, überführen wir diese in die Form eines polynomiellen, binären Optimierungsproblem ohne Nebenbedingungen (PUBO). Dieses kann direkt in die Ising-Form [4] oder durch Quadratisierung [5] zuerst in eine quadratische Form (QUBO) und dann in die Ising-Form übersetzt werden, was in einem Hamiltonian H_C resultiert. Die QUBO-Formulierung benötigt dabei zusätzliche Qubits aufgrund von benötigten Hilfsvariablen, wobei die PUBO-Formulierung in tieferen Schaltkreisen resultiert. Die attributierten Kosten (2) lassen sich direkt als QUBO mit $\min k\,(x) = \sum_x c_i\, x_i$ formulieren und als Hamiltonian H_{c_k} in Ising-Form darstellen. Um das Gesamtproblem abzubilden, erstellen wir nun den Hamiltonian $H_{C_{tot}} = H_{c_k} + \alpha H_C$. Der Regularisierungsparameter α gewichtet die Teilproblem-Hamiltonians, damit sowohl die Bedingungserfüllung als auch die Optimierung in die Problemlösung ausreichend einfließen.

Zur Lösung des Problems auf einem Quantenrechner wurde in AP 1.2 eine Python Bibliothek implementiert, welche die oben beschriebenen Transformationen für beliebige Instanzen automatisiert durchführen kann. Aus diesen Problemformulierungen können

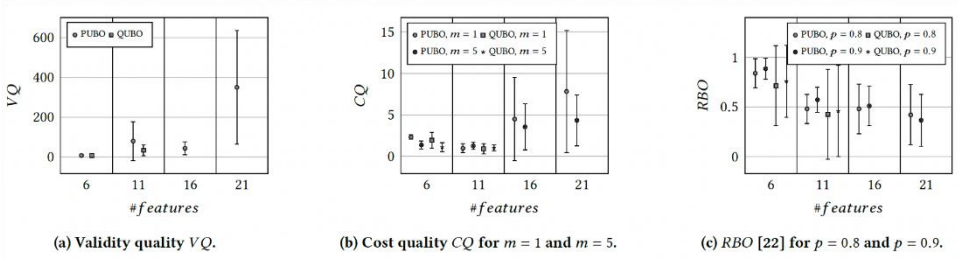
mit der Bibliothek parametrisierte Schaltkreise erstellt und für kleine Probleminstanzen simuliert werden. Die Arbeit wurde an 20 Probleminstanzen evaluiert, welche mit dem Werkzeug FeatureIDE generiert wurden. Zur Evaluation wurden drei Metriken herangezogen. Die Metrik Validitätsqualität

$$VQ = 2^{\#features} \cdot \frac{\sum_{c \in C_v} P(c)}{|C_v|}$$

misst, um wie viel höher die durchschnittliche Wahrscheinlichkeit einer gültigen Konfiguration ist, wenn QAOA verwendet wird, anstatt zufällig zu raten. C_v bezeichnet die Menge der gültigen Konfigurationen. Die Wahrscheinlichkeit P über diese Konfigurationen wird kumuliert und durch das Verhältnis der Anzahl aller möglichen Konfigurationen zur Anzahl der gültigen Konfigurationen geteilt.
Die Kostenqualität

$$CQ = \frac{\sum_{c \in C_v} P(c)}{|C_v|} \cdot \frac{\sum_{c \in C_m} P(c)}{m}$$

misst, um wie viel höher die durchschnittliche Wahrscheinlichkeit einer der besten m-Konfigurationen ist, wenn QAOA verwendet wird, anstatt zufällig nur aus gültigen Konfigurationen zu raten. $C_m \subseteq C_v$bezeichnet die Menge der m besten gültigen Konfigurationen. Rank-biased overlap (RBO) [6] misst die Ähnlichkeit zweier Listen, wobei das Gewicht am Kopf der Listen höher ist. Abb. 39 zeigt die durchschnittliche Ergebnisqualität von VQ, CQ und RBO für die Benchmark-Instanzen. VQ und CQ steigen mit der Größe der Probleminstanz. RBO nimmt mit der Größe der Probleminstanz ab. Damit werden gute Ergebnisse bei der Auswahl von Konfigurationen erzielt, aber für die Priorisierung von Konfigurationen muss der Ansatz weiter verbessert werden.



(a) Validity quality VQ. (b) Cost quality CQ for $m = 1$ and $m = 5$. (c) RBO [22] for $p = 0.8$ and $p = 0.9$.

Abb. 39 Durchschnittliche Ergebnisqualität von VQ, CQ und RBO für Benchmark-Instanzen. Die Fehlerbalken geben die Standardabweichung an. Für alle Metriken zeigt ein höherer Wert eine bessere Ergebnisqualität an.

Referenzen

[1] N. Siegmund, S. Sobernig, and S. Apel, "Attributed variability models: outside the comfort zone," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, Paderborn Germany: ACM, Aug. 2017, pp. 268–278. doi: 10.1145/3106237.3106251.

[2] D. Eichhorn, T. Pett, T. Osborne, and I. Schaefer, "Quantum Computing for Feature Model Analysis: Potentials and Challenges", in *27th ACM International Systems and Software Product Lines Conference (SPLC'23)*, 2023. doi: https://doi.org/10.1145/3579027.3608971

[3] J. Ammermann et al., "Quantum Approach to the Configuration Selection and Prioritization Problems", 2024 ACM/IEEE International Workshop on Quantum Software

Engineering (Q-SE 2024), April 16, 2024, Lisbon, Portugal. ACM, New York, NY, USA, *to appear*.
[4] A. Glos, A. Krawiec, and Z. Zimborás, "Space-efficient binary optimization for variational computing." arXiv, Sep. 15, 2020. Available: http://arxiv.org/abs/2009.07309
[5] N. Dattani, "Quadratization in discrete optimization and quantum mechanics." arXiv, Sep. 23, 2019. Available: http://arxiv.org/abs/1901.04405
[6] Webber, W., Moffat, A., and Zobel, J. A similarity measure for indefinite rankings. ACM Transactions on Information Systems (TOIS) 28, 4 (2010), 1–38.

2.2 Arbeitspaket 2 – Algorithmendesign

Definiertes Ziel des zweiten Arbeitspakets ist laut Projektbeschreibung die Weiterentwicklung bestehender Quantenalgorithmen, insbesondere hybride Quantenalgorithmen. Zusätzlich werden neue, quantenbasierte KI-Algorithmen entwickelt und mit klassischen Methoden verglichen. Die erarbeiteten Algorithmen fließen kontinuierlich in die Anwendungsfälle aus Arbeitspaket 1 – Anwendungsfälle ein, wo jene Arbeiten zum großen Teil schon beschrieben worden sind. Im Folgenden werden jene Ergebnisse, welche nicht bereits im vorherigen Kapitel aufgegriffen worden sind, detailliert vorgestellt.

Unter Leitung des **Fraunhofer IPA** wird die Algorithmikforschung in Arbeitspaket 2 anhand dreier Arbeitsschritte durchgeführt (vgl. Gantt-Chart Abb. 2).

AP 2.1 Quantenalgorithmen für Optimierungsprobleme

- Entwicklung eines hybriden quantenklassischen Algorithmus zur Verifikation spezifizierter Eigenschaften neuronaler Netze zur Verkehrszeichenerkennung
- Entwicklung und Bewertung eines HPC-beschleunigten hybriden quantenklassischen ADMM-Algorithmus zur Planung von LKW-Touren
- Implementierung des in [Braine et al. 2019] vorgeschlagenen Algorithmus und Vergleich mit dem ADMM-Algorithmus
- Systematischer Vergleich verschiedener Implementierungen des QAOA-Algorithmus für kombinatorische Optimierung (Ergebnisse siehe 2.1.1) hinsichtlich verschiedener (Hyper-) Parameter
- Erarbeitung problemspezifische automatische QAOA-Schaltkreisgenerierung
- Untersuchung von fehlerresistenten QAOA-Problemcodierungen, sodass mithilfe von Redundanzen entstandene Fehler gefunden oder korrigiert werden können

AP 2.2 Quantenalgorithmen für maschinelles Lernen

- Systematischer Vergleich verschiedener Datenenkodierungsstrategien und Quanten-Feature-Map-Architekturen für Klassifizierungs- und Regressionsprobleme
- Untersuchung von KI-Algorithmen zur automatischen Konstruktion von Quanten-Feature-Maps als Grundlage eines anwendungsorientierten QML-Frameworks

AP 2.3 Quantenalgorithmen für Differentialgleichungen und lineare Systeme

- Entwicklung und Bewertung eines hybriden, quantenklassischen neuronalen Netzwerkalgorithmus zur Fluiddynamiksimulation (Ergebnisse siehe 2.1.2)
- Vergleich mit klassischen KI-Algorithmen unter dem Aspekt der Vorhersagegenauigkeit
- Weiterentwicklung des bisherigen Ansatzes (VQLS) bestehend aus einer klassischen Diskretisierung der Differentialgleichungen in Kombination mit Lösung der entstehenden linearen Gleichungssysteme mit einem variationellen Algorithmus

AP 2.4 Quantenalgorithmen für die Resilienzanalyse

- Übersetzung stochastischer Prozesse in Quantentrajektorien zur topologischen Robustheits- und Resilienzanalyse von gekoppelten Infrastrukturnetzen
- Implementierung von inkohärenten Quantenübergängen in Drei-Niveau-Systemen durch Projektion kohärenter Dynamiken in erweiterten Hilberträumen
- Bestimmung von Wartezeitverteilungen für Kaskadenprozesse
- Entwicklung eines Quantenalgorithmus zur Resilienzsteigerung von Infrastrukturnetzen durch Identifikation kritischer Netzwerkparameter

Jenem Arbeitsplan liegen dabei zwei Meilensteine zugrunde:

- **M8:** Identifizierung hybrider Methoden in allen genannten Bereichen. Abschluss der Algorithmenkonzeption und Beginn der Implementierungsphase.
- **M15:**Abschluss der Implementierungen der entwickelten hybriden quantenklassischen Algorithmen. Vergleiche mit klassischen Algorithmen wurden erstellt und die Methoden wurden in Anwendungsfällen angewandt und bewertet.

Beide Meilensteine (M8 und M15) sind planmäßig erreicht worden (vgl. Abb. 2).
Im Folgenden wird der Endstand jener Algorithmikforschung dargestellt, welcher nicht bereits im Zuge der Anwendungsfälle im Vorkapitel 2.1 ausführlich erörtert wurde. Die folgenden Ergebnisse wurden ebenfalls zu wissenschaftlichen Publikationen ausgearbeitet und zum Teil bereits auf der »Quantum Effects« im Oktober 2023 vorgestellt (siehe Publikationen).

2.2.1 Verifikation neuronaler Netze in der Verkehrszeichenerkennung

Problembeschreibung und Algorithmus:

Im Rahmen des Arbeitspakets zur Weiterentwicklung bestehender Quantenalgorithmen haben wir einen hybriden quantenklassischen Algorithmus zur Verifikation von neuronalen Netzen entwickelt. Ziel war es, die Robustheit der Netzwerke in Bezug auf Verkehrszeichenerkennung zu überprüfen. Das Problem ist relevant für die Sicherheit von Fahrerassistenzsystemen und autonomen Fahrzeugen. Es soll sichergestellt werden, dass neuronale Netze, die für die Verarbeitung von Verkehrszeichen verwendet werden, nicht durch Bildartefakte wie Rauschen oder Umgebungsbedingungen wie Reflexionen oder Nebel fehlgeleitet werden können.

Um die Eigenschaften des neuronalen Netzwerks zu überprüfen, haben wir einen Ansatz gewählt, der darauf abzielt, eine Eigenschaft P zu falsifizieren. Hierfür reicht es aus, einen Beispieldatenpunkt (in diesem Fall ein Bild) x zu suchen oder zu konstruieren, der diese Eigenschaft verletzt. Hierzu wurde das Verfahren Quanten-deterministisches Annealing (Quanten-DA) entwickelt und erprobt, was eine quantenmechanische Erweiterung des klassischen Algorithmus "deterministisches Annealing"(DA) ist [1]. Diese Verifikationsmethode ist korrekt, aber nicht vollständig: Ein Verifikationsalgorithmus ist "korrekt", wenn alle gemachten Aussagen tatsächlich wahr sind. Er ist "vollständig", wenn er alle wahren Aussagen auch als solche identifiziert.
Es wurde ein quantenklassischer Hybridalgorithmus entwickelt, der das oben vorgestellte Problem löst. Der Algorithmus ist in Pseudocode in Abb. 24 gezeigt. Ziel des Algorithmus ist es Datenpunkte x^* zu konstruieren, die Ähnlichkeiten mit den tatsächlichen Datenpunkten x aufweisen, deren Klasse aber falsch vorhergesagt wird. Beispiele hierfür finden sich weiter unten im Bericht.

Algorithm Minimizes $F(\mathbb{P})$ over the convex hull of $X = \{x_1, \ldots, x_M\}$

Inputs: $0 < \alpha < 1$ (cooling factor), $T_0, T_{\min} > 0$ (initial and final annealing temperature), integer $K_{\max}$ (maximum size of the credal set)
initialize the annealing temperature $T \leftarrow T_0$
initialize the credal set $\mathcal{Q} = \{U\}$ where $U(x_i) = 1/|X|$ is the uniform probability distribution over X
while $T \geq T_{\min}$ **do**
    for each probability distribution $\mathbb{P}$ in the credal set $\mathcal{Q}$ **do**
        minimize the free energy $F(\mathbb{P}) = E(\mathbb{P}) - T \cdot S(\mathbb{P})$ wrt. $z \in \mathbb{R}^M$, where $E(\mathbb{P}) = h(\mathbb{E}_{\mathbb{P}}[X]) = h\left(\sum_{i=1}^{M} P_i x_i\right)$, $S(\mathbb{P}) = -\sum_{i=1}^{M} P_i \ln P_i$ and the probability distribution $\mathbb{P} = (P_1, \ldots, P_M)$ is encoded as a softmax function of the variable z: $P_i = \frac{e^{z_i}}{\sum_{j=1}^{M} e^{z_j}} \quad \forall i \in \{1, \ldots, M\}$
        compute the $M \times M$ Hessian matrix $H_F(\mathbb{P})$ of the free energy function F
        compute the minimum eigenvalue $\lambda_{\min}$ and eigenvector v of H_F
        if $\lambda_{\min} < 0$ (a phase transition occurs) **then**
            replace $\mathbb{P}$ by $\mathbb{P} + \epsilon v$ and $\mathbb{P} - \epsilon v$ in the credal set $\mathcal{Q}$, unless the maximum size $K_{\max}$ has been reached
        end if
    end for
    make sure the size of $\mathcal{Q}$ does not exceed $K_{\max}$ by removing the distribution with highest free energy $F(\mathbb{P})$ if necessary
    cooling step: $T \leftarrow \alpha \cdot T$
end while
return $x = \mathbb{E}_{\mathbb{P}}[X]$ where $\mathbb{P} = \arg\min_{\mathbb{P} \in \mathcal{Q}} E(\mathbb{P})$

Abb. 40 Algorithmus für Quanten-Deterministisches Annealing für die Verifikation Neuronaler Netzwerke.

Zentral im obigen Algorithmus ist das Auffinden der Eigenwerte der Hessematrix der freien Energie. Diese ist eine quadratische Matrix, die aus den zweiten Ableitungen einer Funktion besteht. Das Auffinden dieser Werte kann mit herkömmlichen Methoden schwierig sein, insbesondere bei großen Matrizen. In dem von uns entwickelten Algorithmus werden die Eigenwerte der Hessematrix daher mit einem Quantencomputer bestimmt. Zum Einsatz kommt hier der Algorithmus VQE (Variational Quantum Eigensolver). Dies ist ein hybrider Algorithmus, der das Variationsprinzip nutzt, um die Erwartungswerte zu minimieren.

Resultate

Der Algorithmus wurde auf den *German Traffic Sign Recognition Benchmark* (GTSRB) Datensatz angewendet. Er enthält 51.839 Bilder von Verkehrsschildern (39.209 Trainingsbilder und 12.630 Testbilder) in 43 Klassen [2]. Beispiele sind in Abb. 41 zu sehen.

Abb. 41 Exemplarische Bilder aus dem GTSRB Datensatz.

Zur Verarbeitung wurden die Bilder auf eine Größe von 30x30 Pixeln reduziert. Die verwendete Neuronale Netz-Architektur, ist in Abb. 42 zu sehen. Dieses Neuronale Netz wurde mittels des oben spezifizierten Quanten-DA Algorithmus verifiziert.

- 2 Conv2D layers (filter=32, kernel_size=(5,5), activation="relu")
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.25)
- 2 Conv2D layers (filter=64, kernel_size=(3,3), activation="relu")
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.25)
- Flatten layer to squeeze the layers into 1 dimension
- Dense Fully connected layer (256 nodes, activation="relu")
- Dropout layer (rate=0.5)
- Dense layer (43 nodes, activation="softmax")

Abb. 42 Architektur des Convolutional Neural Networks (CNN), was durch den Quanten-DA-Algorithmus verifiziert wurde.

Das Netz wurde mit einem Adam Optimierer trainiert. Nach 15 Epochen wurde eine Testgenauigkeit von 96.52% in der Klassifikation erreicht (sieheAbb. 43).

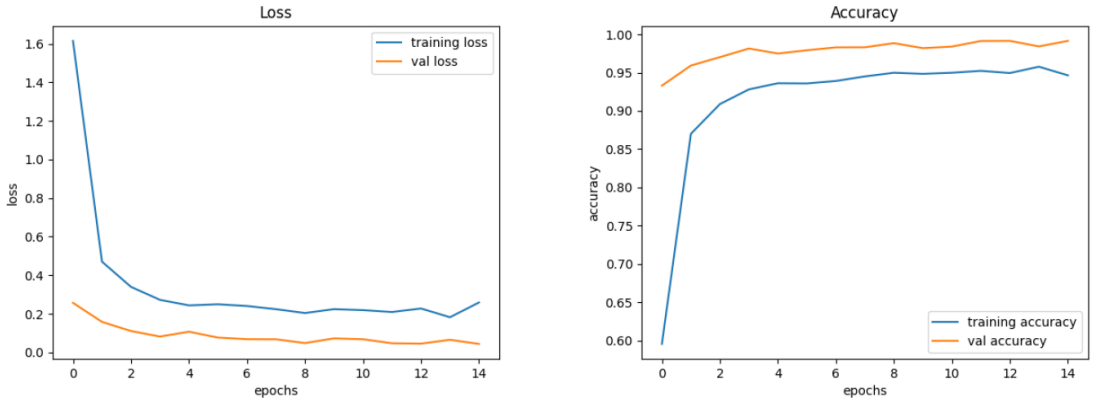


Abb. 43 (links) Trainings und Validationloss während des Trainings des in **Abb. 42** beschriebenen CNNs. (rechts) Genauigkeit als Funktion der Trainingsepochen.

Als zu verifizierende Eigenschaft wurde die Robustheit des Netzes untersucht: Ausgangspunkt ist ein Bild x, was vom CNN
(korrekt) als Klasse c klassifiziert wird. Anschließend wird das Bild unter verschiedenen Szenarien, die an den tatsächlichen Fahrbetrieb angelehnt sind, verfremdet und verifiziert, dass das Bild noch der ursprünglichen Klasse entspricht. Hierfür wurde die *imgaug*-Bibliothek verwendet [3]. Es wurden 11 verschiedene Transformationen auf den GTSRB Datensatz angewendet, darunter Helligkeit, Dunkelheit, Unschärfe und Rauschen. Beispiele sind in Abb. 44 zu sehen.


Figure: Original


Figure: Darkness


Figure: Noise


Figure: Brightness


Figure: Blur


Figure: Fog

Abb. 44 Beispiele für verschiedene Verfremdungen. Ziel ist es sicherzustellen, dass die Klasse unter Einfluss der Transformationen erhalten bleibt.

Der Quanten-DA-Algorithmus konnte verschiedene Adversial Attacks generieren, also Beispiele, die im zu verifizierenden Netz zu einer Änderung der vorhergesagten Klasse führen. Viele der Beispiele sind von menschlichen Beobachtern nicht vorherzusagen und stellen damit ein besonderes Risiko dar. Beispiele für Adverserial Attacks sind in Abb. 45 zu sehen.


Figure: #2157
(class 27)


Figure: QA#2157
(class 18)


Figure: Class 18

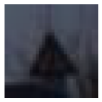


Figure: #9242
(class 18)

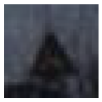


Figure: QA#9242
(class 40)


Figure: Class 40


Figure: #11990
(class 11)


Figure: QA#11990
(class 27)


Figure: Class 27

Abb. 45 Beispiele für verschiedene von unserem Algorithmus gefundene Adverserial Attacks. Die generierten Bilder weichen vom Ursprungsbild durch eine Transformation ab und sorgen für eine Änderung der vorhergesagten Klasse, ohne, dass der Fehler notwendigerweise von einem menschlichen Beobachter nachvollzogen werden kann.

Zusammenfassung

Im Projekt wurde eine quanten-klassische Hybridmethode entwickelt, die auf Basis von deterministischem Annealing eine zu spezifizierende Eigenschaft eines neuronalen Netzes verifiziert. Die Quantenkomponente besteht aus dem Auffinden von Eigenwerten einer im Algorithmus auftretenden Hessematrix. Die Verifikation wurde über Falsifikation

erreicht. Die Methode wurde auf Basis eines großen Datensatzes von Bildern deutscher Verkehrszeichen getestet und die Wirksamkeit wurde demonstriert. Die hier entwickelte Methode wurde modular in einer Python Bibliothek implementiert und steht für die weitere Verwendung zur Verfügung.

Verwertung

Die Ergebnisse wurden mit einem Poster auf der "KQCBW Developer Conference" am 7./8. März 2024 vorgestellt. Weiterhin wurde ein Demonstrator erstellt, der die Methode illustriert. Dieser ist auf der SEQUOIA End-to-End Homepage verfügbar.

Referenzen

[1] K. Rose, Deterministic Annealing for clustering, compression, classication, regression and related optimization problems, Proceedings IEE, vol. 86, num. 11, p. 2210-2239, 1998
[2]https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign
[3] https://imgaug.readthedocs.io/en/latest

2.2.2 Quantenalgorithmen für Orienteering Problems

Im Rahmen von SEQUOIA 1 wurde ein Anwendungsfall betrachtet, bei dem ein klassisches Routenplanungsproblem gelöst werden musste. Hierbei wurde das grundlegende mathematische Problem als *Traveling-Salesperson (TSP)*-Problem definiert. Darüber hinaus wurde der resultierende Graph des TSPs partitioniert (auf einem klassischen Rechner) und nur der schwierige kombinatorische Kern des Problems auf dem Quantencomputer verlagert. Als alternativen Ansatz wurde schließlich das grundlegende mathematische Problem als *Orienteering Problem* (eine Kombination des *Knappsack*- und *Traveling-Salesperson*-Problems) formuliert und auf einem Quantencomputer ausgeführt. Das Ziel lag dabei darin, nicht nur eine alternative Implementierung zu prüfen, sondern die Auswirkungen auf die Komplexität des zugrunde liegenden Optimierungsproblems zu bewerten und mit dem TSP-Ansatz zu vergleichen. Hierfür wurde ein entsprechendes Jupyter Notebook realisiert. Obwohl der Ansatz eine interessante Alternativlösung bietet, konnten jedoch keine größeren Vorteile gegenüber dem klassischen TSP-Ansatz ermittelt werden.

2.2.3 Quantenalgorithmen für maschinelles Lernen

Motivation

Um Daten mit Quantencomputern zu verarbeiten, müssen sie erst in einen Quantenzustand enkodiert werden. Anschließend werden verschiedene Quantenoperationen angewendet. Die Kombination aus Enkodierung und Verarbeitung wird häufig als Quanten-Feature-Map bezeichnet. In der Auswahl und Verschaltung der Quantenoperationen besteht eine große Freiheit. Die Wahl und Anordnung der Quantenoperationen bestimmen die Performance der jeweiligen QML-Methode. Es hat sich gezeigt, dass es entscheidend ist, wie gut die Architektur auf das vorliegende Problem angepasst ist (inductive bias) [1]. In der Forschung sind zahlreiche ad-hoc

Ansätze für die Gestaltung von Quanten-Feature-Maps bekannt. Diese sind allerdings in der Regel nicht auf das zu lösende Problem zugeschnitten. In der Praxis führt das zu Schwierigkeiten beim Training und der Übertragung der Modelle auf Daten, die nicht während des Trainings verwendet wurden. Hierdurch wird die Anwendbarkeit der Algorithmen für industrierelevante Problemstellungen deutlich limitiert. In diesem Arbeitspaket haben wir eine Methode entwickelt, die auf Basis vorliegender Daten automatisiert neue Feature-Map Architekturen bestehend aus passenden Quantenoperationen generiert. Dies ermöglicht das Erstellen von QML-Modellen, die den Anforderungen der derzeit verfügbaren Hardware genügen. Zugleich sind die so entstehenden Modelle deutlich performanter als vergleichbare Modelle aus der Literatur, wodurch ein deutlicher Schritt hinsichtlich der Praktikabilität dieser Algorithmen in der industriellen Anwendung gemacht wird.

Methode

Für die automatische Konstruktion von Quanten-Feature-Maps wurden verschiedene Ansätze aus dem Bereich des Reinforcement Learning (RL) [2, 3] untersucht und angewendet. Hierbei wurden Verfahren vom Typ des modellfreien- und modellbasierten RL verwendet. Im Bereich des modellfreien RL wurde insbesondere die Proximal Policy Optimization (PPO) [2] und eine sogenannte Actor-to-Critic (A2C) Methode getestet. In diesem Vergleich schnitt die PPO in jedem Test besser ab als die A2C Methode.
Als modellbasierter Algorithmus MuZero [3] wurde vorrangig MuZero verwendet. Der Algorithmus gilt als einer der besten state-of-the-art RL-Suchalgorithmen. In umfangreichen Vergleichen hat sich der modellbasierte MuZero Ansatz als die beste Wahl herausgestellt. Nicht nur lieferte er die besten Ergebnisse, aufgrund des modellbasierten Monte-Carlo-Tree-Search Verfahrens ist er auch am effizientesten, da das QML-Modell deutlich seltener evaluiert werden muss als bei den modellfreien Varianten.

RL-Verfahren sind sehr sensibel gegenüber der Definition der sogenannten Belohnungsfunktion. Ist diese in geeigneter Art und Weise konstruiert, kann im Idealfall über eine Feedbackschleife eine Quanten-Feature-Map konstruiert werden, die zu einem performanten QML-Verfahren führt. Es wurden zwei Ansätze für die Definition der Belohnungsfunktion und der Umgebung des RL-Verfahrens erarbeitet und getestet. Im ersten Ansatz wurde eine spezifische QML-Methode (beispielsweise Quanten Support Vector Machine (QSVM), oder Quanten Kernel Ridge Regression (QKRR)) als Interaktionsumgebung für den Algorithmus verwendet. Als Metrik für die Belohnungsfunktion wurde der Mean-Squared-Error (MSE) des Verfahrens auf dem Trainingsdatensatz verwendet. Im zweiten Ansatz basiert die Belohnung rein auf Eigenschaften der Quanten Kernel Matrix aus den Trainingsdaten. Hier wurden verschiedene Metriken zur Analyse der Matrixeigenschaften herangezogen. Untersucht wurden das sogenannte Kernel-Target-Alignment (KTA), die Geometrische Differenz und der Prediction Error Bound (PEB) [4]. Als besonders vielversprechend zur Bewertung der Generalisierungseigenschaften der Methode hat sich hierbei die Metrik des PEB herausgestellt. Beide Ansätze sind mit Quanten Kerneln des Typs Fidelity Kernel [5] und des Typs Projected Kernel [4] getestet worden. Hiermit wurde Meilenstein M8 erreicht.

Aus den umfangreichen Tests der oben beschriebenen RL-Algorithmen, QML-Algorithmen und Metriken hat sich die folgende Konfiguration als die leistungsfähigste herausgestellt: Als RL Algorithmus wurde MuZero gewählt. Die Belohnungsfunktion basiert auf der Performance eines QML-Modells (z.B. R2 für Regressionsdaten oder Genauigkeit für Klassifikationsdaten). Zum Vermeiden von Overfitting wird die Performancemetrik durch Crossvalidation evaluiert. Die QML-Methode, die hierzu trainiert wird, ist eine QSVM mit einem Projected Quantum Kernel. Eine schematische Darstellung des finalen Algorithmus ist in Abb. 46 zu sehen.

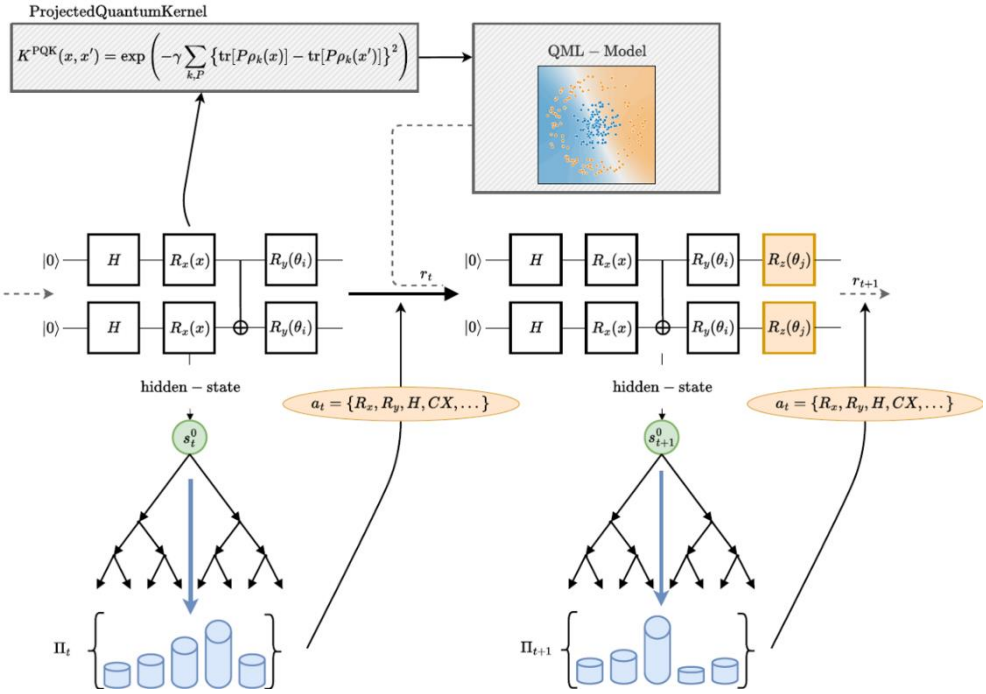


Abb. 46 Finaler Algorithmus: In jeder Iteration evaluiert der Algorithmus, welche zusätzliche Quantenoperation aus einem fixen Gateset optimalerweise durchgeführt wird. Hierzu wird ein klassisches neuronales Netz trainiert. Der daraus resultierende Quantenschaltkreis wird zur Erstellung eines Projected Quantum Kernels genutzt. Dieser von einer QSVR genutzt, um eine Performancemetrik auf einem Testdatensatz zu generieren. Im folgenden Schritt wird die nächste Quantenoperation auf gleiche Weise bestimmt. Dieses Vorgehen wird bis zu einer bestimmten Länge wiederholt. Daraufhin wird ein finaler Score berechnet und von vorne begonnen, bis eine fixe Anzahl an T Iterationen durchlaufen ist.

Benchmarking und Resultate

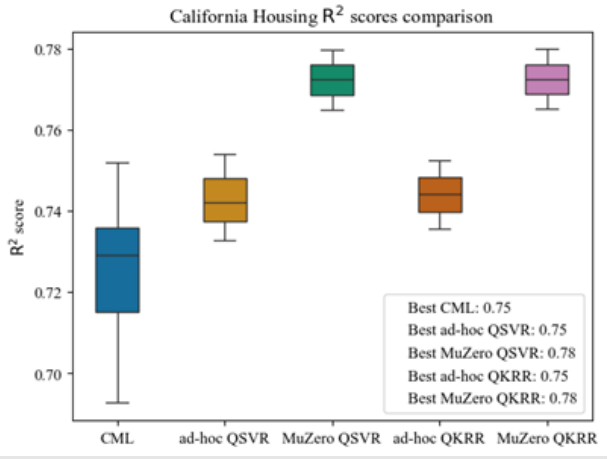
Zur Validierung der Methode wurde ein umfangreiches Benchmarking durchgeführt. Hierbei ist die Auswahl der Datensätze essenziell. Aufgrund der aktuell noch begrenzten Kapazitäten von Quantencomputern und Simulatoren sind Beschränkung Hinsichtlich der Anzahl an Datenpunkten und Features nötig. Gleichzeitig müssen die Datensätze so gewählt werden, dass das resultierende Machine Learning Problem nicht trivial ist. Für unsere Tests wurden die folgenden 4 Datensätze ausgewählt. Dabei handelt es sich um drei Reggresionsprobleme und ein Klassifikationsproblem

1) Quanten Fashion MNIST-Datensatz (QFMNIST) [4] zum Testen des Verfahrens verwendet. Dabei handelt es sich um einen künstlich generierten Benchmark-Datensatz. Dieser wurde selbst implementiert, so dass eine große Freiheit hinsichtlich der Konfiguration möglich war. Der Datensatz besteht aus jeweils 100 Training und Testpunkten und 4 Featuren.

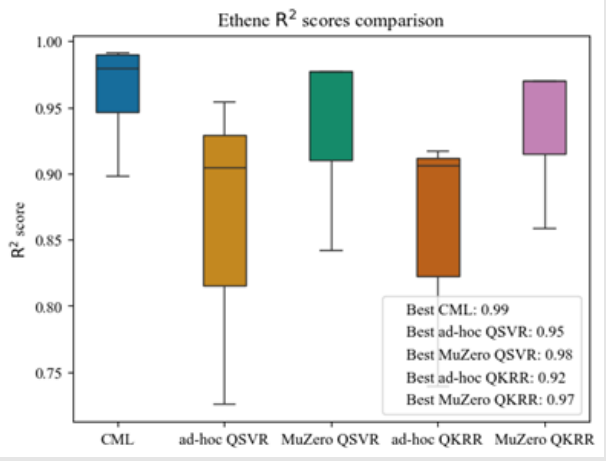
2) California Housing Datensatz. Regressionsproblem bestehend aus der Preisvorhersage von Immobilien. Der Originaldatensatz besteht aus 20640 Datenpunkten. Hiervon wurde eine zufällige Unterstichprobe von 1000 Datenpunkten erhoben. Von den 9 Featuren wurden alle verwendet
3) Ethen Datensatz: Vorhersage der Grundzustandsenergie eines Moleküls aufgrund der Molekülkoordinaten. Der Datensatz enthält 12 Features und besteht aus 100 Trainingspunkten und 200 Testpunkten.
4) PLASTICC Datensatz [6]: Binärer Klassifikationsdatensatz aus der Astronomie. Die 67 Features wurden durch eine Vorverarbeitung (PCA) auf 9 reduziert. Hiervon wurde eine zufällige Unterstichprobe von 1000 Datenpunkten erhoben.

Die Benchmarking-Routine beinhaltet den Vergleich verschiedener QML-Methoden – basierend auf den erstellten Feature-Maps und basierend auf mehreren aus der aktuellen Forschung bekannten ad-hoc Architekturen. Ebenso wurde die Performance mit derer klassischer Methoden verglichen. Hierzu wurden mehrere klassische Verfahren betrachtet, beispielweise die SVMs und die KRR (diese sind die klassischen Äquivalente zu der QSVM und der QKRR). Weitere klassische Modelle wie Neuronale Netze, Random Forests und Gradient Boosting Modelle, wurden ebenfalls betrachtet, um einen fairen Vergleich der bestmöglichen Leistung zu ermöglichen.
Für alle klassischen Methoden wurden die Hyperparameter mittels State-of-the-Art Algorithmen optimiert. Bei den ad-hoc Quanten-Feature-Maps wurden (falls vorhanden) die variationellen Parameter mittels Kernel-Target-Alignment optimiert.

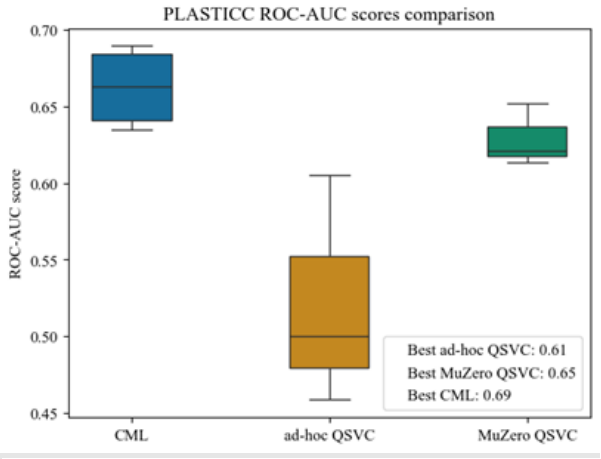
a)



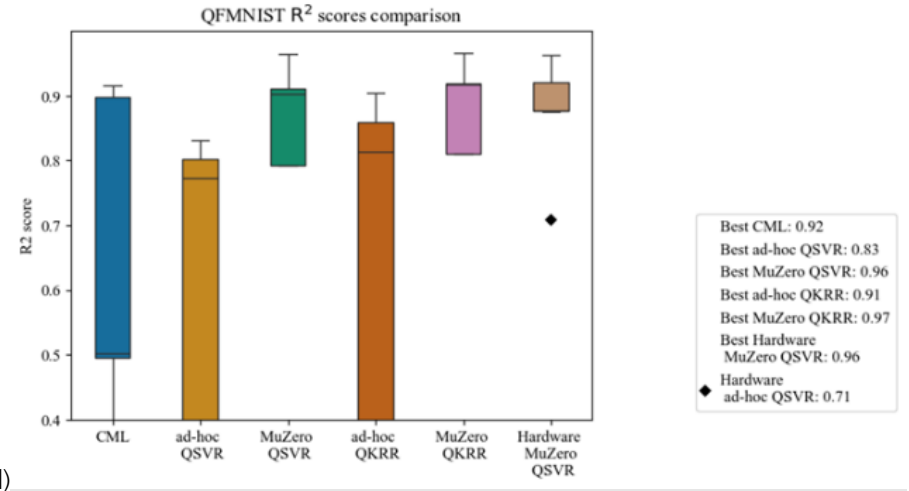
b)



c)



d)



e)

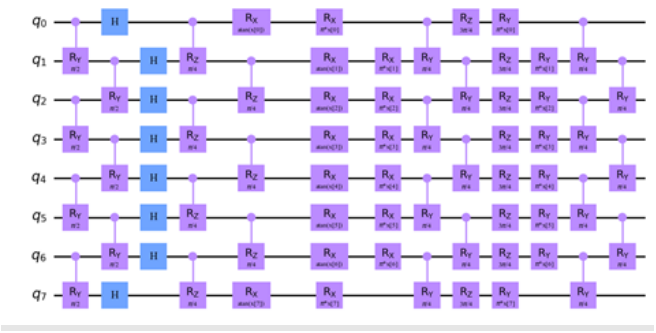


Abb. 47 Box-plot der Performance verschiedener klassischer ML und QML Modelle (QSVM, QKRR). Die QML-Modelle basieren auf ad-hoc und generierten Feature-Maps. Determinationskoeffizient der Modelle für a) den California Housing Datensatz und b) Ethen Datensatz. c) Grenzwertoptimierungswert (ROC-AUC) auf dem PLASTICC Datensatz und d) Determinationskoeffizient für den QFMNIST Datensatz. e) Beispiel Feature-Map basierend auf dem RL-Algorithmus für den California Housing Datensatz.

Die Ergebnisse sind in Abb. 47 zu sehen. Diese ermöglichen einen Überblick über die Bestleistung (horizontale Abschlusslinie) und über die Durchschnittsleistung (horizontale Linie innerhalb der Boxen) der verschiedenen Kategorien. Wir teilen hierbei in die Kategorien klassisches ML (SVM, KRR, Neuronales Netz, usw.), QML-Methoden mit ad-hoc Feature-Maps und QML-Methoden mit vom oben beschriebenen Algorithmus generierter Feature-Map auf. Bei den QML-Methoden werden jeweils die QSVM und die QKRR mit verschiedenen Feature-Map-Architekturen (ad-hoc und vom MuZero-Algorithmus generiert) verglichen. Dies ermöglicht einen Eindruck über die Robustheit der generierten Feature-Map-Architekturen zu gewinnen.

Abb. 47 a)-d) zeigt die Box-Plots für die oben genannten Testdatensätze. Abb. 47e) zeigt exemplarisch eine der generierten Quanten-Feature-Maps für den Fall des California Housing Datensatzen.
In Abb. 47 a) lässt sich erkennen, dass die beste Leistung von der QSVM oder QKRR mit generierten Feature-Maps erreicht wird. Ebenfalls bemerkenswert ist, dass die Streuung der Ergebnisse für verschiedene generierte Architekturen sehr gering ist. Dies lässt schließen das der RL-Algorithmus sehr gut in der Lage ist problemspezifische Architekturen zu entwerfen. Abb. 47 b) zeigt die Ergebnisse auf dem Ethen Datensatz, hier ist zu erkennen, dass die QML-Methoden die klassischen Verfahren nicht übertreffen. Im Vergleich der QML-Methoden schneiden jedoch die entworfenen Architekturen wieder deutlich besser ab als die ad-hoc Ansätze. Dies lässt schließen das problemspezifisches Modelldesign sehr wichtig ist und mittels des hier entworfenen Ansatzes geling. Für den Fall des PLASTICC Datensatzes (Abb. 47c)) bildet sich dasselbe Verhalten ab wie in Abb. 47 b). Bemerkenswert hier ist die deutlich geringere Streuung in der Leistung im Vergleich ad-hoc QSVC – MuZero QSVC. In Abb. 47 d) sind die Ergebnisse auf dem QFMNIST Datensatz gezeigt. Auch hier zeigt sich ein Vorteil der generierten Feature-Maps gegen über den ad-hoc Feature-Maps. Diese schneiden sogar besser als die klassischen Methoden ab. Da der Datensatz explizit quantenmechanischer Natur ist, ist dies jedoch nicht überraschend. Bemerkenswert ist der generierten Feature-Maps auf dem IBM Quantencomputer. Diese ist mit genau so gut wie die der simulierten Methode. Die Hardwareergebnisse werden im Folgenden genauer betrachtet. Der hier erbrachte Vergleich schließt Meilenstein M15 ab.

Anwendung auf realen Quantencomputern

Der in diesem Arbeitspaket erstellte Algorithmus zur Generierung von Feature-Maps wurde von Grund auf so designed, dass die Übertragbarkeit des Trainings in der Simulation auf Rechnungen mit realen (NISQ) Quantencomputern möglichst reibungslos verläuft. Dies wurde im Wesentlichen durch drei Punkte gewährleistet: 1) Bestrafung längerer Schaltkreise in der Belohnungsfunktion und somit Sicherstellen, dass möglichst flache und somit hardwarefreundliche Schaltkreise bevorzugt werden. 2) Wahl eines hardwareeffizienten Gatesets zur Minimierung von zusätzlicher Schaltkreistiefe durch Transpilation. 3) Fokus auf Projected Quantum Kernel, die sowohl eine kürzere Schaltkreistiefe im Vergleich zu Fidelity Kernel erlaubt als auch eine effizientere Nutzung der (teuren) Quantenressourcen, da die Anzahl an Messungen linear und nicht quadratisch (wie im Falle der Fidelity Kernel) mit der Größe des Datensatzes anwächst.
Im Projekt wurde die beste zur Verfügung stehende ad-hoc Architektur aus der aktuellen Forschung für die jeweiligen Problemstellungen ermittelt. Im Anschluss wurden für diese, und auch die besten generierten Architekturen Berechnungen auf der IBM Quantencomputern durchgeführt.

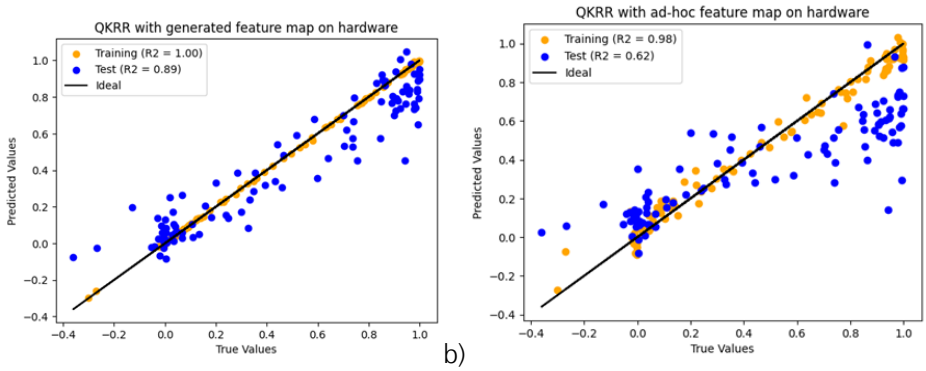


Abb. 48 Rechnungen auf IBM Q Quantencomputer: a) Ergebnis der Quanten Kernel Ridge Regression aufbauend auf der automatisch generierten Feature-Map (**Abb. 50** c)). b) Ergebnis der Quanten Kernel Ridge Regression mit der besten ad-hoc Feature-Map. c) d) Gleiche Rechnungen wie in a) und b) in fehlerfreien Simulationen.

Abb. 48 a) zeigt exemplarisch das Ergebnis der Berechnungen auf dem IBM System Q-Quantencomputer mit dem QFMNIST Datensatz. In Simulation wurden hier für die ad-hoc Feature-Map eine Testgenauigkeit von 0.91 erreicht und für die generierte Feature-Map eine Testgenauigkeit von 0.97. Dies zeigt, dass im Vergleich zur Simulation die Testgenauigkeit abnimmt. Damit ist aufgrund der Fehlerbehaftung der realen Quantencomputer jedoch zu rechnen. Allerdings ist die Abnahme sehr gering, was für die Robustheit der vom RL-Algorithmus kreierten Feature-Map-Architekturen spricht. Dies ist im Wesentlichen auf die oben beschriebenen Designkriterien zurückzuführen. Abb. 48 b) zeigt das Ergebnis der Hardwareberechnung der besten ad-hoc Architektur [7] für dieses Problem. Die Dominanz des Modells auf Basis der generierter Feature-Maps überträgt sich von der Simulation auf die echten Quantencomputer.

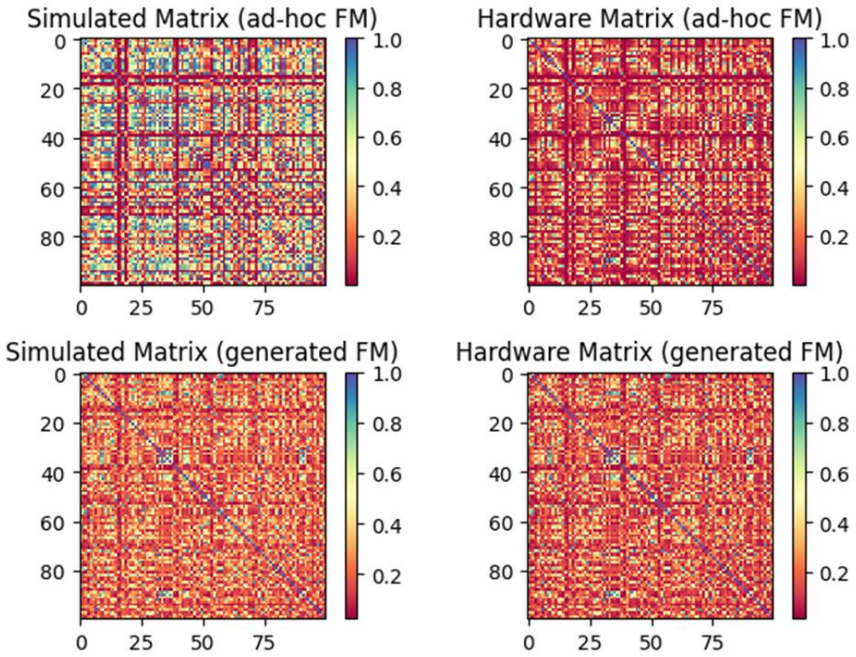
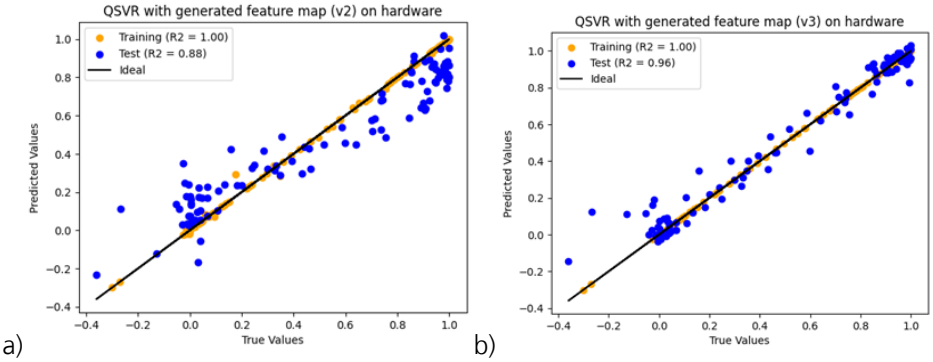


Abb. 49 Visualisierung der Kernel Matrizen. Oben: Kernel Matrix einer ad-hoc Quanten Feature-Map auf Basis einer Simulation (links) und einer Rechnung auf einem echten IBM Quantencomputer (rechts).
Unten: Kernel Matrix einer vom RL-Algorithmus generierten Kernelmatrix.



a) b)

c)

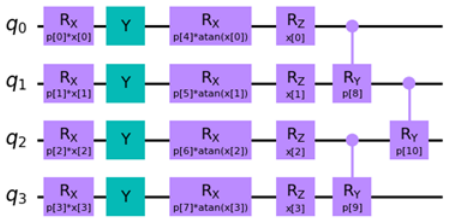


Abb. 50 Ergebnisse der Rechnungen auf IBM Q Quantencomputer a) Ergebnis einer QSVR aufbauend auf einer weiteren Version (V2) der künstlich generierten Feature-Map. b) Ergebnis einer QSVR aufbauend auf einer weiteren Version (V3) der künstlich generierten Feature-Map. c) Version 3 der generierten Feature-Map mit bester Leistung auf der Hardware

Zur Analyse des Verhaltens auf echten Quantencomputern wurden die jeweiligen Kernelmatrizen direkt betrachtet (siehe. Abb. 49). Die Berechnung dieser Matrix ist der essenzielle Teil einer jeden Quanten Kernel Methode und bestimmt maßgeblich die Performance. Aus Abb. 49 lässt sich erkennen, dass durch den Übergang von Simulation auf die Hardware, bedingt durch das Rauschen auf der Hardware, die Werte der Matrizen etwas gedämpft werden und dadurch niedriger sind. Dies gilt für ad-hoc und generierte Ansätze. Für die Performance des finalen quanten-klassischen Modells ist jedoch weniger die absoluten Werte der Matrixelemente als vielmehr die relative Größe entscheidend. Es ist gut sichtbar, dass die allgemeine Struktur auch mit Berechnung auf realen Quantencomputern erhalten bleibt. Diese Beobachtung ist auch für andere Datensätze konsistent. Hier zeigt sich der Vorteil der Projected Quantum Kernel, die gegenüber den Fidelity Kernel eine um die Hälfte reduzierten Schaltkreistiefe und somit weniger Fehler/ Dämpfung in den Matrixelementen aufweisen.

Abb. 50 a-b) zeigt das Ergebnis einer QSVM auf dem QFMNIST Datensatz mit zwei weiteren generierten Feature-Maps. Insbesondere das Ergebnis in Abb. 50 b) ist hervorzuheben, hier wurde die bestmögliche Leistung auf der Hardware für dieses Problem erzielt, die zugehörige Architektur ist in Abb. 50 c) dargestellt. Hier stellt sich heraus, dass der Erfolg dieser Architektur in ihrem simplen Aufbau, d.h. der besonders kurzen Schaltkreistiefe und der geringen Anzahl an 2-Qubit Gates liegt. Dadurch ist die resultierende Fehlerrate gering. Dies zeigt das der auf dem MuZero-Algorithmus basierende Algorithmus sehr gut in der Lage ist problemspezifische Feature-Map Architekturen zu generieren, die ebenfalls auf der Hardware performante Ergebnisse liefern können.

Zusammenfassung und Verwertung

Zusammenfassend wurde ein RL-Algorithmus entwickelt, der es ermöglicht eine für ein vorliegendes Problem zugeschnittene Feature-Map zu erstellen. Die Performance der dadurch resultierenden maßgeschneiderte QML-Modelle übertrifft in unseren Tests immer die ausgewählten Feature-Maps aus der Literatur und performen in manchen Fällen sogar besser als die klassischen ML-Methoden. Die Methode liefert sowohl in Simulationen als auch auf supraleitenden Quantencomputern gute Resultate. Die Ergebnisse wurden in einer Publikation aufbereitet, die bereits als Preprint einsichtbar ist und in den kommenden Wochen bei einem Journal eingereicht wird (siehe Kapitel 3 bzw. Referenz [8]). Der Algorithmus und die Resultate wurden in zahlreichen Vorträgen demonstriert, u.a. auf dem APS March Meeting in Minneapolis am 7.3.2024. Ein Demonstrator wurde erarbeitet und zur Verfügung gestellt.

Referenzen

[1] https://proceedings.neurips.cc/paper/2021/file/69adc1e107f7f7d035d7baf04342e1ca-Paper.pdf "The Inductive Bias of Quantum Kernels".
[2] https://openai.com/research/openai-baselines-ppo "Proximal Policy Optimization".
[3] https://www.nature.com/articles/s41586-020-03051-4 "Mastering Atari, Go, chess and shogi by planning with a learned model".
[4] https://www.nature.com/articles/s41467-021-22539-9 "Power of Data in Quantum Machine Learning".
[5] https://arxiv.org/abs/2101.11020 "Supervised Quantum Machine Learning Models are Kernel Methods".
[6] Shaydulin and Wild, "Importance of kernel bandwidth in quantum machine learning" https://journals.aps.org/pra/abstract/10.1103/PhysRevA.106.042407
[7] https://journals.aps.org/pra/abstract/10.1103/PhysRevA.106.042431 "Training quantum embedding kernels on near-term quantum computers".
[8] F. Rapp, D. A. Kreplin, M. Roth, "Reinforcement learning-based architecture search for quantum machine learning", arXiv:2406.02717 [quant-ph], 2024

2.2.4 Quantenalgorithmen für die Resilienzanalyse

Quantenalgorithmus zur Resilienzsteigerung von Infrastrukturnetzen durch Identifikation kritischer Netzwerkparameter

Im Rahmen der Entwicklung eines Quantenalgorithmus zur Resilienzsteigerung von Infrastrukturnetzen wurde ein Konzept für den Quanten-Optimierungsalgorithmus entwickelt. Als kritischer Netzwerkparameter dient die Gesamtperformanz des Systems, die sich aus der Anzahl funktionierender Knoten im Gesamtnetz ergibt. Knoten stellen hierbei relevante Komponenten des kritischen Infrastrukturnetzes dar. Einzelne initiale Ausfallwahrscheinlichkeiten von Knoten sowie Übertragungswahrscheinlichkeiten von Störungen von einem Knoten zum anderen sind jene Netzwerkparameter, die variiert werden, um die Gesamtperformanz des Systems zu maximieren.

In der QC-Repräsentation des Netzwerks werden die Netzwerkknoten durch einzelne Qubits repräsentiert, ihre Ausfallwahrscheinlichkeit durch eine entsprechende Projektion auf den Zustand $|1\rangle$. Die Netzwerkparameter (initiale Ausfallwahrscheinlichkeit, Übertragungswahrscheinlichkeiten von Störungen, s.o.) werden durch U-Gates bzw. durch kontrollierte U-Gates dargestellt. Die Variation der Netzwerkparameter erfolgt durch weitere, sogenannte „Kontroll-Qubits", die je nach Zustand einen bestimmten Netzwerkparameter variieren. Ein Impactregister misst die Gesamtauswirkung einer Störung im Netzwerk. Ein weiterer Qubit, hier „Indikator-Qubit", bildet ab, falls ein festgesetzter Schwellwert überschritten wird, der einen nicht tolerierbaren Performanzverlust darstellt. Eine genauere Einführung des Ansatzes ist in [1] zu finden.
Zur Netzwerkoptimierung wird nach jenem Netzwerkparameter gesucht, der durch leichte Variation den größten Einfluss auf die Wahrscheinlichkeit hat, den Schwellwert zu überschreiten. Dazu wird der Zustand des „Indikator-Qubits" mithilfe der Quantum Amplitude Estimation (QAE) bestimmt. Dazu wird ein auf einer adaptiven Groversuche basierender Optimierungsalgorithmus verwendet (siehe Abb. 51 Ergebnisse der Groveruche für jeden Schritt des Optimierungsalgorithmus für verschiedene Auflösungen

(5 oben, 6 Mitte, 7 unten). Dazu ist die Wahrscheinlichkeit, dass ein kritischer Netzwerkzustand auftritt (Schwellwert überschritten) für jeden Optimierungsschritt und zugehörigem, variiertem Parameter angegeben. Ein Anstieg der Wahrscheinlichkeit beendet die Optimierung. Der im Schritt zuvor variierte Parameter wird als der Parameter mit dem größten Einfluss auf Gesamtperformanz des Netzes identifiziert (hier P_1).). Mithilfe des Algorithmus kann für das unter Abschnitt 2.1.4 dargestellte Beispielnetz der Parameter mit dem größten Einfluss auf Gesamtperformanz des Netzes erfolgreich ermittelt werden.

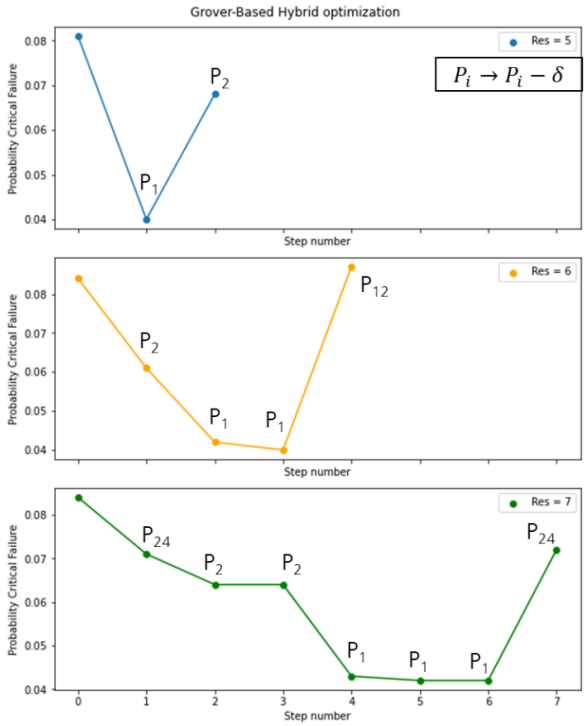


Abb. 51 Ergebnisse der Groveruche für jeden Schritt des Optimierungsalgorithmus für verschiedene Auflösungen (5 oben, 6 Mitte, 7 unten). Dazu ist die Wahrscheinlichkeit, dass ein kritischer Netzwerkzustand auftritt (Schwellwert überschritten) für jeden Optimierungsschritt und zugehörigem, variiertem Parameter angegeben. Ein Anstieg der Wahrscheinlichkeit beendet die Optimierung. Der im Schritt zuvor variierte Parameter wird als der Parameter mit dem größten Einfluss auf Gesamtperformanz des Netzes identifiziert (hier P_1).

Referenzen

[1] M.C. Braun, T. Decker, N. Hegemann, S.F. Kerstan and C. Schäfer. A Quantum Algorithm for the Sensitivity Analysis of Business Risks. arXiv, 2021.

Implementierung von inkohärenten Quantenübergängen in Drei-Niveau-Systemen durch Projektion kohärenter Dynamiken in erweiterten Hilberträumen

To simulate the dynamics of the quantum network on QC, we embedded our open system into an extended Hilbert space where the evolution is unitary [1]. We used a particular realization of this method which was developed in Ref. [2] and which is based on Sz.-Nagy dilation theorem [3]. As compared to alternative methods [4], the approach of [2] uses quantum resources more economically: the unitary extension's dimension is

only twice as large as that of the corresponding open system, i.e., it equals 54 and, hence, can be simulated by 6 qubits.

The first task was to find the operator-sum representation of the solution of (1):

$$\rho(t) = \sum_n M_n(t)\rho(0)M_n^\dagger(t), \quad \sum_n M_n^\dagger(t)M_n(t) = id, \tag{2}$$

where $M_n(t)$ are the Kraus operators, and id is the identity operator. The effective QC-mapping requires an analytical form of the Kraus operators, which can be obtained once the analytical expressions are known for the eigenvalues, $p_n(t)$, and the eigenvectors, $|\psi_n(t)\rangle$, of the network's density operator $\rho(t)$. Then the Kraus operators can be straightforwardly deduced via the recipe developed in Ref. [5].

We have solved this problem for three cases of increasing complexity. For simplicity, it was assumed that the external coherent driving is absent. The assumption is made in order to limit the total initial number of excitations in the network to one. This allows to find the Kraus operators analytically for a three-node system.

First, we studied the non-unitary dynamics of a single node prepared initially in the excited state. In this case, the Hilbert space is spanned by the vectors $\{\{|0\rangle, |1\rangle, |2\rangle\}$ and the Kraus operators are 3×3 matrices that can be found exactly. The next system that was studied included two nodes: one three-level and one two-level system. The effective Hilbert space of such system, prepared initially in state $\{|1,0\rangle$ (that is, the first node in the excited and the second in the ground state is four dimensional, spanned by vectors $\{|0,0\rangle, |0,1\rangle, |1,0\rangle, |2,0\rangle\}$. The Kraus operators in this case are 4×4 matrices, but their derivation in analytical form is relatively straightforward. In particular, the eigenvalues $p_n(t)$ of the density operator that are required for the assessment of the Kraus operator can be found from a quadratic equation. Finally, we considered the network which consists of one damageable and two ideal nodes (that is, without the defect level). Then the effective Hilbert space reduces to a 5-dimensional one, spanned by the vectors $\{|0,0,0\rangle, |0,0,1\rangle, |0,1,0\rangle, |1,0,0\rangle, |2,0,0\rangle\}$. In this case, all roots of the characteristic equation and the corresponding eigenvectors were found exactly as solutions of a cubic equation with the aid of the Cardano formulas, and the 5×5 Kraus operators $M_n(t), (n = 1, \dots, 5)$ were determined analytically for arbitrary times.

From the analytical formulas for $M_n(t)$, their unitary dilations $U_n(t)$ were derived. Since $\dim M_n(t) = 5$, we have $\dim U_n(t) = 10$. Therefore, four qubits spanning a 16-dimensional Hilbert space are sufficient to simulate the simplified network. To that end, the matrices $U_n(t)$ were transformed to 16-dimensional ones, $\widetilde{U}_n(t)$, by appending to them 6×6-dimensional identity matrices as diagonal blocks and padding the off-diagonal blocks with zeros. Thereafter, the initial density operator $\rho(0)$ were decomposed in terms of pure states, $\rho(0) = \sum_k c_k\ |\phi_k(0)\rangle\langle\phi_k(0)|$. We then expanded vectors $|\phi_k\ (0)\rangle$ to 16-dimensional vectors $|\tilde{\phi}_k\ (0)\rangle$ by appending to them 11-dimensional null vectors. With all unitaries and all initial vectors defined, we implemented the unitary evolutions, $\widetilde{U}_n(t)|\tilde{\phi}_k\ (0)\rangle$, on a quantum computer for every n and k.

References

[1] R. Sweke, I. Sinayskiy, D. Bernard, and F. Petruccione, Universal simulation of markovian open quantum systems. Phys. Rev. A, 91:062308, 2015.
[2] Z. Hu, R. Xia, and S. Kais, A quantum algorithm for evolving open quantum dynamics on quantum computing devices. Scientific Reports, 10(1):3301, 2020.
[3] E. Levy and O. M. Shalit, Dilation theory in finite dimensions: The possible, the impossible and the unknown. Rocky Mountain Journal of Mathematics, 44(1):203, 2014.

[4] M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK, 2000.
[5] D. M. Tong, L. C. Kwek, C. H. Oh, J.-L.Chen, and L. Ma, Operator-sum representation of time-dependent density operators and its applications. Phys. Rev. A, 69:054102, 2004.

Übersetzung stochastischer Prozesse in Quantentrajektorien zur topologischen Robustheits- und Resilienzanalyse von gekoppelten Infrastrukturnetzen

While the density matrix of an open quantum system furnishes the complete information about its properties, the evolution of such a system is non-unitary. In general, mapping the non-unitary dynamics onto a QC via a unitary dilation is challenging and so far, we have been able to simulate on a QC only small, three-node networks with defects, after some simplifications. We therefore explored the potential of an alternative approach, called "quantum trajectories", which is based on unraveling the density matrix evolution into an ensemble of pure state's evolutions [6,7]. Using quantum trajectories has allowed us to study, without any simplifying assumptions, the four-node quantum networks with defects. We started with eq. (1) (generalized to 4 nodes), which we decomposed into two parts as follows:

$$\dot{\rho} = -i\left[H_{eff}, \rho\right] + J\rho, \qquad (3)$$

where $H_{eff} = H - i\sum_{\alpha=1}^{4}\sum_{m=1}^{2} L_{\alpha m}^{\dagger} L_{\alpha m}$ is the effective non-Hermitian Hamiltonian generating the continuous evolution of the network's state, and $J\rho = 2\sum_{\alpha=1}^{4}\sum_{m=1}^{2} L_{\alpha m}\rho L_{\alpha m}^{\dagger}$ is the generalized jump operator describing all possible instantaneous incoherent transitions in the 4-node quantum network. Between the jumps, the system's conditioned state $|\psi_c(t)\rangle$ obeys the Schrödinger equation, $\frac{d}{dt}|\psi_c(t)\rangle = iH_{eff}|\psi_c(t)\rangle$. When a random jump occurs at random time t', the modified state is given by $|\psi_c(t')\rangle = \sqrt{2}L_{\alpha m}|\psi_c(t')\rangle / ||\sqrt{2}L_{\alpha m}|\psi_c(t')\rangle||$. Random instants and types of jumps were generated by the Monte Carlo method.

We studied how the performance and resilience of quantum networks is affected by their geometrical structure, i.e., by how the nodes are connected to each other. By analogy with classical networks representing critical infrastructure, we defined the performance of the quantum network as the number of nodes in their qubit states, that is, $|1\rangle$ or $|2\rangle$, versus time. The maximum performance of four nodes equals to 4 which means that neither of the nodes occupies state $|3\rangle$, whereas when one or more nodes sit on the defect level, the performance drops accordingly. Then the network resilience is defined as the average perfomance integrated over time.

Our results for two network configurations with five links are shown in Abb. 52.

These results show strong sensitivity of the network's resilience to its geometric structure which stems from quantum interference. In Abb. 52 a), the structure is symmetric with respect to the laser-driven node. Therefore, interference effects prevail and effectively suppress the transfer of excitation to other nodes, resulting in relatively few jumps and high resilience. In contrast, for the configuration that is asymmetric with respect to the laser driven node (Abb. 52 c)) interference is imperfect, which leads to the effective population of state $|2\rangle$ of all nodes. In this case, not only jumps on individual nodes become more frequent, but also cascades of 3 or 4 jumps on different nodes occur.Apart from the resilience, other important quantities in network's analysis are the waiting time distributions for failures of one or more nodes and the conditional probabilities for cascaded failures. These quantities can be deduced from long enough trajectories monitoring the number of nodes "ON". Here, we report our results on the waiting time distributions.

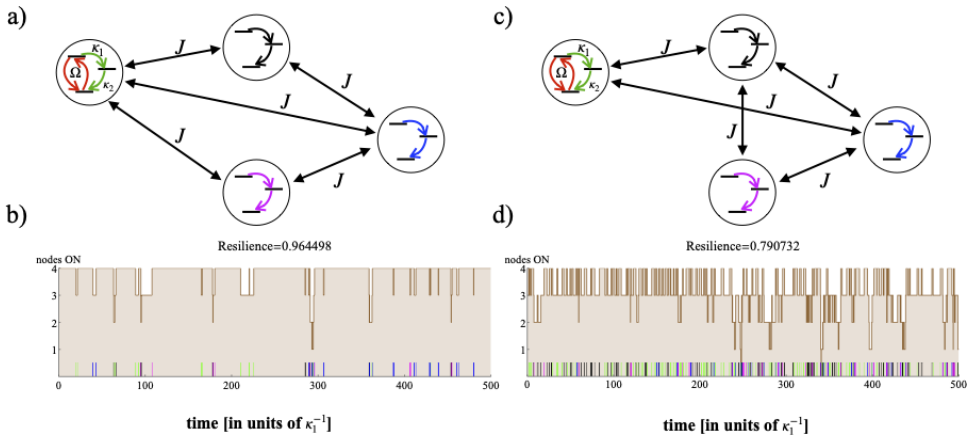


Abb. 52 (a,c) Distinct configurations of three-level nodes coupled by five links with equal strength J. For better visibility the incoherent transitions within a single node are depicted by different colors, and rates κ_1 and κ_2 are the same for each node. One node is excited by a laser field with strength Ω. (b) Performance (nodes ON) versus time (in units of κ_1^{-1}) for configuration (a). The number of jumps (colored vertical lines at the bottom) is relatively rare, yielding the high resilience (the area of the light brown figure) > 0.96. (d) Same for configuration (c). The number of jumps increases substantially, resulting in the drop of the resilience below 0.8. Trajectories of the performance (b,c) were generated with fixed parameters: κ_1=1, κ_2=0.4, J = 5, Ω = 2.

Form the records of the instants when one, two, three, or four nodes are damaged and subsequently repaired, we deduce the time intervals between the corresponding successive events, which are nothing but the random *waiting times*. To obtain the distributions of the latter, we need to run long enough quantum trajectories. Then the numerical waiting time distributions can be visualized by the histograms shown in Abb. 32, which indicate the number of occurrences when the random variable τ falls in a given bin, for the four types of failures. Although we present our exemplary results for one network configuration with six links, the results for the fully connected network, which also reflect the qualitative behavior of the waiting time distributions in other studied cases (e.g., four-node networks with five or four links), are not included in the present report.

The numerical waiting time distributions (yellow bars) can be fitted analytically (blue lines) by the *Log-normal* (Abb. 53 b,c)) or *Weibull* (Abb. 53 a,d)) probability distribution functions. The Log-normal and Weibull distribution functions, respectively, are described by the formulas

$$P_{(\mu,\sigma)}(x) = \begin{cases} \frac{e^{-(\mu-\log x)^2/2\sigma^2}}{\sqrt{2\pi}x\sigma}, x > 0 \\ 0, \ x \leq 0 \end{cases}, \tag{4}$$

$$P_{(\alpha,\beta,\mu)}(x) = \begin{cases} (x-\mu)^{\alpha-1}e^{-((x-\mu)/\beta)^{\alpha}}, x > 0 \\ 0, \ x \leq \mu \end{cases}, \tag{5}$$

with the fitting parameters indicated in the caption to Abb. 53. The number of occurrences of cascades of 3- and 4-node failures is relatively rare for the given length *N* of the trajectory, leading to the discrete, strongly fluctuating distributions of the bars in c) and d); hence, the analytical fits can change once the number *N* increases and/or averaging over the trajectories is performed.

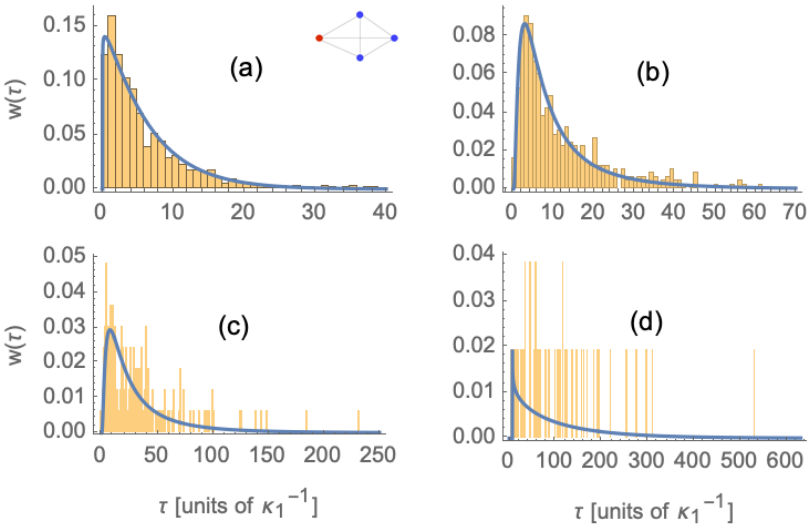


Abb. 53 Probability distributions $w(\tau)$ of waiting times τ between successive failures of (a) one, (b) two, (c) three, and (d) four nodes. The histograms (yellow bars) were derived from a single quantum trajectory of the number of nodes ON (see, e.g. Abb. 31), for the configuration indicated in the central top part of the figure. Each node is characterized by rates $\kappa_1 = 1$, κ_2=0.4, each link has coupling $J = 5$, the driving strength is $\Omega = 5$ (the driven node is highlighted in red). The trajectory consists of $N = 3 \times 10^6$ time steps of duration $dt = 0.001$ (in units of κ_1^{-1}). Blue lines represent analytical fits (see equations (4,5)): (a) Weibull ($\alpha = 1.04536, \beta = 6.20144, \mu = 0.07941$) , (b) Log-normal ($\mu = 2.02305, \sigma = 0.998914$), (c) Log-normal ($\mu = 3.10679, \sigma = 1.04205$), (d) Weibull ($\alpha = 0.899029, \beta = 104.412, \mu = 6.924$).

We note that the Log-normal distribution is categorized as the distribution with a "fat" tail, such that for large values of the waiting time the probability decreases as a power law rather than exponentially. As for the Weibull distribution, it is generally used to describe rare events, while its tail can be fat or thin depending on the distribution's parameters. Both the Log-normal and the Weibull distributions play an important role in nature, economy, finance, and risk analysis. Therefore, the relevance of these functions in analysis of the waiting time distributions of failures in a small quantum network with defects is interesting and deserves further studies.

Presently, we are systematizing the obtained results and attempting to deduce the waiting time distributions and conditional probabilities for jumps in quantum networks with defects. Furthermore, the method of quantum trajectories has been successfully used in quantum algorithms dedicated to study the dynamics of open systems on a QC [8]. Therefore, it would be interesting to consider this approach as an alternative to the unitary dilation method in future extensions of the project.

References

[6] H.J. Carmichael, An open system approach to quantum optics, Springer-Verlag, Berlin, 1993.
[7] J. Dalibard, Y. Castin, K. Mølmer, Wave-function approach to dissipative processes in quantum optics, Phys. Rev. Lett. 68:580, 1992.
[8] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, Variational quantum simulation of general processes, Phys. Rev. Lett. 125:010501, 2020.

2.2.5 Algorithmen zur Konfigurationspriorisierung

In AP2.1 wurde QAOA als Quantenalgorithmus zur Problemlösung für die Konfigurationspriorisierung (ein Problem aus dem Bereich der kombinatorischen Optimierung) ausgewählt. QAOA [1] ist ein hybrider Approximationsalgorithmus und benutzt Quantenschaltkreise, die durch die Winkel $\beta_1 \dots \beta_p$ und $\gamma_1 \dots \gamma_p$ parametrisiert sind. Die Schaltkreise werden in der uniformen Superposition mit Hadamard Gates initialisiert. Darauf folgen p Abfolgen (Schichten) von Phasen-separierenden und mischenden Operatoren. Die Phasen-separierenden Operatoren enkodieren die Kostenfunktion und sind mit $\gamma_1 \dots \gamma_p$ parametrisiert. Die mischenden Operatoren ändern die Amplituden der Lösungen und sind mit $\beta_1 \dots \beta_p$ parametrisiert. Der Algorithmus approximiert Lösungen für kombinatorische Optimierungsprobleme, indem klassische Optimierer verwendet werden, um diese Parameter in den parametrisierten Quantenschaltkreisen in einem iterativen Vorgehen zu optimieren.

Folgende Ergebnisse sind Teil der SEQUOIA-Veröffentlichung von Ammermann et al. [2]:

QAOA-Schaltkreise für das Konfigurationspriorisierungsproblem können mit der bereitgestellten Python-Bibliothek [3] automatisiert für konkrete Probleminstanzen generiert werden. Um ein besseres Verständnis für die Probleminstanzen zu bekommen, kann die Bibliothek Visualisierungen der Optimierungslandschaft erstellen. Für die Probleminstanz mit 6 Variablen $x_1, \dots, x_6$ (d.h., 6 Features $x_1, \dots, x_6$), welche durch die aussagenlogische Formel $(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_3 \vee \neg x_5 \vee \neg x_6)$ und den zugeordneten Kosten $c_1 = 30, c_2 = 20, c_3 = 25, c_4 = 50, c_5 = 10, c_6 = 10$ beschrieben wird, können folgende Visualisierungen erstellt werden:

Abb. 54 zeigt die Optimierungslandschaft dieser Instanz für verschiedene QAOA-Parameter β_1 und γ_1 (da $p = 1$) nach [4]. Diese Optimierungslandschaften zeigen, welche Parameterauswahlen optimal für diese Instanz sind. Gewonnen Erkenntnisse über die Parameter kleiner Probleminstanzen können somit zur Entwicklung einer Heuristik für eine geeignete Wahl der QAOA-Startparameter genutzt werden. Bei einer geeigneten Parameterauswahl misst man aus dem QAOA-Quantenschaltkreis somit minimale Konfigurationen mit hoher Wahrscheinlichkeit und kann die Konfigurationen anhand der gemessenen Anzahl priorisieren.

Abb. 55 zeigt ein $\mu - f$ Diagramm nach [5], welche Rückschlüsse über die Eignung der vorliegenden Probleminstanzen für die ausgewählten Algorithmus (QAOA) ermöglicht. Ein $\mu - f$ Diagramm strukturiert Konfigurationen nach der Energie f einer Konfiguration (Auswertung von dieser Konfiguration auf Gesamtproblem-Hamiltonian) und dem arithmetischen Mittel μ der Differenz in Energie zwischen der aktuellen Konfiguration und allen Konfigurationen mit einer Hamming-Distanz von 1 (die nächstgelegenen Nachbarn). Die Darstellung dient als Indikator für die quantitative Struktur der Optimierungslandschaft – Anzahl, Größe und Tiefe von „Tälern" (vorhanden bei $\mu \geq 0$). Die Optimierungslandschaft einer Probleminstanz kann somit als günstig oder ungünstig für lokale Suchroutinen wie QAOA eingeordnet werden, günstig ist hierbei eine „thin tail" Struktur (vgl. [5]).

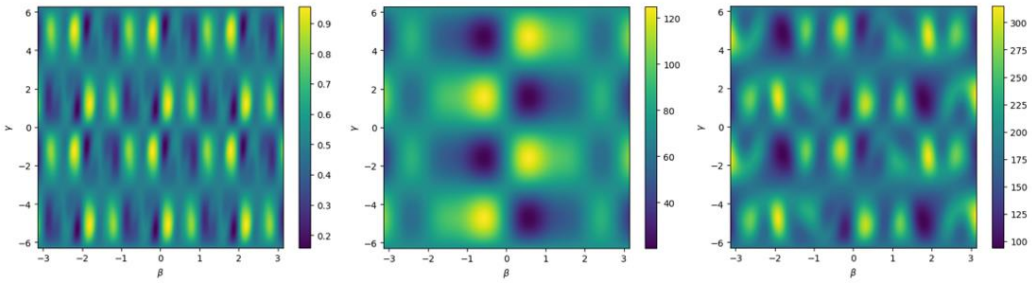


Abb. 54 Optimierungslandschaft für verschiedene QAOA-Parameter β_1 und γ_1 nach [4] für eine Probleminstanz des Konfigurationspriorisierungsproblems mit 6 Variablen (Features) für die Featuremodell-Hamiltonian H_C (l.), den Kosten-Hamiltonian H_{c_k} (m.) und den Gesamtproblem-Hamiltonian $H_{c_{tot}} = H_{c_k} + \alpha H_C$ (r.) mit $\alpha \approx 200$ (siehe Abschnitt 2.1.6). Ein Minimum der Probleminstanz befindet sich z.B. bei $\beta \approx 0.2$ und $\gamma \approx -1$.

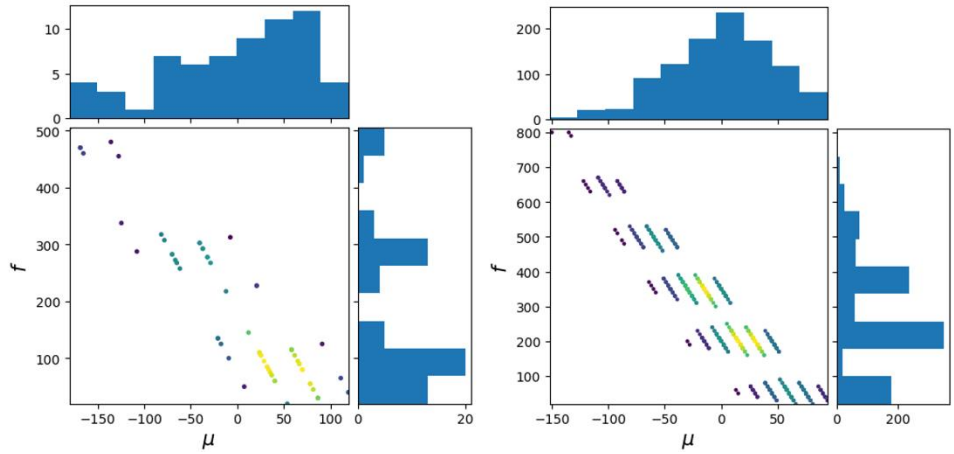


Abb. 55 Optimierungslandschaft als $\mu - f$ Diagramm nach [4] für eine Probleminstanz des Konfigurationspriorisierungsproblems mit 6 Variablen (Features) (l.) und 10 Variablen (Features) (r.). Vor allem das rechte Diagramm zeigt einen "thin tail", die Verteilung ist also bei den Extrema niedrig und in der Mitte höher bzw. dichter, was ein Indikator für eine geeignetere Optimierungslandschaft für die Verwendung von QAOA ist [5].

Für Skalierbarkeit analysieren wir die Quantenschaltungen, die aus Probleminstanzen in der entsprechenden QUBO / PUBO Formulierung erstellt werden. Tabelle 3 zeigt die Größe der Probleminstanzen und entsprechende Tiefe und Breite des Quantenschaltkreises des Phasen-separierenden Operators. Unser Ansatz erfordert Qubits in Höhe der Anzahl der Variablen / Features bei der PUBO- Formulierung und zusätzliche Qubits bei der QUBO-Formel aufgrund von benötigten Hilfsvariablen. Die Schaltungstiefe skaliert bei QUBO besser als bei der PUBO-Formulierung.

Die Verwendung von verschiedenen klassischen Optimierern hatte kaum Einfluss auf die Ergebnisqualität. Die Laufzeit der vorhandenen Probleminstanzen war mit dem klassischen Optimierer COBYLA im Durchschnitt am geringsten. Vor allem die Verwendung von Warmstarts verbessert die Ergebnisqualität maßgeblich. Die Verwendung von anderen Mixern hatte kaum Einfluss auf die Ergebnisqualität (bei den getesteten kleinen Probleminstanzen). Die QAOA-Hyperparameter p und problemspezifischen Regularisierungsparameter α wurden empirisch für die vorliegenden Probleminstanzen bestimmt.

ID	# features	# clauses	Max # literals	Depth		Width	
				PUBO	QUBO	PUBO	QUBO
0	6	7	6	300	67	6	15
1	6	6	2	17	17	6	6
2	6	9	2	11	11	6	6
3	6	8	2	15	15	6	6
4	6	8	2	15	15	6	6
5	11	20	7	813	123	11	29
6	11	18	2	24	24	11	11
7	11	16	6	314	70	11	20
8	11	15	2	26	26	11	11
9	11	21	4	92	53	11	14
10	16	18	6	314	81	16	25
11	16	18	7	744	156	16	32
12	16	20	4	82	52	16	18
13	16	21	6	327	82	16	26
14	16	24	2	33	33	16	16
15	21	29	2	38	38	21	21
16	21	37	4	85	68	21	27
17	21	36	5	149	65	21	25
18	21	26	3	40	40	21	21
19	21	30	6	328	82	21	32

Tabelle 3 Größe der Probleminstanzen und entsprechende Tiefe und Breite des Quantenschaltkreises des Phasen-separierenden Operators.

Referenzen

[1] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem." arXiv, Jun. 25, 2015. Accessed: Feb. 21, 2023. [Online]. Available: http://arxiv.org/abs/1412.6062

[2] J. Ammermann et al., "Quantum Approach to the Configuration Selection and Prioritization Problems", 2024 ACM/IEEE International Workshop on Quantum Software Engineering (Q-SE 2024), April 16, 2024, Lisbon, Portugal. ACM, New York, NY, USA, *to appear.*

[3] https://github.com/KIT-TVA/qc-configuration-problem

[4] S. Boulebnane and A. Montanaro, "Solving boolean satisfiability problems with the quantum approximate optimization algorithm." arXiv, Aug. 14, 2022. [Online]. Available: http://arxiv.org/abs/2208.06909

[5] G. Koßmann, L. Binkowski, L. van Luijk, T. Ziegler, and R. Schwonnek, "Deep-Circuit QAOA." arXiv, Feb. 03, 2023. [Online]. Available: http://arxiv.org/abs/2210.12406

2.2.6 Szenario-basierte Routenplanung

Die zur Lösung des Traveling Salesperson Problems in Sequoia geschriebene QAOA-Implementierung wurde um die erste Version eines Fehlerkorrekturschaltkreises erweitert. Dabei werden die in der Problemcodierung inhärenten Redundanzen genutzt, um während der Berechnung entstandene Fehler zu erkennen und zu korrigieren. Das hat den Vorteil, dass keine zusätzlichen Qubits zur Codierung eines Zustands notwendig sind, sondern die durch die Problemstellung ohnehin geforderten Qubits zur Definition eines Fehlercodes verwendet werden. Lediglich zur Messung des Fehlersyndroms sind zusätzliche Qubits vonnöten.

Eine umfassende Literaturrecherche ergab, dass bisher kein derartiges Fehlerkorrekturverfahren existiert. Daher wurde eine erste Version entworfen und getestet, inwieweit sie entstandene Fehler bei unterschiedlichen Fehlerraten korrigiert. Die dabei verwendete TSP-Instanz hatte die Größe n=3, sodass insgesamt Codewörter der Länge $n^2=9$ zufällig generiert, verrauscht und korrigiert wurden. Das Resultat ist in Abb. 56 festgehalten:

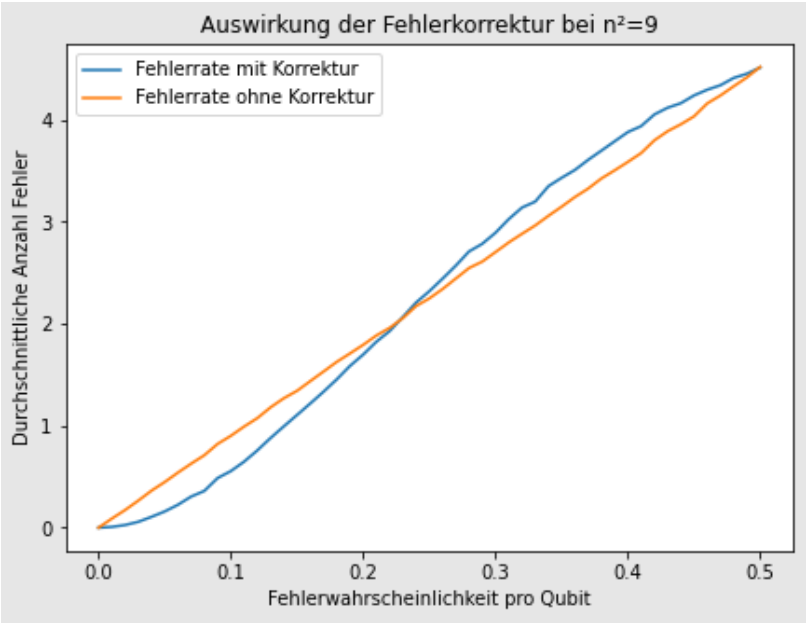


Abb. 56 Fehlerrate zufällig generierter QAOA TSP-Codierungen mit und ohne Fehlerkorrektur bei steigender Fehlerwahrscheinlichkeit pro Qubit.

Auf der x-Achse ist die Wahrscheinlichkeit pro Qubit aufgetragen, dass ein Fehler entsteht. Ist diese Wahrscheinlichkeit 0, ist auch die durchschnittliche Anzahl der Fehler auf der y-Achse 0. Bei einer Fehlerwahrscheinlichkeit von 0.5 hat der resultierende Bitstring $4.5 = \frac{9}{2}$ Fehler im Schnitt. Die getestete Korrektur zeigt für kleine Fehlerraten sichtbare Verbesserungen, verschlechtert allerdings ab einem gewissen Punkt die Ergebnisse sogar. Für starkes Rauschen ist sie also nicht geeignet.

In einem zweiten Schritt wurde das Verfahren als Quantenschaltkreis umgesetzt und dazu benutzt, einen zuvor erstellten Quantenzustand mit mehreren TSP-Touren in Superposition, der durch zufällige Gatter verrauscht wurde, zu korrigieren. Das Verfahren wurde dabei sowohl innerhalb des Schaltkreises als auch beim Postprocessing angewandt, um den gewünschten Zustand wiederherzustellen. Für den Test wurde ein Simulator benutzt, der das von IBM zur Verfügung gestellte Rauschverhalten vom Ehninger Backend miteinberechnet hat. Das Ergebnis ist in der folgenden Tabelle dargestellt:

	Zustand “100010001”	Zustand “001010100”
Ideale Häufigkeit	50%	50%
Ohne Fehlerkorrektur	23.24%	24.02%
Fehlerkorrektur im Schaltkreis	31.45%	32.03%
Fehlerkorrektur im Schaltkreis und im Postprocessing	44.43%	46.39%

Der gewünschte Zustand konnte im Testszenario demnach beinahe vollständig wiederhergestellt werden.

Im dritten Schritt wurde die Fehlerkorrektur in den bestehenden QAOA-Schaltkreis integriert und der gesamte Algorithmus für zufällige TSP-Instanzen ausgeführt. Auch in diesem Fall wurden in einer Simulation unter Rauschen die gewünschten Zustände messbar besser erhalten als ohne Fehlerkorrektur. Allerdings konnten bei Simulationen unter Rauschen keine Verbesserungen in der Ergebnisqualität festgestellt werden. In Ehningen war der erarbeitete Schaltkreis nicht lauffähig. Folglich wurde sich zum Ziel gesetzt, eine Problemcodierung zu entwickeln, die einerseits weniger Qubits benötigt und andererseits größere Kapazitäten zur Fehlerkorrektur bietet.

Infolgedessen wurde eine Problemcodierung für das TSP getestet, bei der anstatt der Sequenz der zu besuchenden Knotenpunkten die dafür verwendeten Verbindungen beschrieben werden. Dadurch wird zum einen die Anzahl verwendeter Qubits reduziert als auch der für QAOA benötigte Phase Separator vereinfacht. Letzteres folgt aus dem Umstand, dass die TSP-Problemformulierung aus einer Matrix mit den Distanzen zwischen Knotenpunkten gegeben ist. Will man die Länge einer Tour berechnen, müssen diese Distanzen also aufaddiert werden. In der ursprünglichen Problemcodierung müssen dazu Paare an Qubits verknüpft werden, um festzustellen, ob die codierte Tour eine bestimmte Verbindung enthält. In der neuen Problemcodierung ist die Verwendung derselben Verbindung direkt durch ein einzelnes Qubit gegeben.

Um die Qualität der Approximationen der beiden Ansätze miteinander vergleichen zu können, wurden 1000 TSP-Instanzen der Größe 4 zufallsgeneriert und von beiden Varianten gelöst. Außerdem wurden mittels eines klassischen Algorithmus die optimalen Lösungen all dieser Instanzen berechnet. Für jede Variante wurden die Ergebnisse mit den optimalen Werten verglichen und die entstandene Differenz, also der Approximationsfehler, in ein Histogramm eingezeichnet.

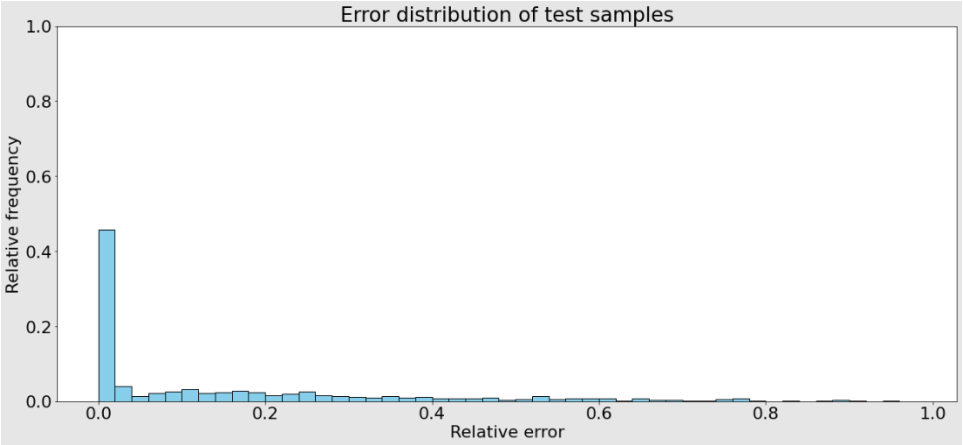


Abb. 57 Verteilung der Approximationsfehler für die ursprüngliche Problemcodierung bei 1000 zufallsgenerierten TSP-Instanzen der Größe 4.

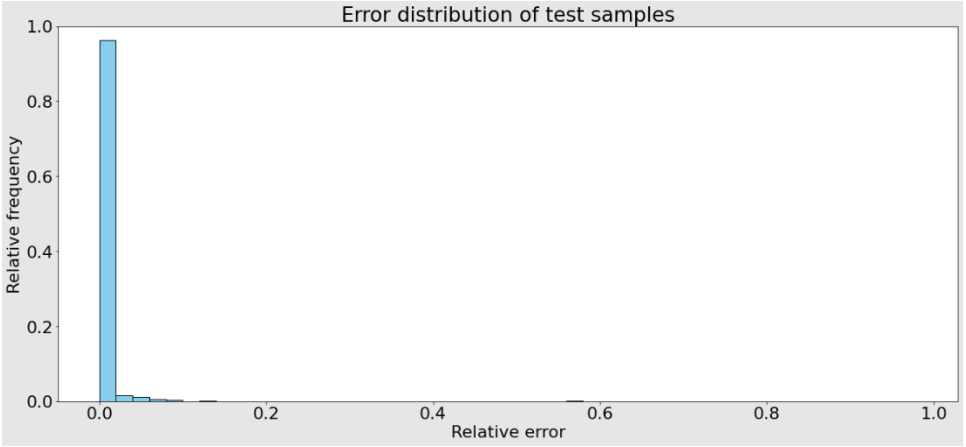


Abb. 58 Verteilung der Approximationsfehler für die neue Problemcodierung bei 1000 zufallsgenerierten TSP-Instanzen der Größe 4**.**

Den beiden Abb. 57 und Abb. 58 ist eindeutig zu entnehmen, dass die Variante, die von der neuen Problemcodierung Gebrauch macht, fast immer die optimale Lösung erzielt, während die herkömmlich codierte Variante in etwa der Hälfte der Fälle suboptimale Touren berechnet. Noch dazu ist die Verteilung bei Letzterer sehr breit, sodass auch extreme Abweichungen von der optimalen Lösung regelmäßig auftreten.

Die Anzahl an Qubits abhängig von der Problemgröße *n* reduziert sich von n^2 auf $(n-1)(n-2)$, was asymptotisch zwar gleich schnell wächst, in diesem speziellen Fall die Anzahl benötigter Qubits aber von 16 auf 6 reduziert. Dies wirkt sich vor allem auf die Geschwindigkeit der Simulation aus, welche von etwa 95 Minuten auf 4,5 Minuten für 1000 zufallsgenerierte Instanzen sinkt. Gleichzeitig ist diese Ersparnis insbesondere der aktuellen NISQ-Ära zuträglich, in der ohnehin nur eine sehr stark begrenzte Anzahl an Qubits zur Verfügung steht.

QCNN-basierte Vorhersagen von Lösungen partieller Differentialgleichungen

Regressionsmethoden werden aufgrund ihrer guten Vorhersagekraft in einer Vielzahl von Bereichen wie z.B. den Materialwissenschaften zur Prognose von Energielandschaften oder dem Finanzsektor zur Vorhersage von Aktienkursen genutzt, um aufwändige und teure direkte Simulationen zu vermeiden. Wir nutzen diese zur Vorhersage von Lösungen partieller Differentialgleichungen (PDEs) mittels Quanten-Neuronaler-Netze (QNN). Numerische Lösungen akzeptabler Genauigkeit zu akzeptablen Kosten zu finden ist noch immer nicht einfach. Oft sind für jedes Problem spezielle Lösungsverfahren notwendig, die genau auf einen Gleichungstyp zugeschnitten sind. Im Gegensatz dazu können neuronale Netze wegen ihrer Fähigkeit direkt aus Datensätzen komplizierte nicht-lineare Bezüge ableiten zu können eine Alternative darstellen, welche es ermöglicht stetige numerische Lösungen mit guter Genauigkeit vorherzusagen. Ein Nachteil tiefer neuronaler Netze ist jedoch der oft immense Rechenaufwand für deren Training, welcher typischerweise auch viel Speicher und schnelle Interprozesskommunikation benötigt. Quantencomputer bieten hier einen interessanten neuen Ansatz, um bessere und einfacher zu trainierende neuronale Netze zu generieren, welche insgesamt weniger Rechenzeit benötigen.

In der vorherigen Phase des Projekts haben wir (das HLRS der Universität Stuttgart) ein Quanten-Convolutional-Neural-Network (QCNN) entwickelt, um 1D-Burgers- und 2D-Poisson-Gleichungen zu lösen. Die (klassischen) Faltungsschichten werden durch

Quanten-Gatter ersetzt, die die Qubits manipulieren und verschränken. Für das (klassische) Pooling wird ein Teil des Netzes gemessen und die Messergebnisse legen die Parameter unitärer Rotationen fest, die auf jeweils benachbarte Qubits angewendet werden. Nichtlinearitäten im QCNN entstehen durch die Verringerung der Anzahl von Freiheitsgraden bei den Messungen der Pooling-Schicht. Faltungs- und Pooling-Schichten werden so hintereinandergelegt, bis die Anzahl der übrigen Qubits die gewünschte Größe erreicht hat. Abschließend wird eine vollständig verbundene Schicht auf alle verbleibenden Qubits angewendet, indem eine einzige unitäre Operation auf allen Qubits ausgeführt wird. Das Endergebnis wird durch Messen einer festgelegten Anzahl von Ausgangs-Qubits erhalten. Wir verwenden einen hybriden Ansatz, bei dem die Messungen im Quanten-Backend durchgeführt werden, während die Optimierung der Parameter des Quantenschaltkreises auf klassische Weise erfolgt. Um das Konvergenzverhalten der klassischen Komponente zu verbessern, wird eine physikinformierte Verlustfunktion testweise in das Netzwerk eingeführt. Abb. 59 zeigt eine detaillierte Darstellung der Struktur des PIQCNN-Modells. In PIQNNs wird die aus dem QNN abgeleitete Vorhersage in die physikinformierte Verlustfunktion $L_p[\boldsymbol{u}]$ übergeben.

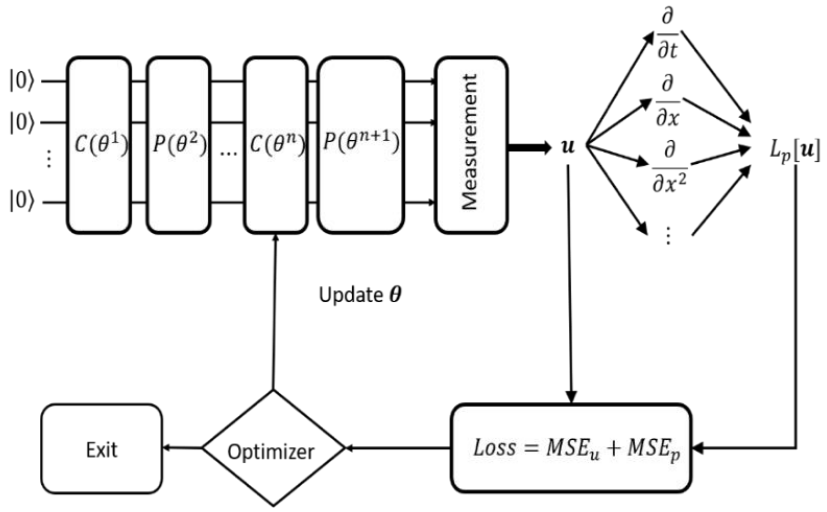


Abb. 59 Detaillierte Darstellung der Struktur des PIQNN

Wir haben solch ein Netz als Regressionsmethode zur Vorhersage der Lösung der verallgemeinerten Poisson-Gleichung genutzt.

$$\nabla \cdot (a(x,y)\nabla u) = -f$$

Die PDE wird auf einem quadratischen zweidimensionalen Gebiet betrachtet. Dabei sind a und der „Quellterm" f auf dem Gebiet gegeben und u wird gesucht. Daher hat die Verlustfunktion für das betrachtete Problem die Form

$$L_p[\boldsymbol{u}] = \frac{\partial u}{\partial t} + \alpha_1 \frac{\partial u}{\partial x} + \alpha_2 \frac{\partial^2 u}{\partial x^2} + \cdots$$

Für den Fall, dass a konstant 1 ist und $\mathrm{f} = 2\pi^2 v$, für v eine der beiden Funtionen

$$v(x,y) = \sin(\pi x)\sin(\pi y),$$

$$v(x,y) = \cos(\pi x)\cos(\pi y),$$

ist die (analytische) Lösung $u = v$ (für Dirichlet- bzw. Neumann-Randbedingungen). Das Ziel ist dann, dass das Netz lernt die Lösung u für gegebenes x, y, f und a vorherzusagen. Trainiert wurde das Netz auf einem mit einem 5x5-Gitter diskretisierten Gebiet und angewandt wurde das Netz zur Vorhersage der Lösung auf einem 11x11-Gitter. Die Eingabedaten wurden betragsmäßig auf 1 normalisiert, ebenso die Trainingsdaten, die vorhergesagten Lösungswerte entsprechend zurück. Abb. 60 zeigt die vom Quanten-Neuronalen-Netz vorhergesagte Lösung und zum Vergleich die analytische Lösung. Das neuronale Netz gibt die Lösung mit einem absoluten Fehler von ca. 1-2% nach den meisten Läufen gut wieder. Das Maximum der vorhergesagten Sinus-Lösung ist typischerweise um 1-3% niedriger als der reale Wert von 1. Bei der Cosinus-Lösung wichen die Werte in den Ecken des Gitters um etwa 5-10% von den realen Werten ab. Dies könnte ein Effekt der Ecken oder ein Artefakt der Skalierung der Trainingswerte sein.

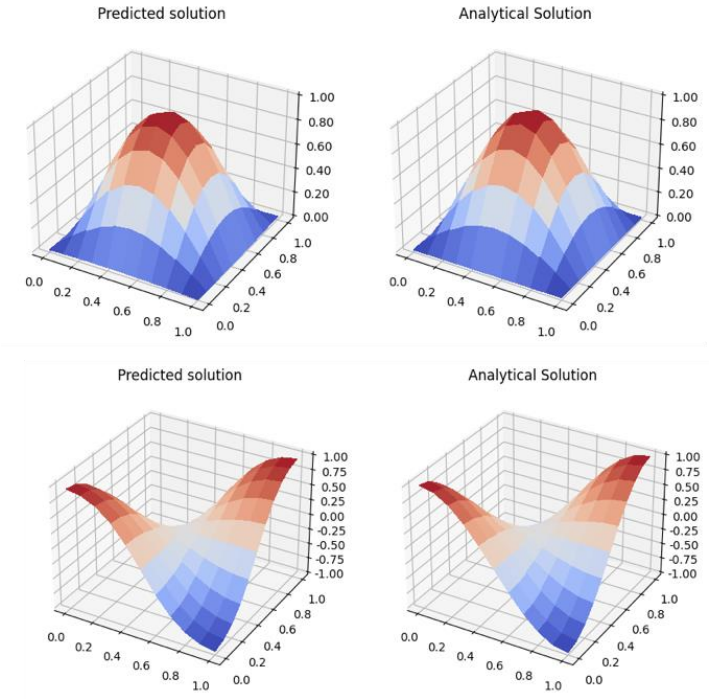


Abb. 60 Vergleich zwischen der klassischen analytischen Lösung (rechts) und der vom QCNN vorhergesagten Lösung (links) der 2D-Poisson-Gleichung als Heat-Map dargestellt.

Der Einfluss der physikinformierten Verlustfunktion auf den Trainingsprozess ist exemplarisch in Abb. 61 dargestellt. Für jeden der drei Trainingsläufe sind die Anfangsbedingungen gleich. Wie deutlich aus den Abbildungen hervorgeht, ist die Konvergenzrate des PIQNN insgesamt besser in diesen Experimenten.

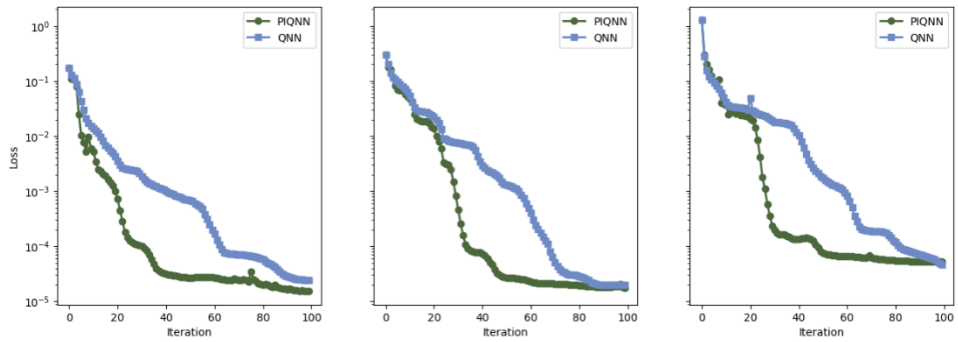


Abb. 61 Vergleich der Konvergenzrate zwischen PIQNN und QNN. In den drei exemplarisch gezeigten Trainingsläufen zeigt das PIQNN (grün) eine bessere Konvergenzrate als das QNN (blau).

ADMM-Algorithmus

Gemischt-ganzzahlige lineare Programme sind eine bedeutende Klasse von Optimierungsproblemen wie sie z.B. bei der Tourenplanung auftreten können, aber auch andere Probleme können als solche dargestellt werden, beispielsweise das im Anwendungsfall **Optimierung von Ladeplänen einer gegebenen Ladesäuleninfrastruktur** betrachtet Problem der Optimierung der Ladung von Elektroautos unter der gegebenen Einschränkung an verfügbaren Ladeplätzen, Netzbelastung, Ladegeschwindigkeit der Autos etc. Während für lineare Probleme mit rein reellen Variablen effiziente klassische Lösungsalgorithmen existieren, ist dies für ganzzahlige Problem und somit auch für gemischt-ganzzahlige Probleme im Allgemeinen klassisch nicht möglich. Wie im Anwendungsfall **Optimierung von Ladeplänen einer gegebenen Ladesäuleninfrastruktur** erläutert, kann das reine ganzzahlige Problem jedoch als QUBO-Problem formuliert werden und mit einem Quantencomputer mittels beispielsweise QAOA gelöst werden. In der Literatur finden sich derzeit zwei verschiedene Wege, wie nun ein gemischt-ganzzahliges Problem mit Hilfe eines klassischen Computers und eines Quantencomputers lösen lassen. Unter geeigneten Voraussetzungen kann das Problem mittels Alternating Direction Method of Multipliers (ADMM) gelöst werden, in dem die das ganzzahlige Problem als eine „Richtung" aufgefasst wird und bei fixierten reellen Werten auf dem Quantencomputer gelöst wird, z.B. mittels QAOA oder VQE. Anschließend wird analog der reellwertige Teil als eine andere Richtung aufgefasst und mit fixierten binären Variablen klassisch gelöst. Dieser Algorithmus ist in qiskit für eine bestimmte Problemklasse mit Konvexitätsbedingungen implementiert.

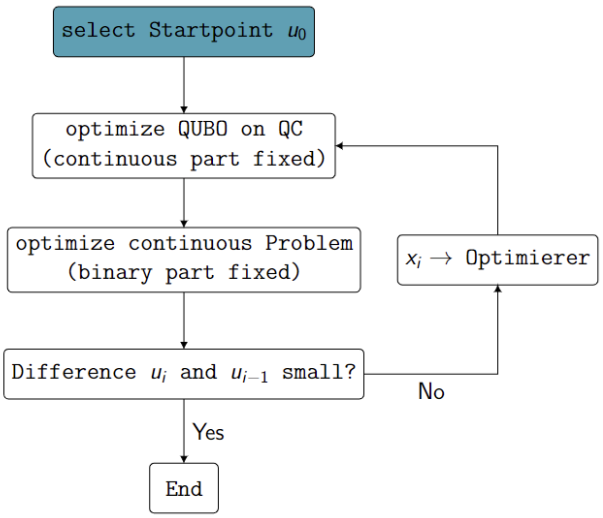


Abb. 62 Schematische Darstellung des ADMM-Algorithmus.

Unsere eigene Implementierung des 2-ADMM-Algorithmus erlaubt uns eine bessere Kontrolle über die Parameter als die in qiskit implementierte Version des 3-ADMM-Algorithmus. Die Implementierung des Algorithmus basiert auf dem VQE-Algorithmus und der estimator-qiskit-runtime-primitive, welche gemäß Dokumentation auch die Interaktion zwischen dem klassischen Teil auf klassischer Hardware und Quantencomputer verbessern sollte.

Wie ausführlich bei der Einführung zum Anwendungsfall Optimierung von Ladeplänen einer gegebenen Ladesäuleninfrastruktur (LamA) dargestellt, soll die Spitzenlast der Ladesäulen auf das Stromnetz unter Einhaltung von Zwangsbedingungen minimiert werden. Das Optimierungsproblem kann vollständig diskretisiert wie folgt formuliert werden:

$$\min \vec{p}_{sum}^{\,T} A \, \vec{p}_{sum}$$

$$s.t. C\vec{p}_{sum} = \vec{e}$$

Wobei $\vec{p}_{sum} = \{\vec{p}_0, \vec{p}_1, \dots, \vec{p}_{K-1}\}^T$ und $\vec{p}_i = \{p_i^0, p_i^1, \dots, p_i^t\}^T$, wobei der Index i für das i-te Auto steht und p_i^s die diskrete Ladeleistung zum Zeitpunkt s darstellt. A ist die konstante Kostenmatrix und die Matrix C formuliert die Zwangsbedingungen zusammen mit der von den Autos benötigen Energie $\vec{e}$. In der gemischt-binären Form (MBO-Form) von LamA ist der Ladezustand nicht mehr ganzzahlig, sondern wird als kontinuierlich angenommen. Daher wird $\vec{p}_i$ nun durch $\vec{\tilde{p}} = \{u_0^t * x_0^t, \dots, u_i^t * x_i^t\}$ ersetzt, wobei u_i^t kontinuierlich ist und repräsentiert, wie hoch die Ladeleistung des i-ten Autos zum Zeitpunkt t, x_i^t ist binär und repräsentiert, ob das i-te Auto zurzeit t geladen wird. Das LamA-Modell in MBO-Form ergibt sich dann als:

$$\min \vec{\tilde{p}}_{sum}^{\,T} A \, \vec{\tilde{p}}_{sum}$$

$$s.t. C\vec{\tilde{p}}_{sum} = \vec{e}$$

Dabei ist $\vec{\tilde{p}}_{sum}^{\,T} = Diag(\tilde{u}_{sum})\vec{p}_{sum}$.

Wie in Abb. 63 gezeigt, erreicht der ADMM-Löser erfolgreich optimale Ergebnisse für das Laden von drei verschiedenen Autos innerhalb des angegebenen Zeitrahmens. Darüber hinaus wird durch die Verwendung der MBO-Formulierung das QUBO-Problem

im Vergleich zum Originalproblem kleiner, was zu einem reduzierten Bedarf an Qubits und Speichergröße führt.

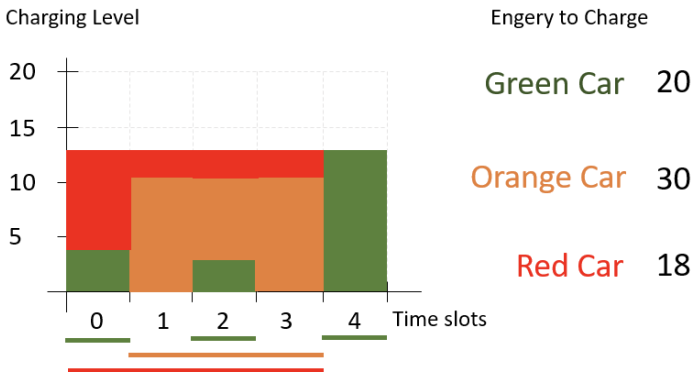


Abb. 63 Schematische Darstellung des Ladeplans als Lösung des Optimierungsproblems. Es werden drei Autos geladen, die jeweils 20, 30 und 18 Energieeinheiten benötigen. Die Autos waren dafür in den Zeitintervallen [0, 2, 4], [1, 2, 3], bzw. [0, 1, 2, 3] verfügbar.

Die Leistung des VQE-Lösers ist entscheidend für die Gesamtleistung des ADMM-Algorithmus, da dieser zur Lösung des binären Teils des Problems eingesetzt wird. Mit Hilfe des cuQuantum[1]-Benchmarking-SDK haben wir die Leistung eines Schrittes der VQE-Auswertung für verschiedene Kombinationen von Frontend und Simulator-Backend bewertet. Wie aus Abbildung 6 hervorgeht, zeigt das auf GPUs basierende Simulator-Backend im Vergleich zum auf CPU basierenden Backend eine signifikante bessere Leistung. Die Nutzung von GPUs ermöglicht eine deutliche Beschleunigung des gesamten VQE-Prozesses als Optimierer.

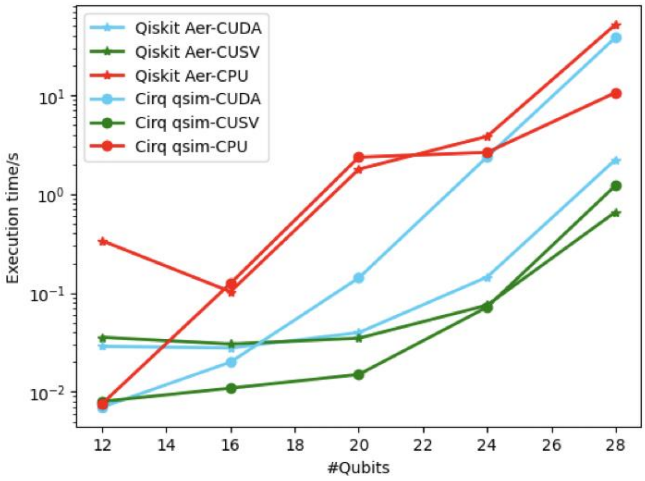


Abb. 64 Benchmark-Ergebnisse für des VQE-Algorithmus auf verschiedenen Frontends und Simulator-Backends. Eine EPYC 7742 CPU ist für das CPU-Backend verantwortlich und eine Nvidia A100-Grafikkarte wird für das GPU-Backend genutzt.

Referenzen

[1] Bayraktar, Harun, et al. "cuQuantum SDK: A high-performance library for accelerating quantum science." *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. IEEE, 2023.

2.3 Arbeitspaket 3 – Hardware-Software-Co-Design

Definiertes Ziel des dritten Arbeitspakets laut Projektbeschreibung ist es, geeignete Werkzeuge im Quantensoftwareentwicklungsprozess hardware-spezifisch zu erforschen und diese Nutzern abschließend in Form von einfach zu bedienenden Best Practices zur Verfügung zu stellen.
Der Fokus liegt hierbei auf Transpilations- und Fehlermitigationswerkzeugen, welche die spezifischen Eigenschaften der Quantenhardware berücksichtigen. Um möglichst genaue Ergebnisse zu erzielen, können Fehler durch möglichst kurze Schaltkreise verringert (Transpilation) bzw. durch klassisches Post-Processing korrigiert werden (Mitigation).

Unter Leitung des **Fraunhofer IAF** (und im Austausch mit den KQCBW-Projekten QuESt+ und QORA II) werden jene Arbeiten im SEQUOIA End-to-End Projekt anhand dreier Arbeitsschritte durchgeführt (vgl. Gantt-Chart in Abb. 2).

AP 3.1 Methoden und Werkzeuge Transpilation

- Optimierte Gattersynthese durch (näherungsweise) Lösung von SAT-Gleichungen
- Vergleich von Metriken zur Auswahl des besten Schaltkreises (Ergebnisse siehe AP1 Abb. 8)
- Verwendung von Cross-Resonance- anstatt CNOT-Gattern als Basisgatter des IBM Quantum System One Ehningen
- Erprobung und Bewertung der Classiq-Lösungen zur Schaltkreisgenerierung und Benchmarking mit eigenen Methoden (Ergebnisse siehe AP 5 in 2.5)
- Optimierung des Qubit Mappings zur Darstellung von Drei-Niveau- und offenen Quantensystemen

AP 3.2 Methoden und Werkzeuge Fehlermitigation

- Zero-Noise-Extrapolation in Verbindung mit Randomized-Gate-Insertions
- Probabilistic-Error-Cancellation (nach [Berg et. al. 2022])

AP 3.3 Best Practices fehlermitigierte Transpilationspipeline

- Zusammenstellen einer Dokumentation der Werkzeuge und Methoden
- Eruierung der Komposition jener Werkzeuge und Methoden

Jenem Arbeitsplan liegen dabei zwei Meilensteine zugrunde

- **M8:** Weiterentwicklung und Benchmarking der Transpilationswerkzeuge von AP3.1 auf dem IBM Quantum System One sind abgeschlossen; Zero-Noise-Extrapolation mit Randomized-Gate-Insertions wurde implementiert.
- **M15:** Probabilistic-Error-Cancellation wurde auf dem IBM Quantum System One demonstriert. Best Practices zur einfachen Bedienung der fehlermitigierten Transpilationspipeline sind erstellt.

Beide Meilensteine wurden erreicht (M15 in modifizierter Form, siehe unten), und die Ergebnisse in einer wissenschaftlichen Publikation veröffentlicht.

Transpilationsmethoden auf Basis von Redundanzansätzen

Für AP3 wurden verschiedene Architekturmuster aus der Domäne der fehlertoleranten Systeme betrachtet, die mittels verschiedenen Redundanzansätzen unterschiedliche Varianten von transpilierten Schaltkreisen erzeugen und anschließend aggregieren. Durch die Erzeugung unterschiedlicher Varianten eines Schaltkreises (auf Basis von unterschiedlichen Transpilationsseeds), werden heterogene aber funktional gleiche Varianten erzeugt, die jeweils zu unterschiedlichen Messungen führen können. Die Messungen werden anschließend (z. B. mittels eines *Linear Opinion Pools*) zu einer akkurateren Messung aggregiert und beugen, auf Basis des redundanten Transpilationsprozesses, NISQ-Fehlern vor.

Erste Resultate in [1] haben hierbei gezeigt, dass die Verwendung eines redundanten Transpilationsprozesses (auf Basis von zufällig generierten Quantenschaltkreisen) zu akkurateren Ergebnissen führen. Weitere Experimente wurden in Rahmen von AP5 durchgeführt. Hierbei war der Effekt von redundanten Transpilationsprozessen jedoch minimal bzw. genauso gut wie einzelne Ausführungen (d.h. ohne redundante Transpilationsseeds).

Referenzen

[1] M. Scheerer, J. Klamroth and O. Denninger, "Fault-tolerant Hybrid Quantum Software Systems," *2022 IEEE International Conference on Quantum Software (QSW)*, Barcelona, Spain, 2022, pp. 52-57, doi: 10.1109/QSW55613.2022.00023.

Benchmark alternativer QAOA-Ansätze

Neben dem bereits in Sequoia 1 implementierten QAOA-Ansatz (im Folgenden als QAOA+ bezeichnet) sind sowohl der Standard-QAOA, der nicht die Definition von festen Randbedingungen in der Problemstellung zulässt, dafür aber kürzere Schaltkreise definiert (im Vergleich zu QAOA+) als auch GQAOA, der in der Ausdruckskraft äquivalent zu QAOA+ ist, die Komplexität des Schaltkreises aber anders lagert, etablierte alternative Ansätze. Beide wurden zum Vergleich mit dem existierenden Ansatz implementiert und untereinander verglichen. Insbesondere die Schaltkreistiefe und die daraus resultierende Anfälligkeit für Rauschen waren dabei von Belang. Im folgenden Schaubild ist der Verlauf der Schaltkreistiefe gegenüber der Wahl des Parameters *p,* der die Anzahl der Wiederholungen der Problem-Hamiltonians angibt, aufgetragen.

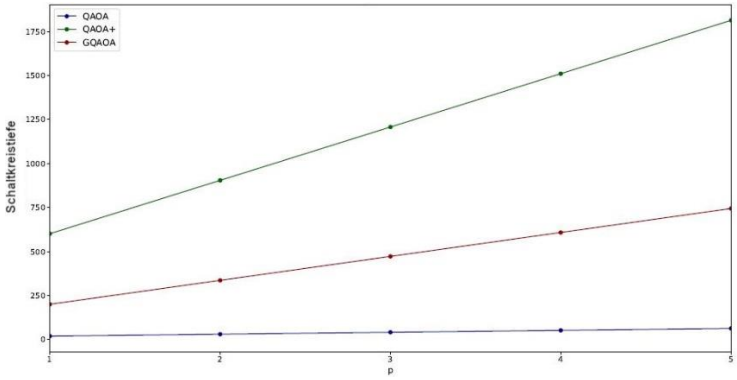


Abb. 65 Verlauf der Schaltkreistiefe der untersuchten QAOA-Ansätze für unterschiedliche Wiederholungen der Problem-Hamiltonians *p*.

Dabei ist deutlich zu sehen, dass QAOA mit Abstand die kleinsten Schaltkreise generiert. Der bisher verwendete Ansatz QAOA+ hingegen definiert Schaltkreise, deren Größe sehr stark wächst. GQAOA vereint dabei geringere Schaltkreistiefe mit der Möglichkeit zur Definition von Randbedingungen, wodurch dieser Ansatz weiterführend als der geeignetste angesehen wird. Die kleinen Schaltkreise von QAOA lassen ungültige TSP-Touren zu, die in der Praxis häufig zu schlechten oder ungültigen Ergebnissen führen.

Die geringere Schaltkreistiefe wirkt sich messbar auf die Anfälligkeit für Rauschen aus. Die Auswirkungen von Rauschen wurde anhand der Fidelity gemessen, also ein Maß dafür, wie sehr zwei Quantenzustände einander ähneln. Die Zustände nach jeweils einer Simulation ohne und einer Simulation mit Rauschen wurden miteinander verglichen, wobei eine hohe Fidelity bedeutet, dass die beiden Zustände sich sehr ähneln, das Rauschen also nur wenig Einfluss hatte.

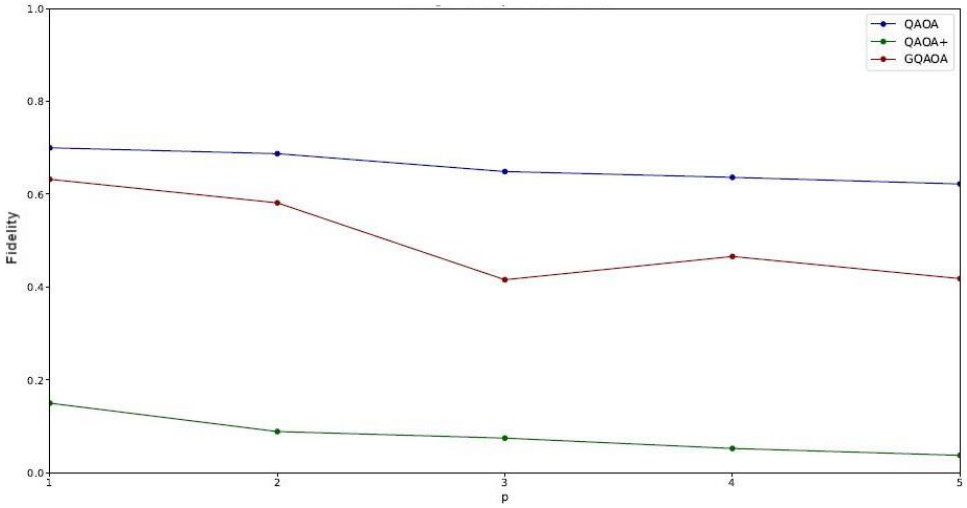


Abb. 66 Verlauf der Fidelity, also der Nähe des gemessenen Zustands zu dem erwarteten, der untersuchten QAOA-Ansätze bei simuliertem Rauschen für unterschiedliche Wiederholungen der Problem-Hamiltonians *p*.

Es wird ersichtlich, dass die Fidelity mit zunehmender Schaltkreistiefe abnimmt. Während das Rauschen auf QAOA+ einen sehr starken Einfluss hat, ist das Rauschen bei GQAOA eher vergleichbar mit QAOA, was die Ausführbarkeit auf NISQ-Rechnern deutlich besser macht.

Methoden und Werkzeuge Fehlermitigation

Im Rahmen von Arbeitspaket 3.2 wurde die bereits in Sequoia 1 integrierte Fehlerminderungstechnik Standard-Zero-Noise-Extrapolation (sZNE) durch die Anwendung von Fehlerrandomisierung (Twirling) nach Wallmann [1] erweitert. Diese sollte eine Steigerung der Qualität der extrapolierten Ergebnisse ermöglichen.

Hierbei werden zufällige Einzelgatter vor und nach jedem CNOT-Gatter sowie auf benachbarten Qubits hinzugefügt. Dies führt zur Vereinfachung der Fehler zu stochastischen Pauli-Fehlern und ermöglicht die Unterdrückung von korrelierten Rauscheffekten zwischen benachbarten Qubits. Abb. 67 Abb. 67zeigt einen Schaltkreis mit drei Qubits, einem CNOT-Gatter und mehreren Einzelgattern. Während des Twirling-

Prozesses werden zufällige Einzelgatter in den ursprünglichen Schaltkreis eingefügt (siehe Mitte). Durch anschließende Vereinfachung und Zusammenfassung der Einzelgatter bleibt die Gesamtzahl der Gatter unverändert (siehe rechts).

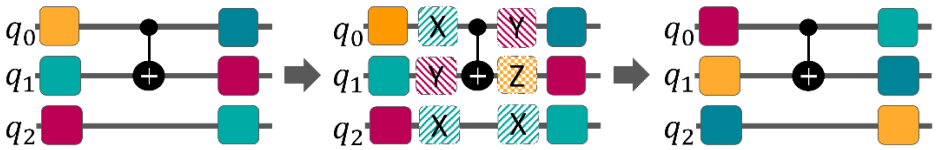


Abb. 67 Fehlerrandomisierung (Twirling). Das Einfügen zufälliger Einzelgatter vor und nach jedem CNOT-Gatter sowie bei angrenzenden Qubits (wegen möglicher Crosstalk-Effekte) wandelt die Fehler in stochastische Pauli-Fehler um. Durch anschließendes Vereinfachen und Zusammenfassen der Einzel-Gatter ist die Gesamtgatteranzahl genauso groß wie zu Beginn.

Die Auswirkung des Twirlings auf die ZNE-Fehlermitigation wird anhand des Grover-Algorithmus mit drei Qubits und einer Implementierung nach [2] mit insgesamt 10 CNOT-Gattern demonstriert. Dieser Algorithmus löst unstrukturierte Suchprobleme. In der hier verwendeten Implementierung gibt es zwei mögliche korrekte Lösungen: $|101\rangle$ und $|110\rangle$. Als zu mitigierende Größe betrachten wir die Wahrscheinlichkeit eines korrekten Ergebnisses, also die Summe der Wahrscheinlichkeiten für die Messung eines der korrekten Zustände.

Zusätzlich wurde eine neue Methode von uns entwickelt und implementiert [3], um die tatsächlichen Fehlerraten von Schaltkreisen zu bestimmen und damit die Extrapolationsergebnisse zu verbessern. Diese Methode bezeichnen wir als „Inverted-Circuit-ZNE" (IC-ZNE). Bei dieser Methode wird der invertierte Schaltkreis an den zu messenden Schaltkreis angehängt, um die Fehlerrate zu messen (siehe Abb. 68).

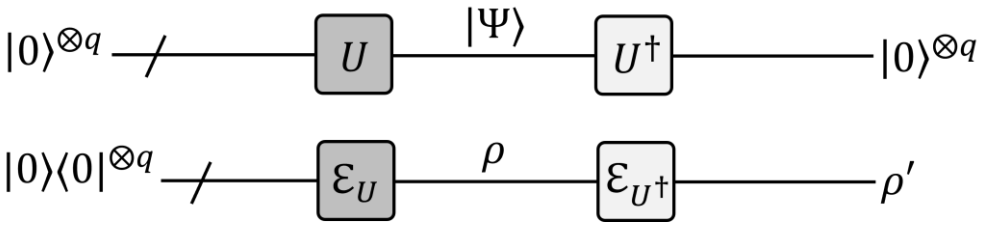


Abb. 68 Schematische Darstellung der IC-ZNE Methode. Ohne Rauschen stellt das Hinzufügen des inversen Schaltkreises $U^\dagger$ nach U den Ausgangszustand $|0\rangle$ wieder her. Die entsprechenden verrauschten Kanäle $\mathcal{E}_U$ und $\mathcal{E}_{U^\dagger}$ erzeugen jedoch eine Dichtematrix ρ', die Informationen über die Gesamtfehlerstärke des Schaltkreises enthält.

Die anhand der Fidelity F des verrauschten Zustands ρ bezüglich des idealen Zustands $|\Psi\rangle$ definierte Fehlerrate $\epsilon = 1 - F$ wird hierbei aus der Wahrscheinlichkeit ermittelt, zum Schluss alle Qubits wieder im Zustand $|0\rangle$ zu finden [3]. Auf diese Weise können wir die Fehlerstärke eines Schaltkreises direkt messen und dadurch überprüfen, ob sich bei den zum Zweck der Zero-Noise-Extrapolation skalierten Schaltkreisen der Fehler tatsächlich um den beabsichtigten Faktor verstärkt (z.B. um einen Faktor 3, falls jedes CNOT-Gatter durch 3 CNOT-Gatter ersetzt wird). In der Tat zeigen unsere Ergebnisse im Fall der oben erwähnten Grover-Schaltung, dass das Ergebnis der Zero-Noise-

Extrapolation genauer wird, wenn man die erwarteten Skalierungsfaktoren durch die direkt gemessene Fehlerstärke ersetzt, siehe Abb. 69

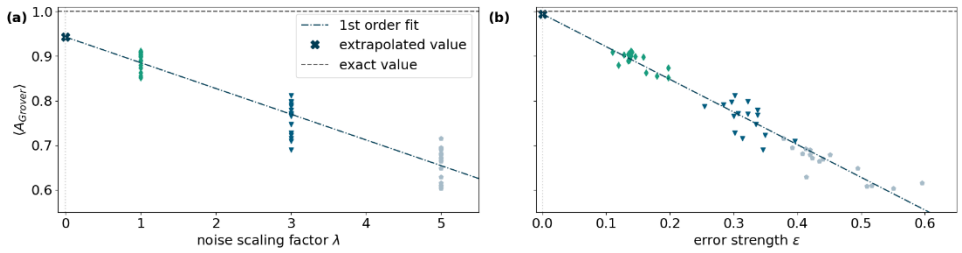


Abb. 69 Standard-ZNE (a) und Inverted-Circuit-ZNE (b) für die Grover-Schaltung auf dem IBM-System ibmq ehningen. Für jeden Skalierungsfaktor $\lambda = 1$ (grüne Rauten), $\lambda = 3$ (blaue Dreiecke) und $\lambda = 5$ (graue Fünfecke) führen wir 16 verschiedene zufällig getwirlte Schaltkreise mit jeweils 625 Shots aus (d. h. 10000 Shots pro Skalierungsfaktor). (a) In der Standard-ZNE werden die gemessenen Erwartungswerte $\langle A_{Grover} \rangle$ gegen den Skalierungsfaktor λ aufgetragen. Eine polynomialer Fit erster Ordnung ergibt den extrapolierten Wert $\langle A_{Grover} \rangle = 0{,}94$, der deutlich kleiner ist als der Idealwert $\langle A_{Grover} \rangle_{ideal} = 1$ (horizontale gepunktete Linie). In (b) Inverted-Circuit ZNE wird die Fehlerstärke ϵ jedes Schaltkreises mit der Methode der invertierten Schaltkreise (mit 625 zusätzlichen Shots pro Schaltkreis) gemessen. Die gleichen Erwartungswerte $\langle A_{Grover} \rangle$ wie in (a) werden nun als Funktion von ϵ anstelle von λ aufgetragen, was ein genaueres Extrapolationsergebnis von $\langle A_{Grover} \rangle = 0{,}99$ ergibt.

Dieses Ergebnis bestätigt sich auch bei 50-facher Wiederholung des für Abb. 69 durchgeführten Experiments. In Abb. 70 sind für diese 50 Experimente jeweils die Rohdaten (Raw, d.h. die ohne Fehlermitigation gemessenen Erwartungswerte) ohne (a) und mit (b) Twirling-Gattern sowie die durch lineare Extrapolation unter Verwendung von Standard-ZNE bzw. IC-ZNE ermittelten mitigierten Erwartungswerte dargestellt. Der Boxplot zeigt, dass der Abstand zwischen dem ersten und dem dritten Quartil von $\langle A_{Grover} \rangle$ (d. h. die Größe der Box) für IC-ZNE im Fall ohne Twirling am größten ist, aber es in beiden Fällen (mit und ohne Twirling) weniger oder keine Ausreißer (offene Kreise) im Vergleich zu den Rohdaten oder sZNE gibt. Darüber hinaus liegen die mit IC-ZNE ermittelten Erwartungswert insgesamt näher am exakten Wert, was zu einem kleineren quadratischen Fehler (Root-Mean-Square Error, RMSE) führt. Das hervorragende Abschneiden der IC-ZNE, die eine im Wesentlichen von systematischen Fehlern freie Extrapolation liefert, ist dabei auf die genauere Bestimmung der Fehlerstärke zurückzuführen.

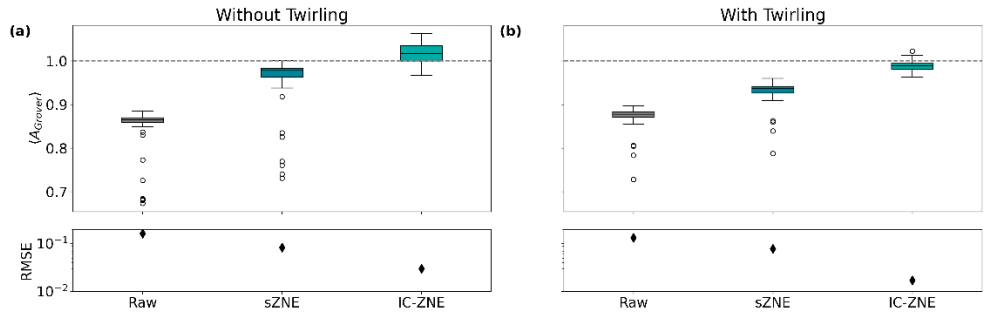


Abb. 70 Vergleich von Inverted-Circuit-ZNE (IC-ZNE), Standard-ZNE (sZNE) und Rohdaten (Raw) für 50 Durchläufe des Grover-Schaltkreises auf dem IBM-Quantensystem ibmq_ehningen ohne (a) und mit (b) randomisierter Kompilierung durch Pauli-Twirling. Die erhaltenen Werte $\langle A_{Grover} \rangle$ sind in einem Boxplot dargestellt. Die entsprechenden mittleren quadratischen Fehler (RMSE), die die Abweichungen vom exakten Wert $\langle A_{Grover,\ ideal} \rangle = 1$ (horizontale gestrichelte Linie) angeben, sind in den Abbildungen unten dargestellt. Das genaueste Ergebnis - mit dem kleinsten RMSE und einer geringeren Anzahl statistischer Ausreißer (offene Kreise) - wird mit IC-ZNE mit Twirling erzielt.

Aus Abb. 70 geht außerdem hervor, dass unsere Methode IC-ZNE besonders gut in Verbindung mit Random Twirling abschneidet, da in diesem Fall die durch den RMSE quantifizierte Abweichung vom idealen Wert am kleinsten ist.
Meilenstein M8 wurde somit erfolgreich erreicht: Eine von uns weiterentwickelte Version der Zero-Noise-Extrapolation wurde in Verbindung mit Randomized-Gate-Insertions implementiert. Es konnte gezeigt werden, dass diese Methode verbesserte Ergebnisse auf dem Quantencomputer ermöglicht.

Parallel dazu wurde die Probabilistic Error Reduction (PER) gemäß [4, 5] implementiert. Ähnlich wie Probabilistic Error Cancellation (PEC) [6] stellt PER ideale Operationen als lineare Kombinationen von verrauschten Operationen dar, die auf der Hardware implementierbar sind. Anstatt jedoch von einem festen Pegel des Hardware-Rauschens auszugehen, wird die Menge der implementierbaren Operationen durch Skalierung des Rauschens erweitert. Durch den Aufbau der Methode umfasst diese sowohl PEC als auch ZNE als Sonderfälle. Die PER kann zur Schätzung von Erwartungswerten bei virtuellen Fehlerskalierungsfaktoren kleiner als 1 verwendet werden, was zu einer teilweisen Mitigation bei geringerem Kostenaufwand führt.

Die PER wurde mithilfe von Simulationen und einem Fehlermodell eines IBM-Systems getestet. Die Ergebnisse der Simulation einer Trotter-Dynamik-Simulation des Ising-Modells sind in Abb. 71 dargestellt. Die PER-Methode (blaue Linie) korrigiert die Fehler teilweise, erreicht jedoch bei weitem noch nicht die exakten Werte (orange Linie) der Simulation. Für die Untersuchungen auf der echten Hardware (ibmq_ehningen) habe wir uns daher dafür entschieden, uns auf die oben dargestellte Weiterentwicklung der Zero-Noise-Extrapolation mit direkter Messung der Fehlerstärke (IC-ZNE) zu konzentrieren, welche insgesamt die besten Ergebnisse liefert. Meilenstein M15 wurde somit in modifizierter Form erreicht.

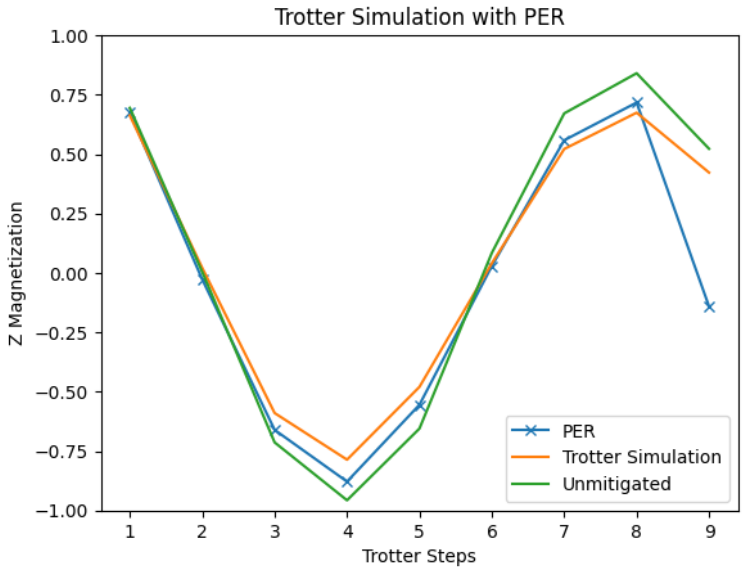


Abb. 71 Trotter-Dynamik-Simulation des Ising-Modells mit transversalem Feld. Im Graphen zu sehen ist die Magnetisierung in z-Richtung auf der y-Achse und die Anzahl der Trotter-Schritte auf der x-Achse. Eine Simulation ohne Fehler ist in Orange dargestellt, das unmitigierte Ergebnis der Simulation mit einem Fehlermodell ist in grün dargestellt. Die blaue Linie zeigt die mit PER durchgeführte Simulation. Die PER verbessert die Simulation, erreicht allerdings die exakten Werte noch nicht.

Die IC-ZNE ist als Sequoia-Demonstrator in Form eines Jupyter-Notebooks dokumentiert und zur Benutzung verfügbar. Die zu Grunde liegende Theorie, sowie die Ergebnisse von

Simulationen und auf ibmq_ehningen durchgeführten Versuchen für verschiedene Testschaltkreise (Grover und HHL) wurden außerdem als Preprint veröffentlicht [3].

Referenzen

[1] J. J. Wallman and J. Emerson, "Noise tailoring for scalable quantum computation via randomized compiling", Phys. Rev. A 94, 052325 (2016)

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

[3] K. F. Koenig, F. Reinecke, W. Hahn and T. Wellens, „Inverted-circuit zero-noise extrapolation for quantum gate error mitigation", arXiv 2403.01608 (2024)

[4] B. McDonough, A. Mari, N. Shammah, N. T. Stemen, M. Wahl, W. J. Zeng, P. P. Orth, „Automated quantum error mitigation based on probabilistic error reduction", IEEE, 2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS) (2022), doi: 10.1109/qcs56647.2022.00015

[5] A. Mari, N. Shammah, W. J. Zeng, „Extending quantum probabilistic error cancellation by noise scaling", Phys. Rev. A 104, 052607 (2021)

[6] E. van den Berg, Z. K. Minev, A. Kandala, and K. Temme, "Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors", Nature Physics 19, 1116–1121 (2023)

2.4 Arbeitspaket 4 – Benchmarking

Definiertes Ziel des vierten Arbeitspakets ist laut Projektbeschreibung das Erarbeiten einer Methodik für die Durchführung von Benchmarks zur Performance-Analyse verschiedener SDKs auf klassischen Infrastrukturen. Es wird erforscht welche Diskrepanz Resultate aufweisen, die entweder durch QC-Emulatoren oder reale QC-Systeme wie dem IBM Q System One berechnet wurden und welche Performance erzielt werden kann. Weiterhin werden Benchmarks für Anwendungsfälle von Industriepartnern sowie ein systematischer Vergleich klassischer Optimierer durchgeführt. Ergebnis des Arbeitspakets ist eine Benchmark-Suite.

Unter Leitung des **HLRS der Universität Stuttgart** wird die Benchmark-Forschung anhand dreier Arbeitsschritte durchgeführt (vgl. Gantt-Chart in Abb. 2).

AP 4.1 Systematischer Vergleich klassischer Optimierer

- Vergleich aller Klassen (stochastisch, genetisch, simplizial…) bezüglich einer einheitlichen Metrik und mit Blick auf die verschiedenen Einsatzgebiete: u.a. perfekte Simulation, Simulation mit Shot-Noise, Quantencomputer. Bewertet werden sollen insbesondere etablierte Optimierer wie COBYLA, SLSQP, (QN-)SPSA und CMA-ES.
- Dokumentation von Best Practices und Empfehlungen: Welcher Optimierer ist in welcher Situation mit welchen Parametereinstellungen
- Entwicklung eines leistungsfähigen klassischen globalen Optimierers zum Trainieren von variationellen Quantenschaltkreisen, um die so genannte "Barrenplateaus" und lokale Minima zu vermeiden

AP 4.2 SDK-Performance auf klassischer Hardware (CPU-GPU-HPC)

- Installation von QC-Emulatoren (SDKs) wie Qiskit und Cirq auf der HPC-Infrastruktur des HLRS mit anschließender Konfiguration der QC-Emulatoren (u.a. Beschleuniger wie GPUs, falls unterstützt)
- Auswahl einiger in und mittels der SDKs implementierten Algorithmen. Nach Möglichkeit mit Bezug zu Anwendungsfällen. Implementierung dieser Algorithmen mit Classiq.
- Vergleich bezüglich Implementationsaufwand, Skalierungseigenschaften (u.a. Multi-Node, mit/ohne GPU-Beschleuniger) und Genauigkeit der Ergebnisse.
- Vergleich mit Ergebnissen von realem Backend und Dokumentation der Ergebnisse

AP 4.3 Benchmark End-to-End Demonstratoren

- Festlegung der Benchmark-Szenarien mit Industriepartnern und Bewertungsmetriken
- Analyse von QUBO-Kapazitäten anhand des Anwendungsfalls „Fertigungsstraßen". Bewertung von verschiedenen Testumgebungen: von klassischer Optimierung mit Standardhardware, über HPC bis zu Fraunhofer IAIS Evo Annealer und QAOA-Bewertung der Leistungsfähigkeit der MIPs Algorithmen des Anwendungsfalls „Routenplanung" auf verschiedener Hardware (Multi-Processing/-Threading, Multi-Node, Beschleunigerunterstützung)

Jenem Arbeitsplan liegen dabei zwei Meilensteine zugrunde:

- **M8:** Installation und Konfiguration von QC-Methoden und QC-Emulatoren ist abgeschlossen; erste Bewertung klassischer Optimierer hat stattgefunden.
- **M15:** Finaler Vergleich klassischer Optimierer inklusive Best Practices liegt vor; Benchmark-Suite ist implementiert, dokumentiert und Benchmark-Ergebnisse aus AP4.2 und AP4.3 liegen vor.

Zum Abschlussbericht sind dabei M8 und M15 planmäßig erreicht worden (vgl. Abb. 2). Im Folgenden wird der Stand der zugrundeliegenden Forschung dargestellt.

Entwicklung eines klassischen globalen Optimierers zum Vermeiden von Barrenplateaus (Umgang mit schwierigen Optimierungslandschaften)

Barren Plateaus sind Phänomene in der Optimierung von variationellen Quantenschaltkreisen, bei denen der Gradient der Zielfunktion in Richtung des globalen Minimums sehr klein wird. Das macht es für Optimierungsalgorithmen schwierig effizient zu konvergieren. Ähnlich wie bei "vanishing gradients" in klassischen neuronalen Netzwerken können Barren Plateaus das Training verlangsamen und die Konvergenz zu einer optimalen Lösung verhindern. Im Rahmen dieses Unterarbeitspakets wurde ein klassischer Optimierer entwickelt, der mit diesen unvorteilhaften Optimierungslandschaften gut umgehen können soll. Der Algorithmus basiert auf einer Kombination von Simultaneous Perturbation Stochastic Approximation (SPSA) und Deterministic Annealing (siehe AP 1, Verifikation von Neuronalen Netzen). Der Psuedo-code des entwickelten Algorithmus ist in Abb. 72 zu sehen.

Algorithm Global minimization of $f(\theta)$ over $[a,b]^n$

INPUTS: maximum number of seeds $K_{\max}$, initial and final temperatures T_0, $T_{\min}$, cooling factor α
initialize $T \leftarrow T_0$, $K \leftarrow 1$ and the seed $\mathbb{P}^{(1)} \leftarrow \left(\frac{1}{2}, \ldots, \frac{1}{2}\right)$
while $T > T_{\min}$ **do**
    for each seed $k = 1$ to K **do**
        minimize the free energy $F_T(\mathbb{P}^{(k)})$ at temperature T with respect to $\mathbb{P}^{(k)}$ using SPSA and step size $d(T)$
    end for
    slowly decrease the temperature: $T \leftarrow \alpha * T$
    for each seed $k = 1$ to K **do**
        compute the Hessian $H^{(k)}$ of the free energy of $F_T(\mathbb{P})$ at $\mathbb{P}^{(k)}$ using numdifftools.Hessian
        compute the minimum eigenvalue $\lambda_{\min}^{(k)}$ of $H^{(k)}$ and the corresponding eigenvector $e^{(k)}$ using numpy.linalg.eig
        if $\lambda_{\min}^{(k)} < 0$ (phase transition) **then**
            replace $\mathbb{P}^{(k)}$ by $\mathbb{P}^{(k)} - \epsilon e^{(k)}$, add a new seed $\mathbb{P}^{(k)} + \epsilon e^{(k)}$ and increment K by 1
        end if
    end for
    keep the best $K_{\max}$ seeds (with lowest value of the free energy) in case K exceeds $K_{\max}$
end while
return the parameters $\theta(\mathbb{P}^k)$ for the seed $\mathbb{P}^{(k)}$ with lowest value of $f(\theta(\mathbb{P}^{(k)}))$

Abb. 72 Der im Projekt entwickelte klassische Algorithmus basierend auf SPSA und DA zur Optimierung variationeller Quantenschaltkreise

Im Laufe des Projekts hat sich herausgestellt, dass der Ansatz nicht vielversprechend ist und nicht die gewünschten Resultate bringt. Eine Weiterverfolgung wurde als nicht zielführend eingestuft. Stattdessen wurden die Bestrebungen in der automatischen Feature-Map Generierung in AP 2, die zum damaligen Zeitpunkt äußerst vielversprechend waren, weiter intensiviert. Dies wurde bereits im Zwischenbericht vermerkt. Die zusätzlichen Personalkapazitäten wurden genutzt, um für die für die Feature-Map Generierung entwickelte Methode ein umfangreiches Benchmarking zum Nachweis der Effektivität durchzuführen.

Analyse von QUBO-Kapazitäten und Bewertung von Testumgebungen

Arbeitspaket 4.3 widmet sich der Analyse von QUBO-Kapazitäten anhand des Anwendungsfalls „Fertigungsstraßen“. Der Anwendungsfall ist in den Ergebnissen von AP 1 (Kostenoptimierung und Auslegung von Fertigungsstraßen, Kapitel 2.1.3)

dokumentiert. Die Bewertung von verschiedenen Testumgebungen steht dabei in besonderem Fokus: von klassischer Optimierung mit Standardhardware, über verschiedene Quantenalgorithmen.

Anwendungsfall	*Beispiel 1*	*Beispiel 2*	*Beispiel 3*
# Aufgaben	4	4	20
# Arbeitsplätze	2	2	3
# Maschinen	2	3	5
Taktzeit	40	40	1000
Kanten im Ablaufgraf	3	3	16
Nichtnegative Bearbeitungszeiten	6	9	57
Dimension lineares Problem	16	24	186
Dimension QUBO	64	87	494

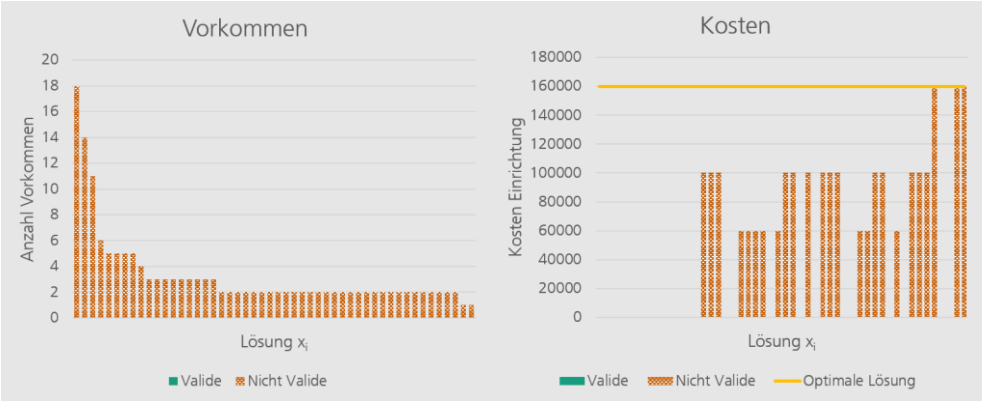
Tabelle 4 Zum Benchmarking verwendete Instanzen des Assembly Line Balancing Problems. Die Tabelle zeigt die jeweiligen Problemparameter (Anzahl der Aufgaben, Arbeitsplätze und Maschinen, Taktzeit, Anzahl der Kanten im Ablaufgraf, Nichtnegative Bearbeitungszeiten) sowie die Dimension der resultierenden Problemformulierungen als Anzahl der binären Entscheidungsvariablen.

Zum Benchmarking des Lösungsansatzes wurden drei Instanzen des Assembly Line Balancing Problems in verschiedenen Größen gewählt (siehe Tabelle 4). Zunächst wurden für jede Instanz mit dem in AP 1 beschriebenen Verfahren die optimalen Lagrange Parameter ermittelt. Der Suchraum wurde hierfür für Beispiel 1 und 2, für jeden Lagrange Parameter, auf 20 logarithmisch verteilte Werte im Intervall $[1, 10^4]$ eingeschränkt. Die Ergebnisse für alle Beispiele werden in Tabelle 5 dargestellt. Für Beispiel 3 konnten auch mit wiederholter Anpassung des Suchraums keine Parameter gefunden werden, um Lösungen zu produzieren, welche die Nebenbedingungen erfüllen. Hier wird vermutet, dass die Optimierungslandschaft des QUBOs zu kompliziert für das verwendete Lösungsverfahren ist.

Problem	λ_0	λ_1	λ_2	λ_3
Beispiel 1	335,98	18,33	29,76	335,98
Beispiel 2	206,91	18,33	11,29	78,48
Beispiel 3	-	-	-	-

Tabelle 5 Optimale Lagrange Parameter für die jeweiligen Instanzen. λ_0, λ_1, λ_2 und λ_3 werden jeweils den Nebenbedingungen (2), (3), (4) und (5) in **Tabelle 4** zugeordnet. Für Beispiel 3 wurden im gewählten Suchraum keine Lösungen gefunden, welche die Nebenbedingungen erfüllen.

Zusätzlich zu den in AP 1.2 aufgeführten Lösungen für Beispiel 1 wurden auch für Beispiel 2 1.000 Samples mit DWAVE + QBSolv produziert. Für Beispiel 2 sind unter den ersten 50 Lösungen keine, welche die Nebenbedingungen erfüllen, zu finden. Die erste solche Lösung wird an 181. Stelle gefunden. Noch gravierender ist die Situation bei Beispiel 3. Hier wurden trotz intensiver numerischer Suche keine passenden Lagrange Parameter gefunden, die eine nennenswerte Menge valider Lösungen produziert haben.



a) b)

Abb. 73 Top 50 Ergebnisse von 1000 Samples von QBSolv mit DWAVE Quantum Annealer, angewendet auf Beispielinstanz 2. Beide Plots zeigen, ob eine Lösung die Nebenbedingungen erfüllt oder nicht erfüllt. a) zeigt die Häufigkeit der jeweiligen Lösungen, b) zeigt die Werte der ursprünglichen Kostenfunktion aus [1] der jeweiligen Lösungen. Außerdem ist der Wert der bestmöglichen Lösung markiert.

Die Ergebnisse zeigen zum einen, dass mit wachsender Problemdimension die Wahrscheinlichkeit, die optimale Lösung zu finden geringer wird. Hier müssen mit wachsender Dimension immer mehr Samples erzeugt werden, um qualitativ hochwertige Lösungen zu erhalten. Das zugrundeliegende Problem liegt in der Anzahl der zu lösenden Sub-Systeme sowie der Komplexität der heuristischen Suche. Zum anderen wird deutlich, dass die Umformulierung von Optimierungsproblemen mit Nebenbedingungen in nebenbedingungsfreie Formulierungen mit steigender Problemgröße immer schwieriger zu handhabende Optimierungslandschaften erzeugt, deren Hyperparameter (Lagrangeparameter) nur schwer zu optimieren sind.

Da für Beispiel 3 keine Lösungen gefunden werden konnte, wurde im Folgenden nur mit Beispielen 1 und 2 fortgefahren. Für diese Probleminstanzen wurden, zusätzlich zu den in AP 1.2 erzeugten Ergebnissen mittels DWAVE + QBsolv, 1000 Samples mit dem simulated Annealing Algorithmus Neal erzeugt. Außerdem wurden alle drei Instanzen in der ursprünglichen Formulierung mit dem linearen Solver Gurobi [1] gelöst. Die Ergebnisse wurden hinsichtlich ihrer Lösungsgüte und Laufzeit miteinander verglichen. Bei der Bestimmung der Laufzeit wurde die Suche der Lagrange-Parameter ausgeschlossen. Tabelle 6 zeigt die Metriken in welchen die Lösungen miteinander verglichen wurden.

Metrik	*Formel*	*Ziel*
Kosten	$\sum_{j=1}^{r}\sum_{k=1}^{m} EC_j \cdot y_{jk}$	Minimal
Anzahl gekaufter Maschinen	$\sum_{j=1}^{r}\sum_{k=1}^{m} y_{jk}$	Minimal
Auslastung/Effizienz	$\sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{k=1}^{m} \frac{x_{ijk} \cdot t_{ij}}{m \cdot ct}$	Maximal
Laufzeit inkl. Setup		Minimal

Tabelle 6 Metriken zur Bestimmung der Lösungsgüte sowie Laufzeit.

Zum Vergleich mit der Lösung mit Gurobi wurde pro Metrik und Instanz die jeweils beste Lösung der produzierten 1000 Samples der Solver Neal und DWAVE + QBSolv gewählt.

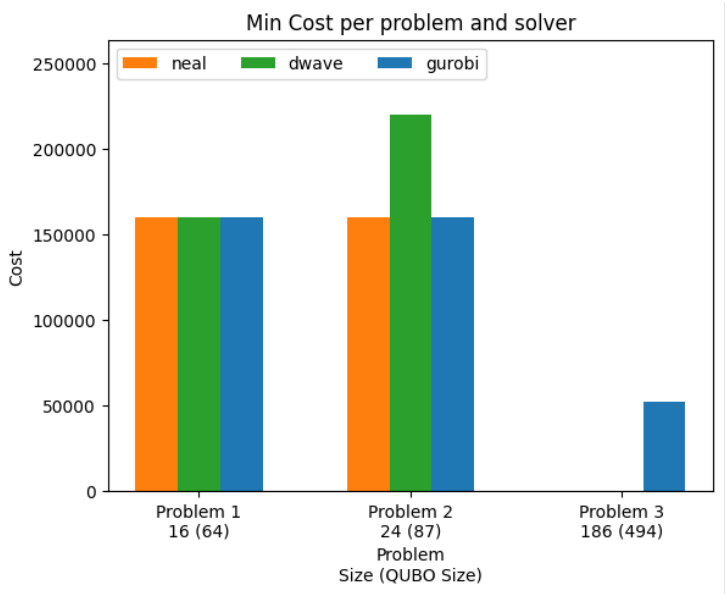


Abb. 74 Kosten zum Einrichten der Fertigungsstraße für die jeweiligen Probleminstanzen durch Lösung mittels der angegebenen Lösungsverfahren. Bei Neal sowie DWAVE wurde die jeweils beste Lösung der 1000 produzierten Samples gewählt. Für Beispiel 3 konnten mit Neal und DWAVE keine Lösung produziert werden.

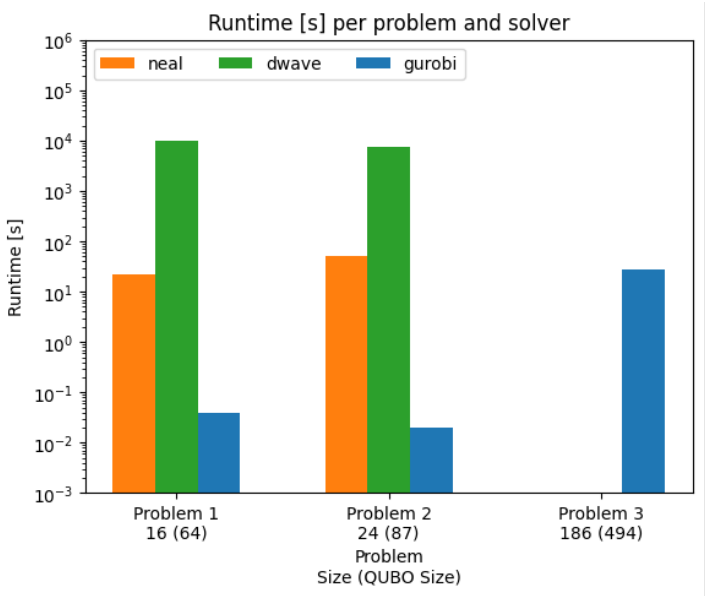


Abb. 75 Laufzeit in Sekunden der angegebenen Lösungsverfahren für die jeweiligen Probleminstanzen. Für Beispiel 3 konnten mit Neal und DWAVE keine Lösung produziert werden.

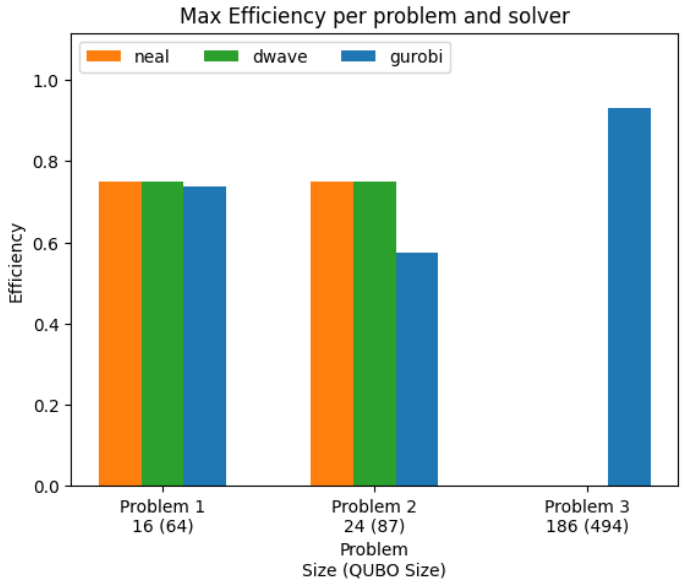


Abb. 76 Auslastung der eingerichteten Fertigungsstraße für die jeweiligen Probleminstanzen durch Lösung mittels der angegebenen Lösungsverfahren. Bei Neal sowie DWAVE wurde die jeweils beste Lösung der 1000 produzierten Samples gewählt. Für Beispiel 3 konnten mit Neal und DWAVE keine Lösung produziert werden.

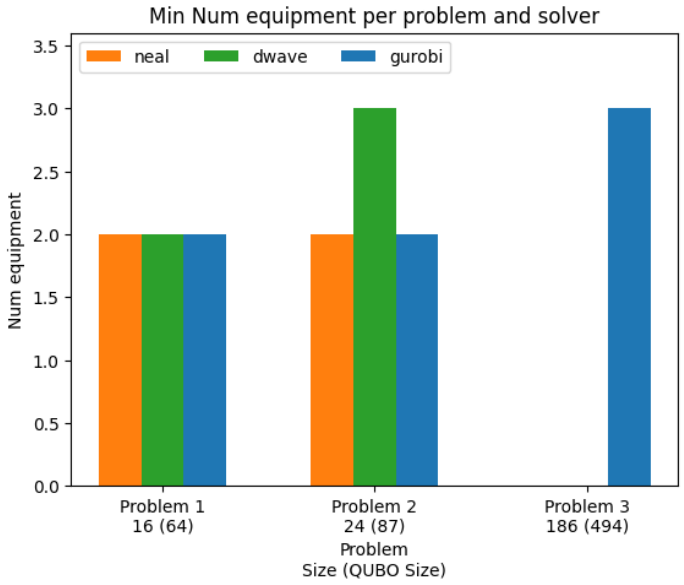


Abb. 77 Anzahl der verwendeten Maschinen in der eingerichteten Fertigungsstraße für die jeweiligen Probleminstanzen durch Lösung mittels der angegebenen Lösungsverfahren. Bei Neal sowie DWAVE wurde die jeweils beste Lösung der 1000 produzierten Samples gewählt. Für Beispiel 3 konnten mit Neal und DWAVE keine Lösung produziert werden.

Abb. 74 bis Abb. 77 zeigen die jeweiligen Ergebnisse der Metriken, Instanzen und Lösungsverfahren. Hierbei werden nur Lösungen einbezogen, welche alle Nebenbedingungen erfüllen. Für Beispiel 1 sind alle verwendeten Verfahren in der Lage, die optimale Lösung zu produzieren. Für Beispiel 2 ist es mit DWAVE + QBSolv nicht möglich die optimale Lösung zu produzieren. Die beiden anderen Lösungsverfahren erfüllen dies. Bei allen drei Instanzen ist die Lösung mit Gurobi am schnellsten, gefolgt von Neal und zuletzt DWAVE + QBSolv. Neal und DWAVE + QBSolv produzieren auch Lösungen mit einer höheren Auslastung der Fertigungsstraße, wobei zu bemerken ist,

dass diese möglicherweise auch suboptimale Lösungen bezüglich der Kosten sein können. Bei Beispiel 2 findet DWAVE + QBSolv keine Lösung mit minimal möglicher Anzahl gekaufter Fertigungsmaschinen.

Die Ergebnisse zeigen auf, dass für kombinatorische Optimierungsprobleme ein gewisses Potenzial für die Lösung mittels Quantencomputing Methoden besteht. Das Projekt zeigt jedoch auch die Grenzen der aktuell verfügbaren Lösungsverfahren auf. Die hybride Lösung aus Quantum Annealing und QBSolv stößt schnell an seine Grenzen und kann bereits für kleine Instanzen nur schwierig gute Ergebnisse erzielen. Für eine Lösung mittels Quantum Annealing ohne QBSolv muss die Qualität der Quantencomputer noch steigen. Hier spielt vor allem die Konnektivität der Qubits eine Rolle. Auch Simulated Annealing hat bei den größten betrachteten Probleminstanzen keine Ergebnisse geliefert, was die Schwierigkeiten bei der Umformulierung des beschränkten Optimierungsproblems in ein unbeschränktes Problem illustriert. Für beide Verfahren kann durch Optimierung der Solver-Parameter noch weiteres Potenzial ausgeschöpft werden, was jedoch mit einem großen Aufwand verbunden ist. Ein Faktor, der für alle Algorithmen, die auf die Umformulierung eines Problems mit Nebenbedingen in ein Problem ohne Nebenbedingungen berücksichtigt werden muss, ist, dass das Überführen der Nebenbedingungen in die Kostenfunktion neue Hyperparameter (Lagrangeparameter) erzeugt, deren Wahl nicht trivial ist. Das Finden geeigneter Lagrangeparameter ist mit enormen (simulativen) Kosten verbunden. Für die Verwendung von QAOA muss auch die Qualität und Größe der Quantencomputer stark verbessert werden, da hierfür die Ausführung großer und sehr tiefer Schaltkreise nötig ist.

Referenzen

[1] Gurobi Optimization, LLC. 2023 Gurobi Optimizer Reference Manual. *https://www.gurobi.com*

Benchmarking quantum algorithms @HLRS

The HLRS has been engaged in benchmarking both quantum simulators on HPC via the cuQuantum (cQ) benchmark libraries as well as exploring metrics for evaluating the current capability and limitations of quantum devices via the application orientated benchmarks suite from the Quantum Economic Development Consortium (QED-C). Our aim was to compare a) how well classical computers are able to emulate quantum circuits with an optimal setup for both the software and hardware, against b) how well NISQ devices are able to perform today. We chose to utilize existing solutions wherever possible and believe the combination of state of the art in quantum circuit emulation libraries from cQ and the insightful volumetric benchmarking approach used by the QED-C [1] gives a coarse grained picture of how much quantum devices need to improve to start to be considered advantageous compared with classical devices.

Volumetric benchmarking of VQE and QAOA

Unlike classical computers, the performance of current quantum devices is primarily constrained by errors at the quantum gate and qubit level rather than size or speed of the processors. The speed and accuracy of a given quantum algorithm is often highly dependent on the architecture of the quantum device, for example, which gates sets the quantum device supports. In fact, taking the same algorithm and carefully compiling it into the native gate sets of different hardware providers will more than likely produce circuits with vastly different circuit depths and therefore runtimes. How to reliably benchmark and measure the performance of such a wide class of devices and algorithms has become an important topic in both industry and academia. New benchmarking

suites and libraries are starting to emerge as a result. One of the most comprehensive is the open source application orientated benchmark suite from the QED-C.

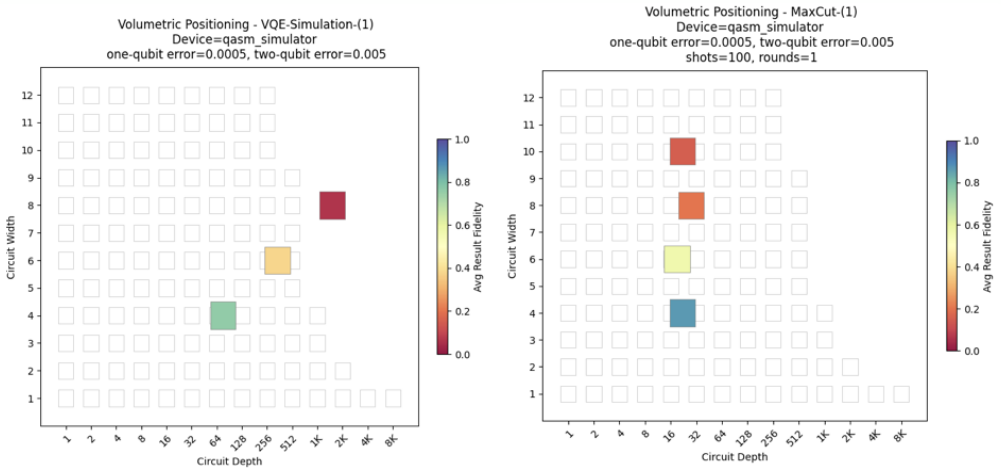


Abb. 78 Volumetric benchmarking plots for VQE simulation (left) and QAOA (right) on a QasmSimulator from Qiskit. Each circuit execution is shown as a coloured square, with the colour being a measure of the result quality, the depth being measured in terms of a universal gate set and the width being the number of qubits.

Here the applications of the VQE applied to a quantum chemistry problem and QAOA applied to the Max-Cut problem were benchmarked using the QED-C benchmarking suite. The algorithms are prepared so that they can scale with circuit size and their performance is measured for in the space of circuit width, i.e. the number of qubits, and normalised circuit depth. The circuit depths are expressed in terms of a universal basis gate set consisting of Rx, Ry and Rz and CX gates, i.e. a normalised circuit depth, with this measure of the depth being the best compromise to compare both different algorithms and different processors. Alternatively measuring depth in terms of the gates an algorithm is expressed in, i.e. algorithmic depth makes it easy to directly visualise how changes to the algorithm affect the result but hard to compare between algorithms. Another measure is the number of gates needed on a real device after the circuit has been transpiled. As the physical operation of different quantum computers is vastly different with this definition it becomes very challenging to compare different processors.

Abb. 78 shows some volumetric benchmarking plots where volumetric simply indicates the space of circuit width and depth is being explored. Abb. 79 shows the performance of the algorithms against various algorithms such as the fidelity, a measure of the closeness of the solution to the ideal solution (the ideal solution is precomputed and assumes a perfect noiseless quantum computer), compile time, execution time and circuit depth (both normalised and algorithmic). Even when using generous error rates which are a lower bound for devices available today the fidelity for both algorithms rapidly drops off at around 8-10 qubits. The compilation and execution times are under a second indicating that the limiting factor is indeed the one and two qubit error rates of the gates.

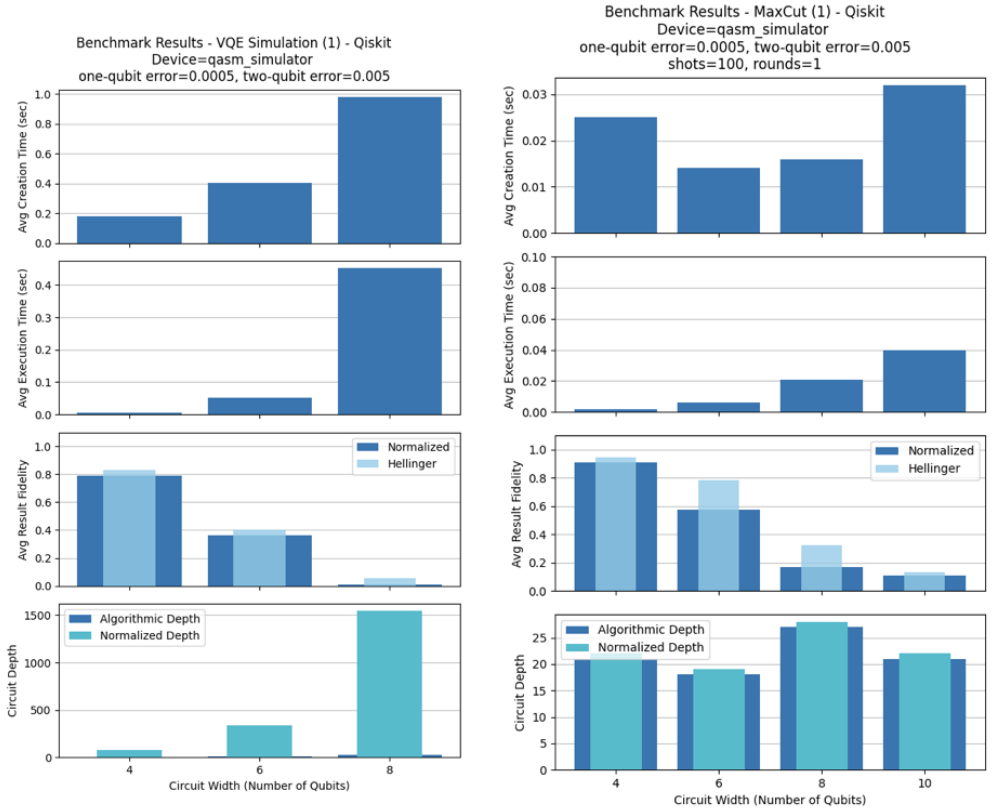


Abb. 79 Bar plots of performance metrics for VQE simulation (left) and QAOA (right) on the QasmSimulator from Qiskit. Increasing circuit widths is plotted against circuit depth, the fidelity (a measure of the result quality), execution time and compile time.

References

[1] https://github.com/SRI-International/QC-App-Oriented-Benchmarks

Benchmarking classical emulation frameworks

Here we benchmark emulation of quantum circuits i.e. running quantum circuits classically with the software optimised for classical hardware. Emulation of quantum circuits will be immensely useful in the NISQ era and beyond for verification of results, calibration of real quantum devices and potentially a tool for resource estimation before deciding to invest in real QPU compute time or devices. Simulating quantum states classically is known to be a memory bound problem with the amount of memory required doubling for each additional qubit simulated when simulating the full state vector (SV). For SV simulators at around 50 qubits, to give a rough ball park figure, classical HPC clusters start to suffer from topology constraints where, for example, the communication overhead becomes so large that it becomes impractical to simulate the quantum state. Tensor network (TN) methods can alleviate the memory constraints associated with simulating the full SV by building a TN representation of the circuit and optimising the path that the TN is contracted over. This is usually done by heuristic algorithms and comes at the cost of extra preprocessing time and also generality as the TN has to be formed for the quantity that you want to compute. However, this approach opens up the possibility to simulate much higher qubit counts (100s and 1000s) for practical quantum circuits and potentially, as these methods improve, deeper quantum circuits as well.

In our benchmarks we explored different i) frontend libraries such as Qiskit and Cirq. Frontends can be thought of as the tool used to write a quantum program. ii) Backend

simulator libraries such as Qiskit Aer and Cirq qsim. Simulator backends can be thought of as the tool that executes said program on classical hardware. iii) The use of the accelerator libraries which can be accessed from simulator backends in much the same way as popular machine learning frameworks like PyTorch leverage accelerator libraries like cuDNN. The lower left part of Abb. 80 gives a simplified overview of the quantum computing stack. The top part of the figure motivates how we may wish to optimally distribute a quantum circuit to different processors. One way of doing this is using bit swapping techniques where e.g. swap gates are inserted so that the only external dependance a given chunk has is via swap gates. The lower right part illustrates one the key steps in a TN based approach. For a more in depth introduction to TNs see [1].

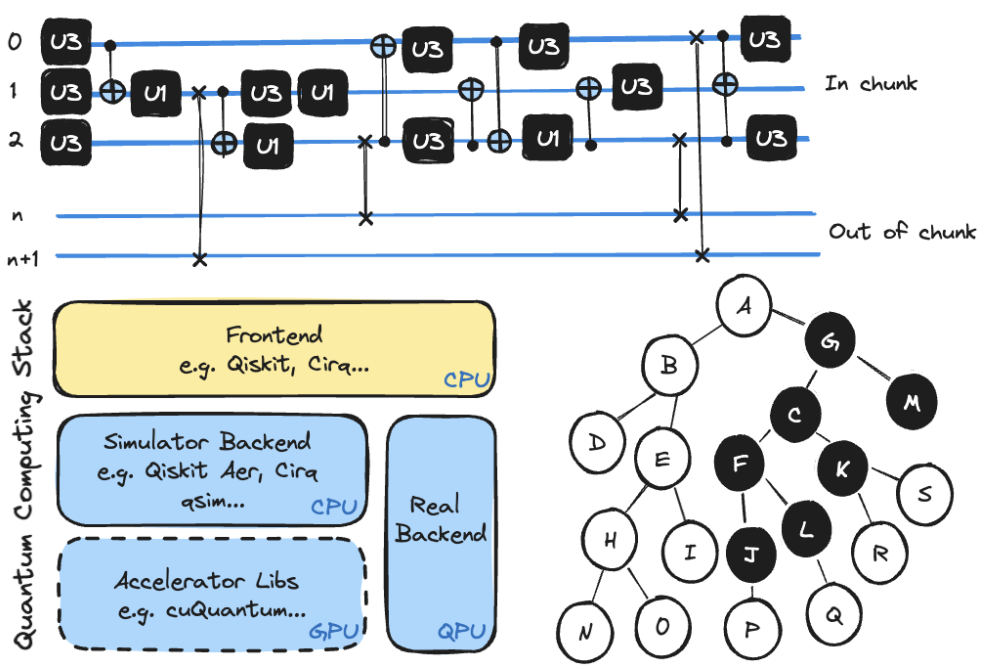


Abb. 80 Upper) Illustration of how to divide the quantum circuit into chunks by inserting swap gates. Lower left) Simple overview of the quantum computing stack. Yellow denotes the frontend and blue shows backend types including both real backends and classical emulation which may in turn utilise accelerator libraries. Dashed lines indicate this part of the stack is not essential to run the circuit. Lower right) Depiction of the slicing and reconfiguration steps when finding the optimal path in tensor network contraction.

Our hardware consisted of server grade AMD EPYC 7702 64-core CPUs and NVIDIA A100 40 GB GPUs with a Infiniband HDR based interconnect. Each node has 8 GPUs and 2 CPUs meaning 128 CPU hardware threads are available in total for this application. The cuQuantum Appliance was used as the container runtime with all software optimised for the correct processor type and node architecture. Within this container we installed the cuQuantum benchmarks framework [2], which already has a robust codebase for recording metrics, accounting for warm up runs, averaging over runs and using optimal methods for recording the performance time. We used their default values whenever possible e.g. averaging over 10 runs with 3 warm up runs. We investigated 3 different classes of scaling study: a) Scaling of the execution time with qubit count for a fixed processor size for both CPU and GPU. b) Multiprocessor performance. For this purpose we looked at weak scaling on GPUs. This measures how the system performs for a fixed problem size per processor. c) Simulation of higher qubit counts by utilising TNs and comparing against a top-performing SV backend. This was done on a single GPU with

both the preprocessing times and execution times being compared. For SV simulation the preprocessing time is the compile time whereas for TNs the preprocessing time consists mainly of the search time for an optimal path (CircuitToEinsum runtime + time to construct TN for a given target e.g. computing a given probability amplitude: in this case the 00…00 amplitude) and the computation of the path itself. The benchmark results presented in Abb. 81 are for the QAOA Max-Cut problem. Other algorithms show similar scaling relationships. Code and documentation to reproduce our results can be found at: https://code.hlrs.de/QC.

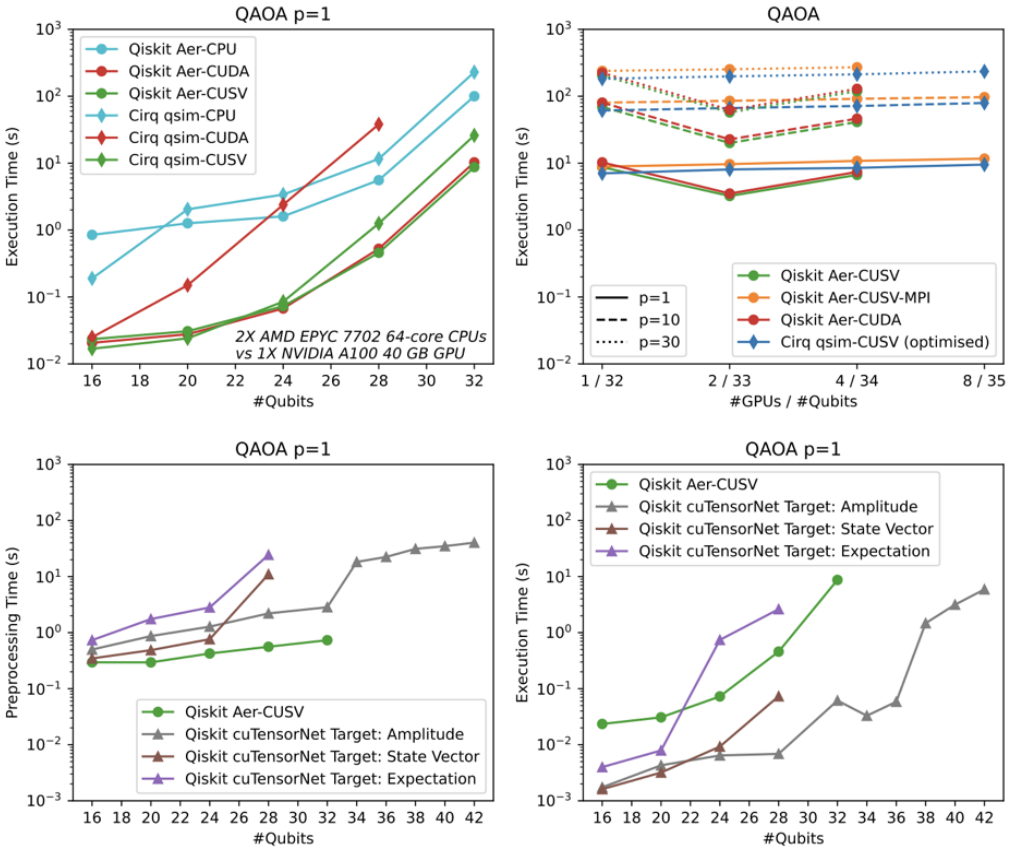


Abb. 81 Benchmarks were executed inside the NVIDIA cuQuantum appliance docker container (cQA) on our Hawk HPC system. Top left) Scaling of the execution time with qubit count for a fixed processor size for both CPU and GPU. Top right) Weak scaling results for GPUs in one compute node. Bottom) Preprocessing times and execution times for tensor networks in comparison with a top performing state vector backend on one GPU. Legend explanation: CUDA runs the native GPU backend for a given library. CUSV runs the cuStateVec integration. CUSV-MPI runs CUSV across multiple processors using MPI. Optimised indicates this backend was previously optimised for performance within the cQA.

For 32-34 qubits the average execution time for QAOA was 7.06 s using GPU acceleration and, for comparison, the time for quantum volume circuit was 25.46 s. Circuits sizes of 32 qubits are executable on one GPU with the execution time being around 10s. In comparison, the execution time on 2 maximally threaded CPUs was around 100 seconds. The GPU provides around a 10X speedup over this CPU setup for most data points. If we instead make a rough comparison against consumer grade CPUs (e.g. mid-range range laptop), we expect at least an order of magnitude difference in performance between consumer grade and server grade CPUs. Therefore, as a rough rule of thumb, one can anticipate on the order of a 100X speed up via GPU acceleration when transitioning from a local workload to an optimal HPC workload. SV simulations

are a highly memory bound problem and at 33 qubits the SV no longer fits into the memory of one GPU. Above this threshold the number of processors needs to be increased in powers of two in order to fit the extra qubits into memory. 35 qubits were executable by using all 8 GPUs on the node with Qiskit Aer's cuStateVec integration with additional MPI parallelisation and Cirq qsim's optimised multi-gpu implementation. Points for which the execution did not finish within a time frame of a few hours were considered failures and are not included. E.g. implementations without the additional parallelisation did not finish for 35 qubits in the weak scaling study. Most benchmarks we run with a QAOA depth parameter p=1 unless specified otherwise. Using a Qiskit frontend this corresponds to a gate count of 1553 including state preparation and measurement gates for 32 qubits for p=1. For the same settings but with p=30 the total number of gates was 45633. Overall we observed ideal scaling relationships for both the execution time vs qubit count study and the weak scaling study.

Utilising TN based methods yielded a substantial increase in the number of qubits which could be simulated on one GPU alongside a correspondingly large decrease in the memory required. When using the amplitude as a target 42 qubits were simulable. In comparison a full SV simulation of the same number of qubits would have required 128 nodes with 8 GPUs per node. As before qubit counts that did not finish within a reasonable time cut off were excluded, beyond 42 qubits we found the simulation started to become more unstable, sometimes finishing within a reasonable cut-off time and sometimes not, often this was accompanied by highly unfavourable scaling behaviour for these points e.g. the time for the benchmark to execute increasing by an order of magnitude for a small increase in qubit size. We also observe less regular scaling behaviour below 42 qubits which is to be expected as circuits for some qubit counts could have more optimal contraction paths than others. Overall the significant gains in execution time and reductions in memory requirements are accompanied by a significant increase in the preprocessing time. For most data points, the preprocessing time is significantly larger than the execution time, so much so that there is little observable difference for the TN based approaches when plotting total time instead of preprocessing time.

Conclusion

Our aim was to compare the current state of classical emulation of quantum circuits to their execution on quantum devices. In the quantum case we found that the error rates of the gates are a strong limiting factor. Circuits execution with widths of just 8-10 qubits had very poor fidelity. Furthermore, generous error rates were chosen and the noise model likely does not capture some of the more complex error sources present on real devices. Therefore, based on these results alone, just 10 qubits seems a reasonable upper bound for the sizes of quantum circuits that can be executed reliably on quantum devices today, regardless of if they have higher available qubit counts. Contrastingly, in the classical case, 30 qubit SV simulations could run in around 10s on server grade GPUs. However, at a certain qubit count (32 for QAOA) every extra qubit requires the number of GPUs to be doubled. TN methods could alleviate the memory constraints but at the cost of much higher preprocessing time. If this trade-off between memory and time is managed well it is reasonable to assume, at least for shallow circuits, that circuit widths of 100+ qubit circuits can be straightforwardly simulated classically.

[1] https://docs.nvidia.com/cuda/cuquantum/23.03.0/cutensornet/overview.html
[2] https://github.com/NVIDIA/cuQuantum/tree/main/benchmarks

End-to-End Demonstrator Resilienzanalysen

Insgesamt wurden vier verschiedene Netzwerke untersucht. Die Details Tabelle 7 können entnommen werden. Für die komplexeren Systeme, bei denen ein drei-Level Knoten an einen, bzw. zwei weitere Knoten gekoppelt wurde, wird jeweils nur ein Unterraum des eigentlichen Hilbertraums betrachtet, der maximal eine Anregung enthält, sodass die Anzahl der benötigten Qubits weniger stark ansteigt. Bei der Tiefe der Circuits wird deutlich, dass die Hinzunahme eines dritten Knotens im Falle eines offenen Quantensystems einen enormen Anstieg in der Komplexität der Simulation bedeutet. Für Circuits der Tiefe 399 sind die Ergebnisse sehr stark von Rauschen dominiert, während beim nächstkleineren System und einer Circuit Tiefe von 87 noch qualitativ gute Ergebnisse erhalten wurden.

Anzahl an drei-Level Knoten	Anzahl an zwei-Level Knoten	Benötigte Qubits	Circuit Tiefe
0	3	3	67
1	0	3	67
1	1	3	87
1	2	4	399

Tabelle 7 Spezifikationen der untersuchten Netzwerke.

2.5 Arbeitspaket 5 – Quanten-Software-Engineering

Definiertes Ziel des fünften Arbeitspakets ist laut Projektbeschreibung die Verbesserung des Verständnisses und der Dokumentation beim Vorgehen zur Entwicklung hybrider Quantensoftware. Fokus liegt dabei auf (1) dem Verständnis der Parameter und Freiheitsgrade bei Auswahl und Auslegung von Algorithmen und Transpilation und (2) der Auswahl und Nutzung der passenden Werkzeuge, entsprechend den Erfordernissen der unterschiedlichen Phasen im Entwicklungsprozess. Im Fokus der Untersuchung von Werkzeugen und Methoden stehen die bisher wenig erforschten Felder »Test und Debugging« sowie die Bereitstellung von Quantenanwendungen für Unternehmen.

Unter Leitung des **Forschungszentrums für Informatik FZI** wird dies anhand von vier Arbeitsschritten realisiert (vgl. Gantt-Chart in Abb. 2)

AP 5.1 Design Space Exploration

- Formalisierung der Parameter bei Auswahl und Auslegung von Algorithmen sowie insbesondere der Transpilation (Aufgreifen von Erkenntnissen aus AP2 und AP3)
- Bereitstellung von Referenzanwendungen zur Evaluierung der Design Space Exploration (Nutzung Erkenntnisse/Ergebnisse aus AP 4)
- Erweiterung eines Werkzeugs zur Design Space Exploration für hybride Quantensoftware
- Evaluierung der Design Space Exploration für die Referenzanwendungen
- Dokumentation der Erkenntnisse zu einzelnen Parametern aus der Evaluierung

AP 5.2 End-to-End-Lösungen: Werkzeuge und Werkzeugketten

- Werkzeuge zur effizienten Ausführung variationeller Algorithmen (z.B. Runtime)
- Werkzeuge zur einfachen, automatischen Ausführung von Quanten-Algorithmen auf realen Backends (Schaltkreisgenerierung, Pulsoptimierung, Job-Management)
- Einbindung von Werkzeugen in Transpilationspipelines
- Werkzeuge zur Arbeit mit verschiedenen SDKs

AP 5.3 Test und Debugging

- Entwicklung einer Testtheorie für Quantenprogramme
- Werkzeugunterstützung zur Testfallerzeugung
- Bewertung von erzeugten Testfällen

AP 5.4 Konzept zum Deployment und PlanQK-Integration

- Konzept zur Bereitstellung der erarbeiteten Software via Quantensoftware-Plattformen wie z.B. PlanQK (exemplarisch)

Jenem Arbeitsplan liegen dabei zwei Meilensteine zugrunde

- **M 8:** Konzept zur Design Space Exploration von Quantensoftware ist bereitgestellt (Publikation erstellt), Testtheorie inklusive Werkzeugunterstützung für Quantensoftware wurde entwickelt (Open Source)
- **M15:** Ergebnisse der Analyse von Entwurfsparametern von Quantensoftware ist bereitgestellt (Publikation erstellt), Erweiterung für Werkzeug zur Design Space Exploration ist veröffentlicht (Open Source), Testkonzept qualitativ und quantitativ wurde bewertet (Publikation erstellt)

Meilensteine M8 und M15 wurden planmäßig erreicht (vgl. Abb. 2). Das Konzept zur Design Space Exploration von Quantensoftware ist bereitgestellt und veröffentlicht.

Im Folgenden wird der Endstand der Aktivitäten dargestellt.

Design Space Exploration

Im Rahmen von Arbeitspaket 5 wurden in einem ersten Schritt die verschiedenen Parameter und Freiheitsgrade analysiert, die im Laufe der Entwicklung von Quanten-Software-Lösungen entstehen und berücksichtigt werden müssen. Hierbei wurde festgestellt, dass in allen Phasen und auf verschiedenen Ebenen des Entwicklungsprozesses, Entwurfsentscheidungen entstehen, die einen Einfluss auf die Qualität des Quanten-Software-Lösung haben. Die Ergebnisse der Analyse wurden veröffentlicht, siehe (1).

Die Erkenntnisse aus (1) dienen somit als Grundlage für das AP5.1 »Design Space Exploration«. Eine der wesentlichen Erkenntnisse aus (1) war, dass Entwurfsentscheidungen in allen Entwicklungsphasen auftreten; eine holistische Entwurfsraumexploration ist daher nicht möglich, da zu viele Ebenen involviert sind. Somit wurde der Fokus in AP5.1 auf Entwurfsentscheidungen gelegt, die ausschließlich der Reduzierung von NISQ-induzierten Unsicherheiten dienen. Hierfür wurde ein Framework umgesetzt, das auf (3) aufbaut und erweitert. In (3) werden verschiedene fehlertoleranten Architekturmuster diskutiert, die im Kontext des Quantencomputing angepasst und eingesetzt werden, um akkurate Messungen zu erzielen und NISQ-induzierten Unsicherheiten vorzubeugen. In AP5.1 werden diese Arbeiten für weitere Architekturmuster und Fehlererkennungsmechanismen erweitert. Grundlage bildet hierbei ein Software-Framework, das der Community als Open-Source Software bereitgestellt wird, welches erlaubt verschiedene fehlertolerante Architekturmuster zu konstruieren und zu evaluieren. Der Entwurfsraum wird hierbei durch die verschiedenen Muster sowie den verschiedenen Freiheitsgraden innerhalb der Muster aufgespannt. Mittels des Frameworks können nun zum einen (i) verschiedene Muster konstruiert sowie (ii) unterschiedliche Varianten der Muster erprobt werden. Dies ermöglicht verschiedene Varianten leichtgewichtig zu konstruieren, zu explorieren und zu bewerten.

Konkret wird die Arbeit in (3) um zwei Architekturmuster und drei Fehlererkennungsmechanismen erweitert. Grundsätzlich besteht jedoch die Möglichkeit, dass Framework, um weitere Muster und Fehlererkennungsmechanismen zu erweitern. Als Architekturmuster wurden das *Voter-*, *Sparing-* und *Comparison*-Muster umgesetzt. Für jedes Muster hat sich grundsätzlich die Frage gestellt, inwiefern sich heterogene und homogene Redundanz bei Quantenkomponenten ausprägen. Hierfür wurden verschiedene Ansätze diskutiert und umgesetzt. Das Voter-Muster besteht aus *N* redundanten Varianten einer Quantenkomponente. Eine Eingabe wird somit an jede Quantenkomponente $i \coloneqq 1, \dots, N$ weitergeleitet und ein Resultat bzw. Messung erzeugt. Die *N* Messungen werden in einer Voter-Komponente zusammengeführt. Als Voter-Komponente wurde hierbei ein *Linear Opinion Pool* verwendet, der maßgeblich die *N* Messungen zu einer akkurateren Messung bzw. Wahrscheinlichkeitsverteilung aggregiert. Das Sparing-Muster hat grundsätzlich denselben Aufbau wie das Voter-Muster, nur das jede redundante Quantenkomponente mit einer Fehlererkennungskomponente ausgestattet wird. Statt einer Voter- kommt eine Switch-Komponente zum Einsatz, die aus der Menge der *N* Messungen und den *N* Ergebnissen der Fehlererkennungskomponente entscheidet, welche Messungen weiterhin verwendet wird. Während das Voter- und Sparing-Muster hauptsächlich dazu dienen Fehler in Messungen zu tolerieren, dient das Comparison-Muster dazu Fehler zu erkennen. Konkret werden beim Comparison-Muster zwei (idealerweise heterogene) redundante Quantenkomponenten betrachtet und ihre Messungen miteinander verglichen. Durch den Vergleich der Messungen können nun Fehler erkannt werden, da die Messungen sich sonst entweder widersprechen oder übereinstimmen.

Da insbesondere das Sparring-Muster von Fehlererkennungsmechanismen bzw. -komponenten abhängt, wurden im Rahmen von AP5.1 drei Ansätze betrachtet. Der erste Ansatz umfasst das Comparison-Muster, da dieses der reinen Fehlererkennung dient. Als zweiter Mechanismus wird eine reine Ergebnisprüfung betrachtet. Bei manchen Problemstellungen bzw. verwendeten Quantenalgorithmen lässt sich die Validität der erzeugten Lösungen direkt prüfen, z.B. bei SAT-Problemen; dies ist jedoch nur für einen kleinen Bruchteil von Lösungen bzw. Algorithmen möglich. Als dritter Ansatz werden Distanzfunktionen für Wahrscheinlichkeitsverteilungen herangezogen, um Messungen hinsichtlich Rauschartefakte und Qualität zu bewerten. Hierbei wird eine Familie von Distanzfunktionen betrachtet, die sogenannten *f-Divergenzen*. Eine f-Divergenz $D_f\,(P\,||\,Q)$, definiert über die Verteilungen *P* und *Q*, hat allgemein die folgende Struktur:

$$D_f\,(P\,||\,Q) = \mathbb{E}_Q\left[f\left(\frac{dP}{dQ}\right)\right]$$

Für unterschiedliche Funktionen f können verschiedene Distanzfunktionen erzeugt werden; beispielweise erzeugt $f(x) = x \cdot \log(x)$ die Kullback-Leibler-Divergenz. Mittels einer geeigneten f-Divergenz kann nun der Abstand einer Messung zu einer Referenzverteilung ermittelt werden. Beispielsweise lässt sich der Abstand zu einer Gleichverteilung berechnen, wobei eine Gleichverteilung eine maximal verrauchte Messung repräsentiert; je näher der Abstand zur Gleichverteilung, desto höher die Wahrscheinlichkeit einer fehlerhaften Messung bzw. Ausführung des Quantenalgorithmus.

Für eine umfangreiche Evaluierung des Ansatzes wurde die MQT-Benchmark-Bibliothek (MQT Bench) verwendet. MQT Bench umfasst verschiedene Quantenalgorithmus-Implementierungen, die von Deutsch-Jozsa bis zu Algorithmen der Amplituden-Schätzung reichen. Jede Implementierung kann von 2 bis 130 Qubits skaliert werden und wird als QASM-Datei bereitgestellt. Die Liste der angebotenen Quantenalgorithmus-Implementierungen ist öffentlich zugänglich[1]. Für unser Experiment haben wir alle bereitgestellten Algorithmen berücksichtigt, die aus 2 bis 10 Qubits bestehen. Für das Experiment wurden sechs unterschiedliche Muster (wie oben beschrieben) mittels des Frameworks definiert und evaluiert (auf Basis von MQT Bench). Hierbei wurde jeder Quantenalgorithmus einmal simuliert (um die korrekte Ausgabe zu ermitteln) und direkt auf dem Quantencomputer ausgeführt. Anschließend wurde jedes Muster auf jeden Algorithmus angewandt und auf dem Quantencomputer ausgeführt. Um schließlich einen direkten Vergleich zu schaffen, wurde die Anzahl der korrekten Ausführungen der Muster den Ausführungen der einzelnen Algorithmen gegenübergestellt. Eine vereinfachte Übersicht der Ergebnisse ist in der folgenden Tabelle abgebildet:

Muster ID	<	=	>
C_{seed}	0.17	0.67	0.16
C_{back}	0.13	0.71	0.16
C_{opt}	0.16	0.61	0.23
S_{noise}	0.17	0.70	0.13
S_{com}	0.19	0.65	0.16
M_{hyb}	0.38	0.60	0.02

Zusammenfassend hat die Evaluierung gezeigt, dass unser Ansatz bzw. Framework es erlaubt verschiedene Muster und Entwurfsentscheidungen von Mustern zu explorieren

[1] https://www.cda.cit.tum.de/mqtbench

und zu evaluieren. Das Framework ist Open-Source und wird der Community bereitgestellt. Weiterhin bietet das Framework verschiedene Erweiterungspunkte, so dass weitere Freiheitsgrade bzw. Entwurfsoptionen integriert werden können. Hinsichtlich der Evaluierung der einzelnen Muster haben sich lediglich drei Varianten des Voter-Musters als vielversprechend erwiesen (siehe Tabelle Zeile 1-3), die annäherungsweise dieselbe oder eine höhere Ausführungsgenauigkeit (wie einzelne Quantenalgorithmen) erreichen.

Der Meilenstein M8 wurde maßgeblich durch die Veröffentlichung (1) sowie der Bereitstellung des Frameworks als Open-Source-Software erreicht.

Zusätzlich wurde eine Methode zur Qualitätssicherung hybrider Quantensoftware im Rahmen von AP 5.2 bzw. AP 5.3 entwickelt. Durch die hohe Komplexität von Quantenalgorithmen, sowie die enorme erforderliche Expertise bei deren Entwicklung, ist Qualitätssicherung im Kontext von Quantensoftware besonders wichtig. Die entwickelte Methode basiert auf der Übersetzung von Quantenalgorithmen in einen äquivalentes Javaprogramm. Ein auf diesem Wege generiertes Javaprogramm kann dann mit klassischen Analysemethoden untersucht werden und somit indirekt die Qualitätssicherung des Ursprungsprogramms betrieben werden. Hierfür wurden zuerst Syntax und Semantik einer generischen Quantenschaltkreissprache sowie einer minimalen While-Sprache definiert. Die Übersetzung kann dann als eine Menge syntaktischer Transformationen zwischen diesen beiden Sprachen beschrieben werden. Für diese theoretische Methode wurde die Äquivalenz der Übersetzung zu dem ursprünglichen Quantenschaltkreis bewiesen. Als nächster Schritt wurden die Lücken zwischen der While-Sprache und einer real existierenden Programmiersprache untersucht. Die größten Herausforderungen waren hierbei der durch Messungen induzierte Nichtdeterminismus von Quantenprogrammen sowie die Tatsache, dass Quantenprogrammen mit komplexen/reellen Zahlen operieren, während die meisten klassischen Programmiersprachen nur Fließkommazahlen unterstützen. Letzteres kann dazu führen, dass durch Fließkommazahlen Rundungsfehler entstehen, die schlimmstenfalls das Ergebnis verfälschen können. Diese Befürchtung konnte durch Experimente entkräftet werden, wenn auch nicht endgültig ausgeschlossen. Es zeigte sich, dass für viele Quantenprogramme sowohl Fehler entdeckt als auch korrekte Programme als solche verifiziert werden konnten. Eine formale Analyse der Natur der möglichen Rundungsfehler sowie daraus resultierend eine Aussage, ob Rundungsfehler überhaupt möglich sind, wurde untersucht und veröffentlicht, siehe (4, 5). Um den Nichtdeterminismus aufzulösen, wurden zwei Möglichkeiten untersucht.

A) es werden nicht alle möglichen, sondern nur das wahrscheinlichste Messergebnis berücksichtigt. Dies erlaubt nicht alle theoretisch möglichen Eigenschaften von Quantenprogramm zu testen allerdings sind Quantenalgorithmen aus offensichtlichen Gründen meist so konzipiert, dass das gewünschte Ergebnis mit der höchsten Wahrscheinlichkeit auftritt. Die Eigenschaft, dass das wahrscheinlichste Messergebnis also bestimmter Natur ist, ist somit eine relevante Eigenschaft.

B) Es können in der Übersetzung spezielle Sprachkonstrukte benutzt werden, die es erlauben den Nichtdeterminismus von den eingesetzten Analysewerkzeugen auflösen zu lassen. Dies macht eine der Stärken des vorgestellten Ansatzes deutlich: Basierend auf der Übersetzung kann jede beliebige Analysemethode angewendet werden, die für die entsprechende Sprache anwendbar ist. Konkrete gezeigt wurde im Rahmen dieses Projektes, dass die Methode für Unit-Tests und für einen Bounded-Model-Checker (eine Art von vollautomatischem Verifikationswerkzeug) geeignet ist. Somit konnten mehrere bekannte Quantenalgorithmen (unter anderem Grover, Shor und Deutsch-Jozsa) sowohl getestet als auch vollständig verifiziert werden. Weitere Vorteile dieses Ansatzes sind, dass durch die vollständige Automatisierbarkeit kein zusätzliches Expertenwissen nötig ist und auch hybride Quantensoftware (mit klassischen Codeabschnitten) verifiziert werden kann. Außerdem ist die Vorgestellte Methode ein direkter Beitrag zu einer

durchgehenden Werkzeugkette für Quantensoftware, da sie dank ihrer vollautomatischen Natur direkt Teil einer CI/CD-Pipeline werden könnte.

Der Ansatz wurde veröffentlicht und auf mehreren internationalen Konferenzen vorgestellt (siehe (2)). Außerdem ist eine prototypische Implementierung der Übersetzung für Java als Open-Source-Software verfügbar.

Referenzen

(1) M. Scheerer, J. Klamroth, S. Garhofer, F. Knäble and O. Denninger, "Experiences in Quantum Software Engineering," *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, St. Petersburg, FL, USA, 2023, pp. 552-559, doi: 10.1109/IPDPSW59300.2023.00095.

(2) J. Klamroth, B. Beckert, M. Scheerer and O. Denninger, "QIn: Enabling Formal Methods to Deal with Quantum Circuits," *2023 IEEE International Conference on Quantum Software (QSW)*, Chicago, IL, USA, 2023, pp. 175-185, doi: 10.1109/QSW59989.2023.00029.

(3) M. Scheerer, J. Klamroth and O. Denninger, "Fault-tolerant Hybrid Quantum Software Systems," *2022 IEEE International Conference on Quantum Software (QSW)*, Barcelona, Spain, 2022, pp. 52-57, doi: 10.1109/QSW55613.2022.00023.

(4) J. Klamroth, B. Beckert, "On Rounding Errors in the Simulation of Quantum Circuits," *2023 Workshop on Services and Quantum Software (SQS)*, Rom, Italy, 2023. To appear.

(5) J. Klamroth, B. Beckert, "Bounding Rounding Errors in the Simulation of QUantum Circuits" *2024 IEEE International Conference on Quantum Software (QSW)*, Shenzhen, China, 2024. To appear.

Werkzeuge und Werkzeugketten für End-to-End-Lösungen

Die Forschungsergebnisse des vorherigen Projekts »SEQUOIA« umfassen einen Quanten-Software-Komponentenbaukasten, der als Grundlage für die Umsetzung von Quantenanwendungen dient, sowie ein Quanten-Software-Engineering-Modell (Quantumsoftware-Lifecycle). Hierbei wurde festgestellt, dass mit aktuellen Software Development Kits (SDKs), wie z. B. Qiskit, eine praktische Umsetzung von Quantenanwendungen auf High-Level-Ebene nicht möglich ist. Darüber hinaus ergab sich, dass gegenwärtig die Disziplin »Quantencomputing« stark wissenschaftlich geprägt ist und sich die Entwicklung von Software für Quantencomputer noch in einem frühen Entwicklungsstadium befindet. Dies bedeutet, dass wenige etablierte Methoden und Werkzeuge im Quanten-Software-Engineering vorhanden sind im Gegensatz zum klassischen Software-Engineering. So sind im Bereich des Quantencomputings State-of-the-Art Techniken aus »Clean Code« und »Software Design Patterns« (Entwurfsmuster) nur sporadisch im Einsatz. Um zukünftig Unternehmen eine effiziente Entwicklung sowie Integration von Quantenanwendungen zu ermöglichen, ist ein transparenter und sich stetig weiterentwickelnder Quanten-Software-Entwicklungsprozess essenziell. Ziel ist es daher diesem Defizit entgegenzuwirken, indem gegenwärtig bis zum Ende der Projektlaufzeit Techniken aus Clean Code und Software Design Patterns innerhalb der Quanten-Software-Engineering-Pipeline untersucht werden.
In der klassischen Softwareentwicklung führte die Einführung von Clean Code sowie Software Design Patterns zu einer erheblichen Verbesserung der Qualität bei der Entwicklung von Softwarelösungen. Der Einsatz dieser Techniken ermöglicht Softwareanwendungen mit robusteren und flexibleren Softwarearchitekturen. Die Software Design Patterns stellen eine Art »Bauplan« dar und sind bewährte Lösungsansätze für häufig wiederkehrende Probleme in der Softwareentwicklung. So ermöglichen sie es den Entwicklern wiederkehrende Probleme effizient zu lösen,

wodurch der Entwicklungsprozess beschleunigt sowie die Qualität des Quellcodes verbessert wird. Weiterhin wird die Skalierbar-, Anpassbar- und Erweiterbarkeit der Software erleichtert, da der Quellcode modularisiert vorliegt. Somit können beispielsweise der Softwareanwendung neue Funktionen einfacher hinzugefügt werden. Sowohl durch den sauberen als auch strukturierten Aufbau des Programmcodes wird dessen Verständlichkeit, Wartbarkeit und Fehlerbehebung deutlich erleichtert. Auch verringert sich die Wahrscheinlichkeit von vorhandenen Fehlern in der Softwarelösung. Ferner können durch den Einsatz von etablierten Clean Code und Design Patterns Techniken Softwareentwickler leichter miteinander kollaborieren, kommunizieren sowie Lösungsideen austauschen. Zudem verringert sich die Gefahr von Missverständnissen während des Software-Engineering Prozesses.

Aufbauend auf den State of the Art Techniken aus Clean Code und Software Design Patterns im Rahmen klassischer »Software Technology and Engineering« (STE), umfassen die derzeitigen Forschungsarbeiten Untersuchungen von Best Practices für Clean Code und Software Design Patterns im Hinblick auf die neu entstehende Fachdisziplin »Quantum Software Technology and Engineering« (QSTE). Dabei liegt zunächst ein besonderes Augenmerk auf der Übertragbarkeit und Anwendbarkeit von Best Practices aus dem klassischen Clean Code und den Software Design Patterns in das Quanten-Software-Engineering. Da Quantencomputer eine andere Funktionsweise gegenüber klassischen Computern haben, steht zu erwarten, dass sich ausschließlich bestimmte Techniken aus dem klassischen Software-Engineering in das Quanten-Software-Engineering übertragen und anwenden lassen. Um die Herausforderungen der effizienten und transparenten Entwicklung von anwendungszentrierten End-to-End-Quanten-Softwarelösungen bewältigen zu können, wird es voraussichtlich erforderlich sein, spezifische QSTE-Guidelines und Best Practices für Clean Code sowie Software Design Patterns zu definieren, die im Augenblick in der klassischen STE nicht existieren.

Die in den kommenden Wochen im Rahmen des Quanten-Software-Engineerings entstehenden Erkenntnisse und Forschungsergebnisse über Best Practices in Clean Code sowie Software Design Patterns werden in Hinsicht auf AP 6 in das bestehende QC-Schulungsprogramm integriert und den Projektpartnern, dem assoziierten Unternehmensnetzwerk sowie den Wissenschaftlern im KQCBW zur Verfügung gestellt.

The Q-Ctrl – Error Suppression Software Tool

Quantum computers are highly susceptibility to noise and errors, especially today's noisy intermediate-scale quantum (NISQ) devices, which critically limits the performance and capabilities of NISQ as well as any future quantum computing devices. Quantum control deals with efficient execution of quantum logic operations and quantum algorithms. Robustness to errors can be built into quantum algorithms with quantum control software tools enabling the application and integration of quantum control in quantum computing research. Such software tools not only serve the needs of algorithm developers but also of hardware R&D teams and end users.

Here we have tested the capabilities of the software tool from Q-Ctrl, specifically FIRE-OPAL (FO) [1], in suppression of the errors incurred in execution of quantum algorithms on NISQ devices. FO is a Python package that applies a complete suite of error suppression techniques to improve the quality of quantum algorithm results. We evaluate FO on the Quantum Phase Estimation algorithm.

Quantum Phase Estimation (QPE): The QPE algorithm is used as a subroutine in many important quantum algorithms including the simulation of chemical materials such as drugs or fertilizers and Shor's algorithm for RSA decryption. QPE provides an estimate on

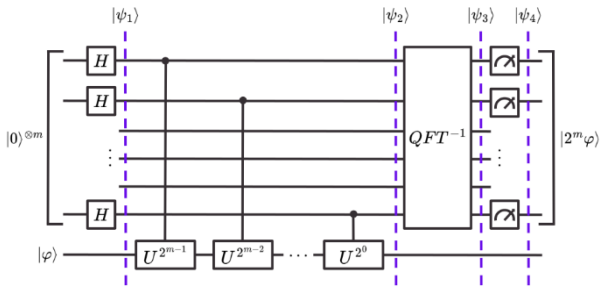


Abb. 82 The Quantum Phase Estimation circuit diagram

the eigenvalues of a unitary operation U. For an n qubit unitary U, the QPE algorithm finds the phase λ in the eigenvalue equation $U|\varphi\rangle = e^{2\pi i\lambda}|\varphi\rangle$. The algorithm needs m additional "counting qubits" where λ is encoded as a binary string. The phase estimation error ϵ becomes exponentially small with the number of counting qubits as, $\epsilon = O\left(\frac{1}{2^m}\right)$. In Abb. 45 the quantum circuit of QPE is shown. It can be seen in this circuit that the quantum circuit implementing U is applied $O\left(\frac{1}{\epsilon}\right) = 2^m$ times. For an input state vector $|\psi\rangle$, which is not an eigenstate of U, QPE algorithm projects the input qubits to a specific eigenstate $|\varphi_j\rangle$ of U with a probability p_j=$|\langle\psi|\varphi_j\rangle|^2$ and computes the corresponding phase φ_j. Thus, QPE samples the eigenvalues of U from a probability distribution defined by the input state vector. By repeating the algorithm many times, an estimate of all eigenvalues of U with nonzero p_j can be obtained.

Problem setup

A QPE circuit is constructed to estimate the value of a given phase (λ) and is executed four times for different numbers of counting qubits. These circuits are run four times, 1) on a quantum simulator which give exact results without any errors, 2) on the IBMQ hardware with no error mitigation, 3) with best error mitigation protocols available within Qiskit, and 4) on the IBMQ hardware using the FIRE-OPAL error suppression protocol. We then evaluate the capabilities of the FIRE-OPAL software tool by comparing the results. We use the 16 qubit *ibmq_guadalupe* system for our experiments.

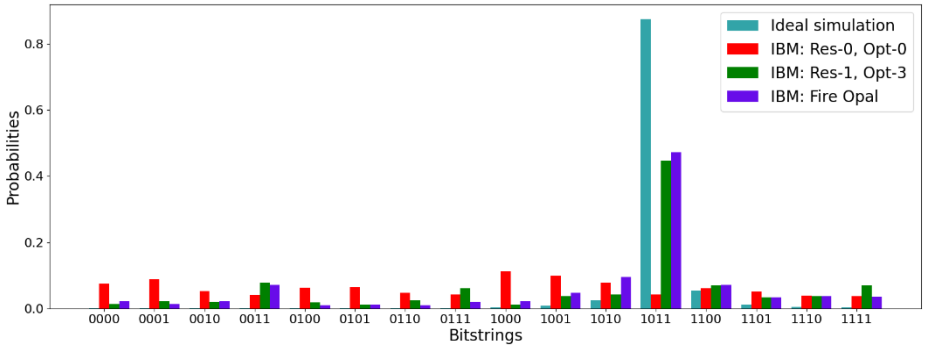


Abb. 83 Four counting qubits ($m = 4$) estimating $\lambda = 0.7$ -- Comparison of the probability distribution obtained from simulation, on IBM backend with no circuit optimization and no error mitigation, with optimization level set to 3 and resilience to 1 and on IBM backend with FIRE-OPAL error suppression protocol. $\lambda = \frac{q}{2^m}$, where q is the encoded integer in the bitstring. The estimated $\lambda = \frac{11}{16} = 0.6875$ with error $\epsilon = 0.0125 < \frac{1}{2^4} = \frac{1}{16} = 0.0625$.

Results

In Abb. 46 the probability distributions obtained for QPE circuit estimating $\lambda = 0.7$ with four counting qubits ($m = 4$) are shown. We see that the execution on the IBMQ hardware results in a random probability distribution. Using the highest circuit optimization level (optimization_level=3) and noise/error resilience (resilience_level=1, best possible when using sampler primitive), the probability distribution resembles that of the simulated values. The results are slightly better but not really game changing when FIRE-OPAL error suppression protocol is used compared to results obtained with Qiskit provided noise mitigation algorithms. Thus, it can be concluded that for this problem the FIRE-OPAL tool does not provide any additional benefit over Qiskit algorithms.

To further investigate the FIRE-OPAL tool we performed another experiment where we estimate $\lambda = 0.05305$. Here, a relatively small value is being estimated and thus a minimum number of counting qubits, $m > 4$, is needed such that the error in the estimation is smaller than the value to be estimated. We choose $m = 6$ in this experiment such that error $\epsilon = \frac{1}{2^6} = 0.015625$ is less than $\lambda = 0.05305$. In Abb. 47 two probability distribution plots obtained from two different runs (*a* and *b*) are shown. As shown in Abb. 47a, the IBMQ hardware results with highest circuit optimization level and error resilience level faithfully reproduce the simulated probability distribution. However, in Abb. 47b, the probability distribution results are indecisive, the probabilities for bitstrings values 3 and 4 are 0.047 and 0.049, respectively.

Execution on IBMQ hardware without any optimization or error resilience again produce random distributions. On the other hand, with FIRE-OPAL error suppression protocols, the probability distribution results in both Abb. 84 (a) and (b) are more decisive (the bitstrings values 3 and 4) and consistently resemble the simulation.

Summary and Conclusion

We have tested the FIRE-OPAL error suppression software tool from Q-Ctrl on the quantum phase estimation circuit. We find that results from IBMq hardware without any error mitigation or circuit optimization protocols are random. The recent developments of circuit optimization and noise resilience algorithms in Qiskit enables running circuits on IBMq hardware that produce decisive probability distributions. However, FIRE-OPAL

error suppression tool from Q-Ctrl outperforms in problems where multiple basis states contribute with similar weights in the wavefunction.

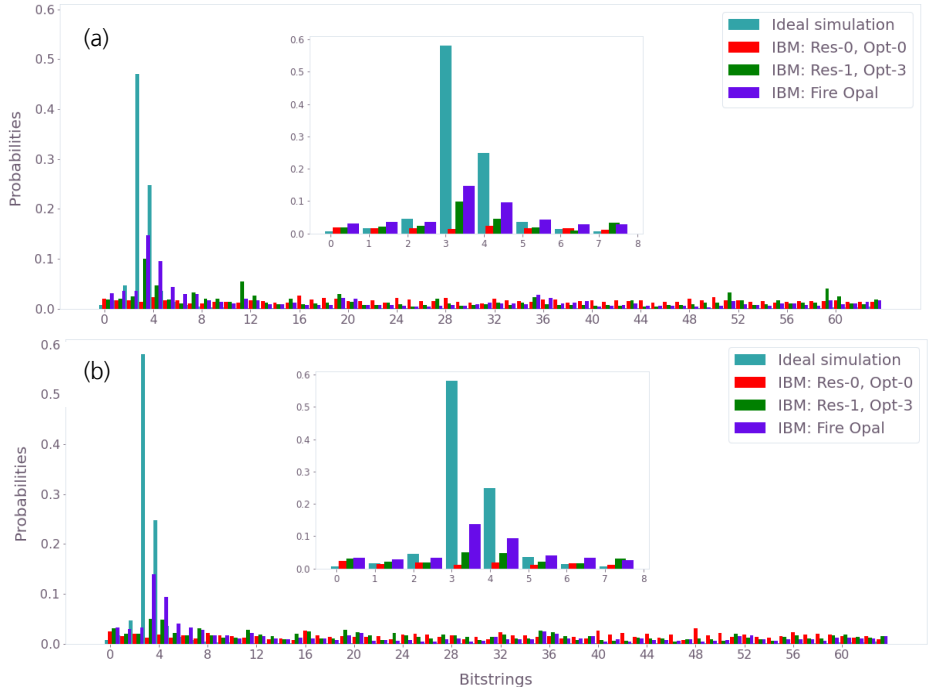


Abb. 84 Six counting qubits ($m = 6$) estimating $\lambda = 0.05305$ -- Comparison of the probability distribution obtained from simulation, on IBM backend (*ibmq_guadalupe*) with no circuit optimization and no error mitigation, with optimization level set to 3 and resilience to 1 and on IBM backend with FIRE-OPAL error suppression protocol. The estimated $\lambda = \frac{q}{2^m} = \frac{3}{64} = 0.04687$ with 0.581 probability and error $\epsilon = 0.006175$ or $\frac{4}{64} = 0.0625$ with probability of 0.248 and $\epsilon = 0.00945$ ($\epsilon < \frac{1}{2^6} = \frac{1}{64} = 0.015625$). The plots (a) and (b) corresponds to two different runs where a difference in the probability distribution is observed, see text. For brevity the bitstrings are shown with their integer equivalents as x-axis labels. The insets show a zoomed version of the plot in the region between x values of 0 and 8.

References

[1] P.S. Mundada, A. Barbosa, S. Maity, Y. Wang, T. Merkh, T.M. Stace, F. Nielson, A.R. Carvalho, M. Hush, M.J. Biercuk, Y. Baum, Phys. Rev. Applied **20** (2023

Automatische Quantenschaltkreis-Synthese mittels Classiq-SW

Die Classiq-Plattform ist eine Software-Lösung, die Quantenschaltkreise aus einem High-Level Model synthetisieren kann und die Schnittstelle zu Hardwareanbietern darstellt. Das Fraunhofer IAO hat die Software mit einer akademischen Lizenz testen und die Eignung für ihre Forschung ermitteln können. Dabei wurde herausgefunden, dass die Plattform für die momentanen Forschungsfragen ungeeignet ist. Classiq ist darauf ausgelegt, ohne explizites Wissen über Quantenschaltungen, Quantenalgorithmen zu synthetisieren und somit auch sehr komplexe Algorithmen zu implementieren. Die meisten Tools dieser Plattform helfen in der Entwicklung von Algorithmen die derzeitige Hardware-Ressourcen deutlich übersteigen und dies liegt zurzeit nicht im Bereich der Forschung am IAO.

Die Plattform wurde im Rahmen der Hamiltonian Simulation mit Methoden des Quantum Signal Processing getestet. Hierbei konnte sie leider keine Hilfe bieten, da das Implementieren von Block-Encoding Operatoren und QSP-Sequenzen nicht in dem Funktionsumfang enthalten ist. Positiv zu bewerten ist die Möglichkeit benutzerdefinierte Hardwarearchitekturen zu spezifizieren. Das Fehlen eines Transpilierers für die Schaltkreise ist nachteilig. Man muss an dieser Stelle aber erwähnen, dass Classiq eine Software-Plattform zur Synthese von Quantenschaltungen eingesetzt werden soll und nicht für deren Transpilation. Eine all-in-one Lösung wäre dennoch vorteilhaft.
Obwohl die erste Einführung in die Software sehr zufriedenstellend war und sie sofort, ohne Probleme benutzt werden konnte, war der nachfolgende Support über E-Mail eher schwerfällig. Classiq besitzt eine sehr aktive Slack-Community, in der Fragen zeitnah beantwortet werden. Die graphische Benutzeroberfläche im Browser ist sehr übersichtlich und ist geeignet, um die Plattform für einen einfache und schnelle Schaltkreissynthesen zu benutzen. Außerdem können die Schaltbilder gut für Präsentationen und Veröffentlichungen verwendet werden.
Abschließend kann man sagen, dass der Einsatz der Classiq-Plattform momentan keinen Nutzen für die Forschung in der Gruppe birgt. Bei dem Übergang zu fehlerkorrigierter Hardware, welche deutlich leistungsfähiger ist, könnte die Software nochmal evaluiert werden.

CLARSA – Computing Infrastructure

Hardware

The classical KQCBW computing infrastructure that we have established is shown in Abb. 48. The High-performance computing cluster consists of a login node and three compute nodes. The login node serves as the entry to the HPC nodes. It also works as network file server where all the data is stored. These are mounted on the compute nodes to access the data during the computation. The HPC configuration is shown in

Tabelle 8. The login node is composed of Intel Xenon processor with 48 compute cores and consists of 1.5 TB RAM. A 1 TB SSD on the login node that serves as NFS is connected to the login node. The three compute nodes are AMD Epyc processors with two chips each with 32 compute cores. Hyperthreading is activated on the compute nodes.

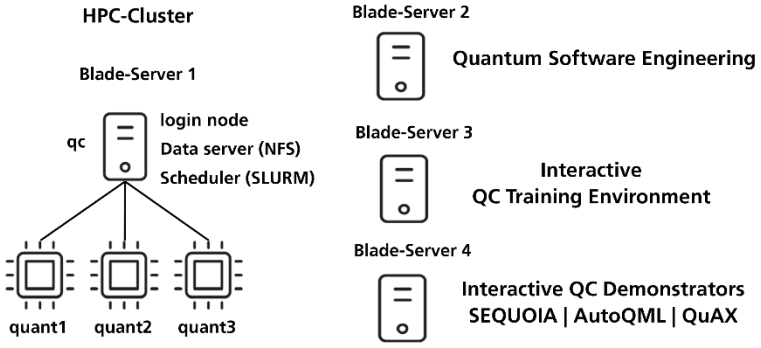


Abb. 85 The classical computing infrastructure established in the project. The resources are distributed in to four categories server different purposes of the project. The HPC-Cluster is used as the computational workhorse where quantum simulators are run. The quantum software engineering server is used to develop the quantum computing software platform. The interactive QC training environment server is used in providing teaching and education and finally the Blade server – 4 is employed to host the demonstrators produced in the project.

Each of these nodes contain 3.6 TB of local fast NVMe SSD drives that can be used for high-speed I/O during computation. Each compute node also contains eight Nvidia RTX A6000 GPU nodes with 48 GB of RAM.

Nodes	Login x 1	Compute x 3
CPU	Intel xenon 2 x 28 cores	AMD Epyc 7542 2 x 32 cores
RAM	1.5 TB DDR4	2 TB DDR4
SSD	1 TB (NFS)	2 x 1.6 TB NVMe
GPU		Nvidia RTX A6000, 48 GB

Tabelle 8 Configuration details of the HPC infrastructure

Scheduler

To manage the computing resources, we installed SLURM package to allocate and schedule the jobs among the available resources based on a queuing system. We have setup three different queuing partitions with different time limits for the convenience of the software development and testing as well as executing heavy and time-consuming programs.

Software

The following software is made available on the HPC for the users as default. Quantum simulators based on Qiskit, Anaconda- a python version management system and jupyter to run iPython notebooks. A jupyter lab is setup for easy access through a web browser. The users can install further software packages based on their requirements.

Benchmarks

We ran benchmarks using the multi-node Nvidia cuQuantum Appliance for three different quantum circuits: Quantum Volume (QV), the Quantum Approximate Optimization Algorithm (QAOA), and Quantum Phase Estimation. The QV model circuits are random instances of circuits used to measure the QV metric, as introduced in [1]. The model circuits consist of layers of Haar random elements of SU(4) gates applied between

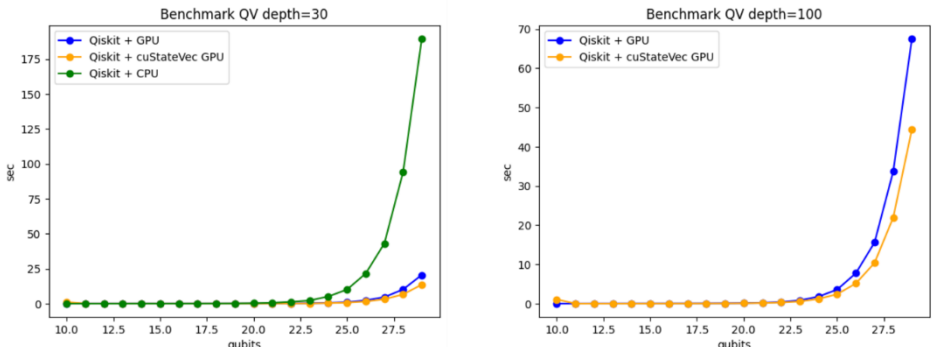


Abb. 86 Scaling of execution time for Quantum Volume circuits with different depth implemented. Qiskit front end is used. CPU and GPU execution time with native Qiskit implementation and Nvidia cuQuantum implementations are compared.

corresponding pairs of qubits in a random bipartition. We ran the QV circuits with a depth of 30 and 100. In Abb. 86, the scaling of the execution time of a QV circuit for up

to 30 qubits when run on a single GPU as compared to execution of 16 CPU cores. Qiskit is used as the front end where Qiskit native implementation of gate execution on GPUs and cuda statevector (cuStateVec) simulator developed in Nvidia cuQuantum appliance are compared with execution time on AMD eypc 7542 32 core processor. As it can be seen in Fig. xx (b) the cuStateVec outperforms Qiskit native implementation of gate execution on GPUs.

References

[1] A. Cross et al. Validating quantum computers using randomized model circuits, Phys. Rev. A 100, 032328 (2019)

Werkzeugketten und Testverfahren für Quanten-Programme

Die Jupyter Notebooks, die im Rahmen von Arbeitspaket 1.2 für den Anwendungsfall Konfigurationspriorisierung entwickelt wurden, sind so entworfen, dass sie den Benutzern das Verständnis für das Verhalten der Hyperparameter erleichtern können. Neben der ausführlichen Dokumentation der Notebooks sind alle Werte anpassbar, so dass Anwender schnell und unkompliziert Hyperparameter anpassen und die Auswirkung auf die Ergebnisse einsehen kann.
Um die Ausführung von Quantum-Code auf realen Backends genauer zu untersuchen, wurde eine Abschlussarbeit betreut, die den bestehenden Toolsupport mit Fokus auf IBM-Infrastruktur analysiert und bewertet. Hierzu wurden Skripte geschrieben, die Backend-Metriken sammeln und clustern. Diese Ergebnisse können in Zukunft dazu genutzt werden, um Quantum-as-a-Service-Lösungen zu entwerfen, bei denen die momentan manuelle Backendauswahl durch Programmanalyse automatisiert stattfindet. Im Rahmen dieser Arbeit wurde das Backend in Ehningen verwendet.
Zusätzlich befasste sich die Abschlussarbeit mit der Analyse der Qiskit Transpilationspipeline. Es wurden systematisch Optimierungslevel und Funktionsparameter gewählt, um zu entscheiden, wie gut die Qiskit Standardwerte funktionieren. Das Ergebnis der Untersuchung hat bestätigt, dass die Defaults im Durchschnitt zuverlässig gute Messungen produzieren und manuelle Eingriffe in die Transpilationsphasen die Messungen eher negativ beeinflussen.

Quantenprogrammiersprachen und Entwicklungsumgebungen sind wesentliche Werkzeuge, um Quantensoftware zu erstellen (AP 5.2). Quantum Computing ist ein interdisziplinäres Forschungsfeld, an dem Stakeholder mit verschiedenen Hintergründen (z.B. Physik, Mathematik, Informatik) mitwirken. Es gibt verschiedene Möglichkeiten, Quantensoftware zu beschreiben (z.B. durch mathematische Formeln, Schaltkreise oder Quellcode), welche von unterschiedlichen Stakeholdern präferiert werden. Aktuelle Quantenprogrammiersprachen und Entwicklungsumgebungen werden den Anforderungen dieses heterogenen Feldes nicht gerecht und haben daher eine hohe Einstiegshürde, z.B. für klassische Softwareentwickler.

Die Konzeption für eine ganzheitliche integrierte Entwicklungsumgebung (IDE) für Quantenprogrammierung wurde begonnen und Abschlussarbeiten angefangen. In einem solchen Entwicklungswerkzeug werden unterschiedliche Möglichkeiten, Quantensoftware zu beschreiben, verknüpft werden, so dass Stakeholder aus unterschiedlichen Domänen entsprechend passende Beschreibungsmöglichkeiten auswählen und verwenden können. Dieses Werkzeug wird mithilfe eines zentralen Datenmodells sowie textuellen und grafischen domänenspezifischen Sprachen (DSLs) realisiert werden. Klassische Features von IDEs, wie Language Smarts (z.B. Autovervollständigung, Refactorings, etc.) und Funktionalitäten, wie Versionskontrolle, Debugging und Quelltextsuche, welche auch für Quantensoftware nützlich sind, müssen ebenfalls unterstützt werden, können in Zukunft als Erweiterungen integriert werden.

Im Bereich der Testverfahren für Quanten-Programme (AP 5.3) wurde eine formale Testtheorie für eine Quanten-While-Sprache entwickelt, die Ideen aus der Conformance-Testtheorie für probabilistische Programme auf Quanten-Programme adaptiert und somit für Quanten-Software nutzbar macht. Diese Testtheorie basiert darauf, dass Quantenprogramme und ihre Eigenschaften als probabilistische Transitionssysteme modelliert werden, so dass die Conformance einer Implementierung eines Quantenprogramms zu seiner Spezifikation (beides dargestellt durch ein probabilistisches Transitionssystem) durch die bekannten formalen Konzepte der (probabilistischen) Simulation (bzw. Bisimulation) beschrieben werden können. Ausgehend von der Conformance-Relation zwischen Spezifikation und Implementierung können dann Kriterien für die Testfallerzeugung formuliert werden, so dass eine Testsuite, welche diese Kriterien erfüllt, auch die jeweilige Conformance Relation bestätigen kann, wenn alle Tests entsprechend durchlaufen. So wird die Bewertung und der Vergleich von Testfällen und Testsuiten basierend auf diesen formalen Abdeckungskriterien möglich. Eine Implementierung dieser Testfallerzeugung auf Basis der formalen Theorie wird aktuell entwickelt. Die formale Testtheorie bildet die Grundlage für weitere aufbauende Arbeiten im Bereich des Testens und der Qualitätssicherung von Quantenprogrammen. Sie geht über existierende Arbeiten hinaus, welche im wesentlichen klassische Testkonzepte auf Quantenschaltkreise übertragen, indem sie die inhärenten Eigenschaften von Quantenprogrammen formal fasst und dann eine Analyse und ein formales Schließen über diese Quanteneigenschaften ermöglichst.

2.6 Arbeitspaket 6 – Wissenstransfer und Verwertung

Definiertes Ziel des sechsten Arbeitspakets war es, laut Projektbeschreibung, die erarbeiteten Forschungsergebnisse den Projektpartnern sowie den Wissenschaftlern im KQCBW, dem assoziierten Unternehmensnetzwerk und der allgemeinen Öffentlichkeit verständlich zur Verfügung zu stellen. Zudem ist das Generieren von Sichtbarkeit des Projekts durch adäquate Öffentlichkeitsarbeit im Fokus des Arbeitspaketes. Mithilfe von interaktiven Austauschformaten und Vertiefungsworkshops wurde die Vernetzung und der Wissenstransfer der Projektergebnisse zielorientiert durchgeführt.

Unter Leitung des **Fraunhofer IAO** wurde dies anhand zweier Arbeitsschritte realisiert (vgl. Gantt-Chart in Abb. 2).

AP 6.1 Öffentlichkeitsarbeit und Vernetzung

- Organisation von Netzwerktreffen
- Pressemitteilungen zu Projektergebnissen
- Aufbau einer Projektwebseite und einer hohen Social-Media-Präsenz zur Sichtbarkeit des Projekts
- Erstellung von Marketingmaterial (bspw. Projektflyer)

AP 6.2 Vertiefungsworkshops und Austauschformate

- Schulungsangebote, Impulsvorträge und Entwickelndentreffen

Jenem Arbeitsplan lagen dabei zwei Meilensteine zugrunde

- **M8:** Aufbau einer deutschen und englischen Projektwebseite mit zielgruppenspezifischen Inhalten sowie Planung und Konzeption eines Entwickelndentreffens zur Vernetzung.
- **M15:** Kommunikation von Projektergebnissen durch Pressemitteilung und einem großen Medienecho sowie Aufbau einer hohen Social-Media-Präsenz über die Kommunikationskanäle der Partner.

Sowohl M8 als auch M15 sind zum Abschluss des Projekts planmäßig erreicht worden (vgl. Abb. 2). Die »SEQUOIA End-to-End« Projekthomepage wurde online genommen und kontinuierlich angepasst.

Im Folgenden wird der finale Stand der Aktivitäten dargestellt.

Öffentlichkeitsarbeit, Veranstaltungen und Wissenstransfer

Wie geplant wurden alle Projektergebnisse kontinuierlich der Wissenschaftscommunity, dem Konsortium sowie der breiten (Fach-)Öffentlichkeit zugänglich gemacht. Zusammenfassend ist dies in Abb. 87 dargestellt.

Die Projektergebnisse wurden in insgesamt 45 Transferveranstaltungen sowohl der breiten Öffentlichkeit kommuniziert als auch der Fachcommunity und dem Konsortium verständlich gemacht. Insgesamt sind dabei ca. 2700 Teilnehmende geschult worden.

Die in der ersten Projektphase erarbeitete Anwenderstudie »Quantencomputing in der industriellen Applikation« wurde seit der Veröffentlichung im Februar 2023 bis zum derzeitigen Zeitpunkt über 1500 Mal abgerufen, wie der Statistik zu entnehmen ist.

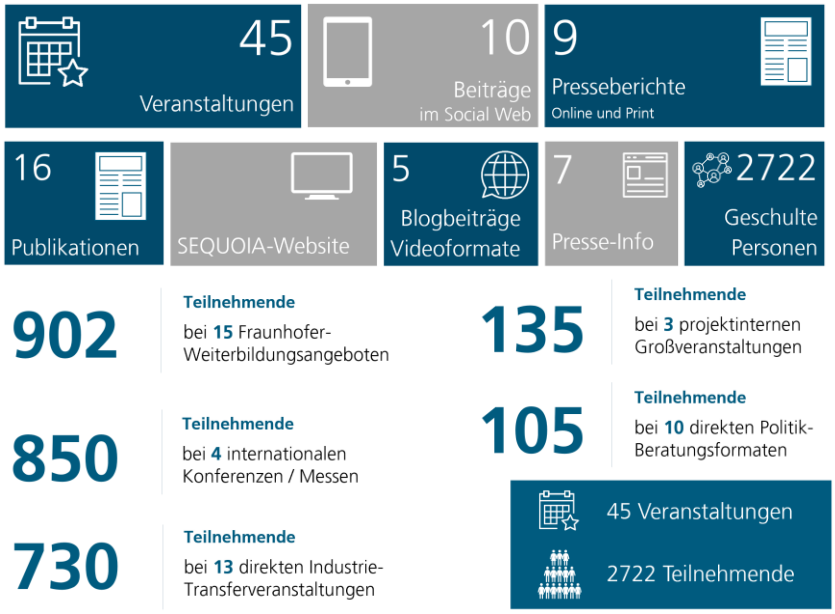


Abb. 87 Der »SEQUOIA End-to-End« Wissenstransfer- und die zugehörige Öffentlichkeitsarbeit

Neben den **projektinternen Veranstaltungen** (inkl. assoziierte Partner)

- 16.02.23 | Projektabschlusstreffen SEQUOIA (45 Teilnehmende)
  | (inklusive Unternehmensbeiratssitzung)
- 17.02.23 | Projekt Kick-off SEQUOIA End-to-End (45 Teilnehmende)
- 8./9.03.24 | KQCBW-weite Entwickelndenkonferenz in Freiburg (45 Teilnehmende)

wurden die erarbeiteten Projektergebnisse in etablierte **Fraunhofer-Weiterbildungsangebote** integriert

- 05.04.23 | 18.10.23 | DigitalDialog from Quantum Awareness to Readiness - Fraunhofer IAO (30 | 30 Teilnehmende)
- 25.04.23 | 04.07.23 | 19.09. | 07.11.23 | Das Quantencomputing-Ökosystem; Das Quantencomputing Ökosystem: Landes- und Bundesförderprojekte - Fraunhofer IAO (32 | 63 | 67 | 87 Teilnehmende)
- 20.06.23 | 04.07.23 | 01.08.23 | 05.09.23 | 07.11.23 | 05.12.23 |
  The Quantum & AI Experience Tour (10 | 6 | 40 | 34 | 33 | 21 Teilnehmende)
- 25.07.23 | Q.AX Tag der offenen Tür (350 Teilnehmende)
- 10.-12.07.23 | Fraunhofer IAO / IAF QC-Schulungsprogramm (8 Teilnehmende)
- 13.-15.11.23 | Fraunhofer IAO/IAF QC-Schulungsproramm (25 Teilnehmende)
  Fokus: Umsetzten von Quantenanwendungen mittels Quantum-Machine-Learning-Techniken sowie nötige Fehlermitigationsverfahren

sowie direkt in die **Industrie transferiert**

- 28.03.23 | Tech-tub HS Pforzheim (30 Teilnehmende)
- 18.04.23 | Eröffnung Quantum & AI Experience Center | Use Case Demonstratoren
  (120 Teilnehmende)
- 21.04.23 | Quantum^{BW} Kick-off | Use Case Präsentation (50 Teilnehmende)
- 14.06.23 | iit Workshop Wiesbaden (30 Teilnehmende)

- 16.06.23 | FpF-Jahrestreffen: Use Case Präsentation (15 Teilnehmende)
- 23.06.23 | Nacht der Wissenschaften Heilbronn (50 Teilnehmende)
- 12.07.23 | Innovationslounge Fraunhofer Alumni (100 Teilnehmende)
- 13./14.07.23 | QuantumBW-Fokustreffen: Quantensoftware-Engineering (50 Teilnehmende, inkl. Unternehmensbeiratssitzung)
- 12.09.23 | Voith Tech Talk (90 Teilnehmende)
- 09.10.23 | Quantum Effects Tour (25 Teilnehmende)
- 26.10.23 | IBM Quantum Industry Exchange (50 Teilnehmende)
- 30.11.23 | Cross-Clustering: Quantum Technologies meets Automotive (40 Teilnehmende)
- 08.12.23 | Zukunftstechnologien des 21. Jahrhunderts | Friedrich-Neuman-Stiftung (30 Teilnehmende)
- 26.01.24 | »Quantum Brunch« Fraunhofer IPA,

Dabei war das Projekt nicht nur auf nationaler, sondern insbesondere auch auf der **internationalen Bühne vertreten:**

- 22.06.23 | Digitalgipfel BW (150 Teilnehmende am Stand)
- 27.-30.06.23 | Teilnahme an World of Quantum (150 Teilnehmende am Stand, 4-tägig)
- 10./11.10.23 | Fachmesse »Quantum Effects« (2000 Teilnehmende auf Messe, ca. 500 Teilnehmende am Stand, Vortrag auf QuantumBW-Bühne, interaktive Demonstrationen und Projekt- und Ergebnisvorstellung auf Bildschirmen)
- 05.03.24 | Vortrag »Inverted-circuit zero-noise extrapolation for quantum gate error mitigation« auf APS March Meeting, 5. März 2024, Minneapolis (USA, 50TN)

Es sei an dieser Stelle separat hervorgehoben, dass in Kollaboration mit der »Geschäftsstelle QuantumBW« die Projektergebnisse ebenfalls zur **Politikberatung** genutzt und in die entsprechenden Delegationen transferiert worden sind. Dabei wurden folgende Transferveranstaltungen durchgeführt:

- 11.01.23 | Austausch Fraunhofer IAO x Bosch (4 Teilnehmende)
- 24.04.23 | Parlamentarischer Abend (30 Teilnehmende)
- 21.06.23 | Austausch mit Staatssekretär Dr. Rapp (8 Teilnehmende)
- 03.07.23 | Besuch MdLs Finanzausschuss (15 Teilnehmende)
- 18.07.23 | Besuch MdBs SPD (15 Teilnehmende)
- 28.11.23 | Besuch Grünen-Fraktionen (5 Teilnehmende)
- 28.11.23 | Besuch Hr. Elberfeld, Hr. Hoffmeister, Hr. Gauthe, Dr. Schütte (5 TN)
- 08.12.23 | Besuch S-TEC Landtagsfraktion der Grünen (inkl. Fraktionsvorsitzender Andreas Schwarz und weitere MdL, 10 Teilnehmende)
- 25.01.24 | Besuch SPD AK-Wissenschaft (5 Teilnehmende)
- 01.03.24 | Besuch IZS Landtagsfraktion CDU (inkl. Wirtschaftsministerin Hoffmeister-Kraut und Fraktionsvorsitzender Hagel und weitere MdL, 8 Teinehmende)

Mithilfe all jener fachlichen Vorträge sowie Workshops auf nationaler und internationaler Ebene konnte ein erfolgreicher Wissenstransfer für die unterschiedlichsten Zielgruppen gewährleistet werden. Sowohl Unternehmen als auch andere Forschungsinstitutionen konnten zielgerecht von dem im Projekt erarbeiteten Fachwissen profitieren.

Wie in Abb. 87 verdeutlicht, beschränkte sich die Öffentlichkeitsarbeit und der Wissenstransfer jedoch nicht nur auf Veranstaltungen und Workshops, sondern erarbeitete außerdem eine Vielzahl an:

Pressemitteilungen

- 13.01.23 | Quantentechnologien für die Wirtschaft nutzbar machen
- 24.02.23 | Quantencomputing in der industriellen Applikation
- 21.04.23 | Ba-Wü startet Innovationsoffensive in den Quantentechnologien
- 26.06.23 | Potenziale der Quantentechnologien: Roadmap für BW
- 18.07.23 | Erstes QuantumBW Treffen zu Quantensoftware-Engineering
- 11.10.23 | Innovation, Inspiration und Kollaboration auf der »Quantum Effects«
- 14.03.24 | Developer Conference zeigt Quantenforschung made in BW

Presseberichte

- 19.01.23 | Quantentechnik soll der Wirtschaft helfen (vogel.de)
- 27.02.23 | QC in der industriellen Applikation (messe-stuttgart.de)
- 10.03.23 | Wo kann QC in der Industrie zum Einsatz kommen?
- 19.03.23 | Wie Unternehmen Quantensoftware zielgerichtet einsetzen können (industry-of-things.de)
- 06.06.23 | Potenziale der Quantentechnologien: Roadmap für BW | BioRegio STERN | Thinking business forward (bioregio-stern.de)
- 12.07.23 | Innovationslounge Fraunhofer: Quantencomputing (atreus.de)
- 19.07.23 | Erstes QuantumBW Treffen zu Quantensoftware-Engineering (messe-stuttgart.de)
- 09.10.23 | Quantentechnologien nähern sich der Anwendung (Krankenhaus-IT Journal Online)
- 25.10.23 | Fraunhofer IAO ist neues Mitglied bei Photonics BW: Photonics BW

Glossarbeiträge und Videoformate

- Glossarbeitrag: Quantencomputer – eine Definition von Dr. Christian Tutschku (frankfurt-holm.de)
- Dr. Marco Roth - Was kann Quantencomputing? (Industrie 4.0 & Ilot auch erschienen in Iotdesign)
- Initiative Wirtschaft 4.0 meets Quantencomputing
  - IW4.0 meets Quantencomputing | Teil 1: Wo steht die Technologie - wie ist BW aufgestellt? - YouTube
  - IW4.0 meets Quantencomputing | Teil 2: Wie entwickelt sich der Quantencomputer? - YouTube
  - IW4.0 meets Quantencomputing | Teil 3: Was ist der Nutzen für die Menschheit? - YouTube

Projektwebseiten

- SEQUOIA End-to-End - Transparentes Quanten-Software-Engineering und Algorithmendesign anwendungszentrierter End-to-End Lösungen (fraunhofer.de)
- SEQUOIA End-to-End – Software-Engineering industrieller, hybrider Quantenanwendungen und -algorithmen (sequoia-iao.de)

LinkedIn-Aktivitäten

- LinkedIn Fokusseite: The Quantum Future
- LinkedIn Gruppe: Quantum Village Ehningen

3 Publikationen

Im »SEQUOIA End-to-End« Projekt sind die folgenden Publikationen in Finalisierung bzw. wurden z.T. schon eingereicht oder angenommen:

Weitere Publikationen umfassen:

- **Angenommen**

[1] D. Eichhorn, T. Pett, T. Osborne, and I. Schaefer, "Quantum Computing for Feature Model Analysis: Potentials and Challenges", in *27th ACM International Systems and Software Product Lines Conference (SPLC'23)*, 2023, doi: 10.1145/3579027.3608971.

[2] S. Garhofer and O. Bringmann, "Using an A*-based Framework for Decomposing Combinatorial Optimization Problems to Employ NISQ Computers", *2023 Quantum Information Processing* (QIP), Gent, BEL, 2023, doi: 10.1007/s11128-023-04115-w.

[3] M. Scheerer, J. Klamroth, S. Garhofer, F. Knäble and O. Denninger, "Experiences in Quantum Software Engineering", *2023 IEEE International Parallel and Distributed Processing Symposium Workshops* (IPDPSW), St. Petersburg, FL, USA, 2023, pp. 552-559, doi: 10.1109/IPDPSW59300.2023.00095.

[4] J. Klamroth, B. Beckert, M. Scheerer and O. Denninger, "QIn: Enabling Formal Methods to Deal with Quantum Circuits", *2023 IEEE International Conference on Quantum Software* (QSW), Chicago, IL, USA, 2023 pp. 175-185, doi: 10.1109/QSW59989.2023.00029.

[5] J. Ammermann et al., "Quantum Approach to the Configuration Selection and Prioritization Problems", *2024 ACM/IEEE International Workshop on Quantum Software Engineering* (Q-SE 2024), Lisbon, Portugal, 2024, ACM, New York, NY, USA, *to appear.*

- **Eingereicht**

[6] P. A. Matt and M. Roth, "A Heuristic for Solving the Irregular Strip Packing Problem with Quantum Optimization", 2023, arXiv: 2402.17542. [Publikation war in Vorbereitung am Ende von SEQUOIA, vollendet in SEQUOIA End-to-End]

[7] K. F. Koenig, F. Reinecke, W. Hahn and T. Wellens, "Inverted-circuit zero-noise Extrapolation for Quantum Gate Error Mitigation", 2024, arXiv: 2403.01608.

[8] A. Sturm, B. Mummaneni, L. Rullkötter, "Unlocking Quantum Optimization: A Use Case Study on NISQ Systems", 2024, arXiv: 2404.07171.

[9] N. Schillo, A. Sturm, "Quantum Circuit Learning on NISQ Hardware", 2024, arXiv: 2405.02069.

[10] F. Rapp, D. A. Kreplin, M. Roth, "Reinforcement Learning-based Architecture Search for Quantum Machine Learning", 2024, arXiv: 2406.02717.

- **In Vorbereitung**

[11] L. Rullkötter, B. C. Mummaneni, S. Weber and C. Tutschku, "Hamiltonian Simulation via Variational Block Encoding".

[12] M. Scheerer, J. Klamroth and O. Denninger, "Detecting and Tolerating Faults in Hybrid Quantum Software Systems".

[13] J. Klamroth, M. Scheerer, O. Denninger, B. Beckert, "Considerations on Fault-tolerant Simulations of Quantum Circuits using Floating-Point numbers".

[14] J. Klamroth, M. Scheerer, O. Denninger, B. Beckert, "Verifying QAOA Implementations: A Case Study".

[15] M. Fehling-Kaschek, C. Brockt-Haßauer, V. Shatokhin, A. K. Jain, A. Buchleitner, "Model Network analysis on a Quantum Computer".

[16] O. Voigt, K. Schroven, M. Fehling-Kaschek, "Optimizing Cascading Networks using QC-Methods".

- **Abschlussarbeiten**

[17] Niclas Schillo, "Quantum Algorithms and Quantum Machine Learning for Differential Equations", 2023, doi: 10.18419/opus-13866.

[18] Leons Rullkötter, "Hamiltonian Simulation using Quantum Eigenvalue Transformation with Variational Block-Encoding", 2024.